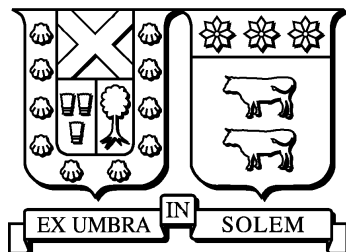


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



“MODELO PREDICTIVO PARA LA RETENCIÓN DE ESTUDIANTES EN PRIMEROS AÑOS”

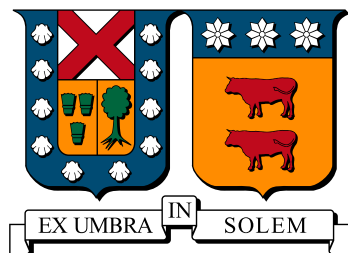
JOSÉ PABLO DEL RÍO ARTEAGA

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: LIOUBOV DOMBROVSKAIA

JUNIO 2018

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



**“MODELO PREDICTIVO PARA LA
RETENCIÓN DE ESTUDIANTES EN
PRIMEROS AÑOS”**

JOSÉ PABLO DEL RÍO ARTEAGA

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO**

PROFESOR GUÍA: LIOBOV DOMBROVSKAIA

PROFESOR CORREFERENTE: PATRICIO RODRÍGUEZ VALDÉS

JUNIO 2018

MATERIAL DE REFERENCIA, SU USO NO INVOLUCRA RESPONSABILIDAD DEL AUTOR O DE LA INSTITUCIÓN

Agradecimientos

Como primera instancia quisiera agradecer a mis padres por la educación y el apoyo que me dieron, que me permitió llegar hasta este punto.

También quiero agradecer a Ligia Calderón por acompañarme y apoyarme en todo momento, con su inmenso cariño en todo el transcurso de esta memoria.

Quiero también agradecer a mi profesora guía Liubov Dombrovskaia, por que sin su guía esta memoria no habría alcanzado de ninguna manera la calidad de los resultados que se obtuvieron.

También quiero agradecer a la Universidad Técnica Federico Santa María por haberme dado todas las herramientas para cumplir con mis objetivos.

Por último quisiera agradecer a mis compañeros, que me acompañaron en todo el transcurso de la carrera.

Resumen

Se crean modelos capaces de predecir la deserción de estudiantes de Ingeniería Civil de la Universidad Técnica Federico Santa María en el primer y segundo año académico.

Se utilizó la información de estudiantes referente al pre ingreso a la universidad, y la información académica del estudiante hasta el término del primer semestre.

Los mejores modelos construidos se lograron con los algoritmos *random forest*, para la predicción de la deserción en el primer año académico y *tgradient boosting* para la predicción de la deserción en el segundo año académico, con valores del 92,4 % y 92,9 % para la métrica *F1 score* respectivamente.

Abstract

Models are created to be capable of predicting the dropout of students of Civil Engineering of Universidad Técnica Federico Santa María in the first and second academic year.

It is used the information of students regarding to pre admission at the university and additionally the academic information until the end of the first semester.

The best constructed models were achieved with the random forest algorithm for classification of the dropout in the first year, and with the gradient boosting algorithm for the dropout in the second year, with a F1 score of 92,4 % and 92,9 % respectively.

Índice de Contenidos

Agradecimientos	III
Resumen	IV
Abstract	V
Índice de Contenidos	VI
Lista de Tablas	IX
Lista de Figuras	XI
Introducción	1
1. Definición del Problema	4
1.1. Deserción en la Universidad Técnica Federico Santa María	5
1.2. Alcances y objetivos de la memoria	8
2. Estado del Arte	10
2.0.1. Deserción en Chile y en UTFSM	16
2.1. Marco Teórico	20
2.1.1. Minería de datos	20

2.1.2. Metodología CRISP-DM	21
3. Propuesta de solución	24
3.1. Comprensión del negocio	24
3.2. Propuesta	24
3.3. Plan de proyecto	25
4. Implementación de la solución	27
4.1. Comprensión de los datos	27
4.1.1. Recolección y descripción de datos iniciales	27
4.2. Preparación de los datos	30
4.2.1. Integración de datos	30
4.2.2. Construcción de datos	32
4.2.3. Exploración de los datos	33
4.2.4. Limpieza de datos	36
4.2.5. Cambio de formato de datos	38
4.3. Fase de Modelado	40
4.3.1. Selección de técnicas de modelado	40
4.3.2. Generación de plan de prueba	42
4.3.3. Selección de factores	44
4.3.4. Ajuste de parámetros de los algoritmos	49
5. Evaluación de la solución	50
5.0.1. Evaluación de modelos que predicen la deserción en el primer año académico	50
5.0.2. Evaluación de modelos que predicen la deserción en el segundo año académico	51

Conclusiones	53
Bibliografía	57
5.1. Anexos	59
5.1.1. Código en Python utilizado en este trabajo	60

Índice de tablas

2.1. Factores utilizados para predecir la deserción en [1].	13
2.2. Factores utilizados para predecir la deserción en [2].	14
2.3. Ranking de factores que inciden en la retención en la UTFSM [3].	19
2.4. Ranking de factores que inciden en la deserción en la UTFSM [3].	20
4.1. Nombres y descripciones de los factores de los datos proporcionados por Dirección de Admisión.	28
4.2. Nombres y descripciones de los factores de la primera fuente de datos pro- porcionada por Dirección de Estudios.	29
4.3. Nombres y descripciones de los factores de las notas proporcionadas por Dirección de Estudios.	30
4.4. Quintiles de puntaje PSU de Lenguaje y Comunicación, puntaje PSU de Ma- temáticas y puntaje NEM.	33
4.5. Porcentajes de deserción y retención en el primer año, según el número de cursos reprobados en el primer semestre.	36
4.6. Porcentajes de deserción y retención en el segundo año, según el número de cursos reprobados en el primer semestre.	36
4.7. Factores escalados para ser utilizados en el algoritmo <i>support vector machine</i>	39

4.8. Factores codificados con codificación <i>One-Hot</i>	40
4.9. Top 20 de dependencia de factores con la deserción en el primer año académico.	46
4.10. Top 20 de dependencia de factores con la deserción en el segundo año académico.	47
4.11. Percentiles utilizados para modelos que predicen la deserción en el primer año académico.	49
4.12. Percentiles utilizados para modelos que predicen la deserción en el segundo año académico.	49
5.1. Resultados de modelos que predicen la deserción en el primer año académico.	51
5.2. Métricas de los resultados de modelos que predicen la deserción en el primer año académico.	51
5.3. Resultados de modelos que predicen la deserción en el segundo año académico.	52
5.4. Métricas de los resultados de modelos que predicen la deserción en el segundo año académico.	52
5.5. Valores y significados para el factor rama educacional del colegio del estudiante.	59
5.6. Parámetros utilizados en los modelos construidos con el algoritmo de clasificación <i>random forest</i>	59
5.7. Parámetros utilizados en los modelos construidos con el algoritmo de clasificación <i>k-nearest neighbors</i>	59
5.8. Parámetros utilizados en los modelos construidos con el algoritmo de clasificación <i>support vector machine</i>	60
5.9. Parámetros utilizados en los modelos construidos con el algoritmo de clasificación <i>gradient boosting</i>	60

Índice de figuras

1.1. Porcentajes acumulados de deserción y retención en el primer año académico por año de ingreso, en el Campus Santiago.	5
1.2. Porcentajes acumulados de deserción y retención en primer año académico por año de ingreso, en el Campus Casa Central.	6
1.3. Porcentajes acumulados de deserción y retención en el segundo año académico, en el Campus Santiago	7
1.4. Porcentajes acumulados de deserción y retención en el segundo año académico por año de ingreso, en el Campus Casa Central	8
2.1. Etapas del proceso CRISP-DM [15].	23
4.1. Tasas de reprobación en el curso de Introducción a la Física por cohorte. . .	34
4.2. Tasas de reprobación en el curso de Matemáticas por cohorte.	35
4.3. Esquema <i>k-fold</i> utilizado para la selección de percentil, para modelos de predicción de deserción en el primer año académico.	48

Introducción

La deserción universitaria es el acto de que un estudiante universitario se retire de la carrera que cursa de forma definitiva por diversas razones. Esta situación afecta a diversos actores como a universidades, padres, estado y estudiantes, debido a ello es del interés de muchos de que este problema sea mitigado al corto y largo plazo.

En el siguiente documento se presentará la confección de una herramienta automatizada que tiene como objetivo ayudar a mitigar el problema de la deserción universitaria en la Universidad Técnica Federico Santa María (UTFSM), mediante la predicción anticipada de estudiantes de las carreras de Ingeniería Civil que desertarán en el primer o segundo año académico, haciendo uso de la información disponible sobre los estudiantes hasta el término del primer semestre académico. La información utilizada corresponde a la del pre ingreso a la universidad y a la del desempeño académico en el primer semestre. La temprana predicción de esos estudiantes, le permitirá a la UTFSM tomar medidas cautelares sobre dichos estudiantes en riesgo y así tratar de ayudarlos para prevenir la desafortunada situación.

En el capítulo 1 se presenta la definición del problema de la deserción universitaria, además de mencionar a los diversos actores a los cuales les afecta. También, se detalla la situación de la deserción en el primer y segundo año académico de las carreras de Ingeniería Civil de la UTFSM, realizando un análisis sus dos campus más grandes, Campus Santiago y Campus Casa Central, y se comparará con la situación a nivel nacional para las carreras de Ingeniería Civil en las universidades pertenecientes al Consejo de Rectores.

En el capítulo 2 se habla acerca de trabajos previos referentes a la herramienta a construir y a la deserción en instituciones de educación superior. Se detallarán los modelos similares

construidos cuyos objetivos sean detectar la deserción temprana de estudiantes, y los factores utilizados para sus construcciones. También se detallará la situación y las causas de la deserción en carreras de Ingeniería en el mundo y en Chile.

En el capítulo 3 se planteará el problema como un problema de clasificación de minería de datos y se presentará una solución al problema, la cual consiste en la creación de modelos para predecir la deserción en el primer o segundo año académico en la UTFSM, utilizando la metodología CRISP-DM para proyectos de minería de datos. Por último, se detallará el plan de proyecto para este trabajo, mencionando las actividades que se realizarán en cada etapa de la metodología utilizada.

En el capítulo 4 se detalla todo el proceso de minería de datos utilizando la metodología CRISP-DM. Se describen las fuentes de datos entregadas por diferentes unidades de la UTFSM con las cuales se construirán los modelos. Después se detalla la preparación de los datos, la cual consiste en la integración de las bases de datos, la construcción de nuevos factores a partir de la base de datos integrada, seguido por la exploración de los datos que consiste en la búsqueda de pistas ocultas en la información de manera de guiar el proceso y por último se detalla la limpieza realizada a los datos.

Después de la preparación de los datos se procede a la fase de modelado, en donde se seleccionaron las técnicas (algoritmos) de modelado, las cuales son *random forest*, *k-nearest neighbors*, *support vector machine* y *gradient boosting*. Se genera un plan de prueba en donde se definen las métricas para evaluar los modelos construidos y también se selecciona el criterio para medir el desempeño de los modelos con el fin de poder seleccionar los mejores. Adicionalmente, se realiza un proceso de selección de factores por cada uno de los diferentes modelos, seleccionando el mejor percentil de factores considerando la dependencia entre los factores con lo que se quiere predecir la deserción o retención en el primer o segundo año académico, utilizando el algoritmo de información mutua (*mutual information*). Finalmente se ajustan los parámetros a valores óptimos para cada uno de los modelos.

En capítulo 5 se evalúa el desempeño de los modelos utilizando el criterio escogido en la fase de modelado. El mejor modelo para predecir la deserción en el primer año académico es aquel construido con el algoritmo *random forest*. Para el caso de la predicción de la deserción

en el segundo año académico, el mejor modelo es aquel construido con el algoritmo *gradient boosting*.

Finalmente, se presentan las conclusiones correspondientes a este trabajo, considerando los modelos seleccionados y los factores con más importancia para la predicción de la deserción en el primer y segundo año académico. También se destaca el uso de la herramienta seleccionada, la metodología utilizada y los posibles trabajos a futuro.

Capítulo 1

Definición del Problema

El aumento en la retención de estudiantes es un objetivo a largo plazo para cualquier institución de educación superior, mientras que los efectos negativos de la deserción de estudiantes son evidentes para todos los involucrados. Para los padres y el gobierno, dependiendo el caso, significa perder el dinero invertido en la carrera del estudiante. Para la universidad significa la pérdida del ingreso proporcionado al no prestar más el servicio al estudiante, además de afectar a su acreditación, asignación de fondos y ambiente universitario. Por último, el gran afectado, el estudiante, perdió todo el esfuerzo y dedicación puesta en la carrera. Por otro lado, una de las grandes consecuencias positivas sobre los estudiantes que no desertan es que serán mejores profesionales, y por ende podrían optar a un mejor estándar de vida.

Este problema no es ajeno para las carreras de Ingeniería Civil de la Universidad Técnica Federico Santa María por lo que es necesario tener un mecanismo para poder detectar tempranamente la deserción de los estudiante, con el fin de poder tomar las medidas necesarias para la retención de estos estudiantes en peligro.

1.1. Deserción en la Universidad Técnica Federico Santa María

Con el fin de poder mostrar la situación con respecto a la deserción de estudiantes en las carreras de Ingeniería Civil en la Universidad Técnica Federico Santa María (UTFSM), se mostrarán los porcentajes de deserción en el primer y segundo año académico, considerando una muestra de 11156 estudiantes pertenecientes a las carreras de Ingeniería Civil en los campus Casa Central (Valparaíso) y Santiago.

En la figura 1.1 se muestran los porcentajes de deserción y retención en las carreras de Ingeniería Civil en el primer año académico, según el año de ingreso entre el 2009 y 2016, en el Campus Santiago. Para este caso los porcentajes de deserción varían entre 7 % y 25 %, además se puede observar que existe un aumento paulatino cada año desde el 2009 hasta el 2013, alcanzando un peak del 25 %. El porcentaje promedio de deserción para este caso es del 15,13 %, según [4] se encontraría por debajo del promedio nacional de 24,3 %, para las carreras de Ingeniería Civil en universidades pertenecientes al Consejo de Rectores.

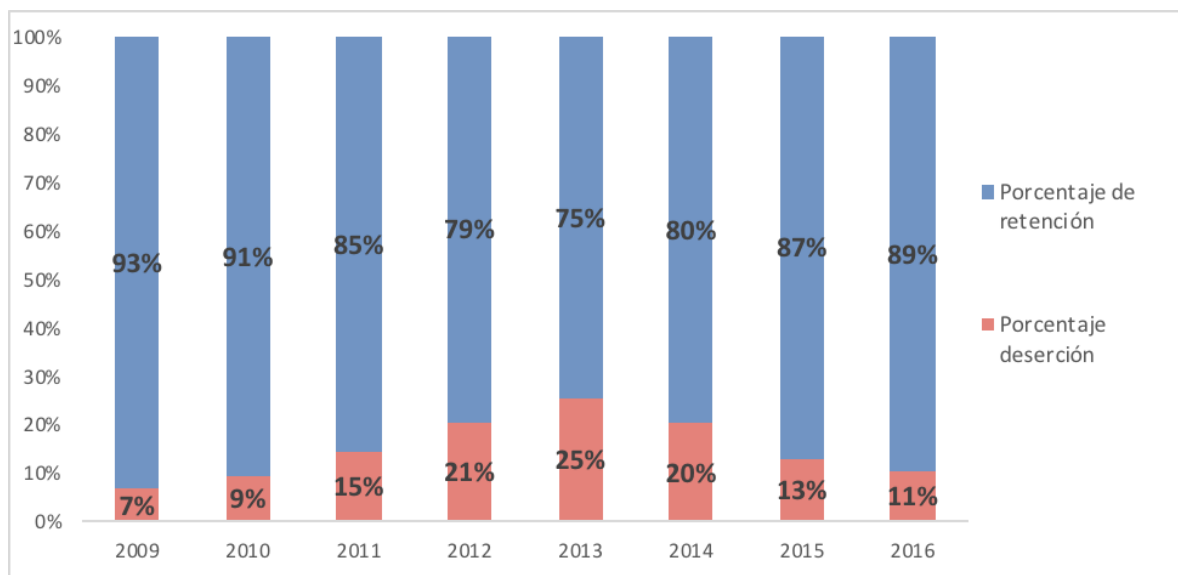


Figura 1.1: Porcentajes acumulados de deserción y retención en el primer año académico por año de ingreso, en el Campus Santiago.

En la figura 1.2, se muestran los porcentajes de deserción y retención para el primer año académico en el Campus Casa Central. En este caso se pueden observar que el rango de porcentajes de deserción varía entre 7 % y 21 %, alcanzando la máxima en el año 2014 con un 21 %. El porcentaje promedio de deserción en el primer año para el Campus Casa Central es de 14,71 %, también por debajo de la tasa promedio nacional.

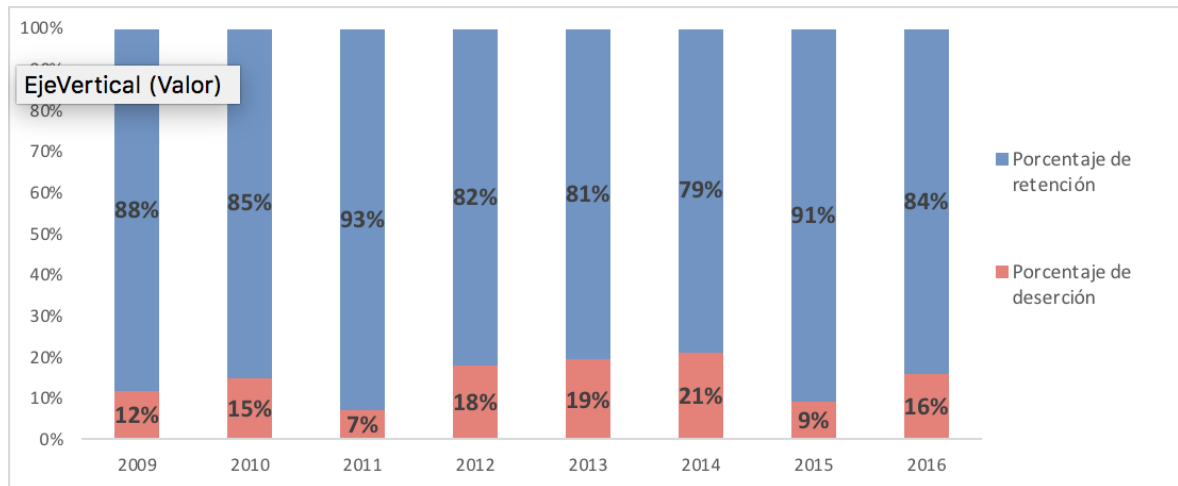


Figura 1.2: Porcentajes acumulados de deserción y retención en primer año académico por año de ingreso, en el Campus Casa Central.

En la figura 1.3, se pueden observar los porcentajes de deserción y retención en el segundo año académico en el Campus Santiago. Los porcentajes de deserción aumentan notablemente para el segundo año académico, el rango varía entre el 19 % y 42 %, alcanzándose el peak para el cohorte del año 2013 con un 42 %. El porcentaje de deserción promedio para este caso es del 29,13 %, según [4] se encontraría por debajo del promedio nacional de 37,6 %, para las carreras de Ingeniería Civil en universidades pertenecientes al Consejo de Rectores.

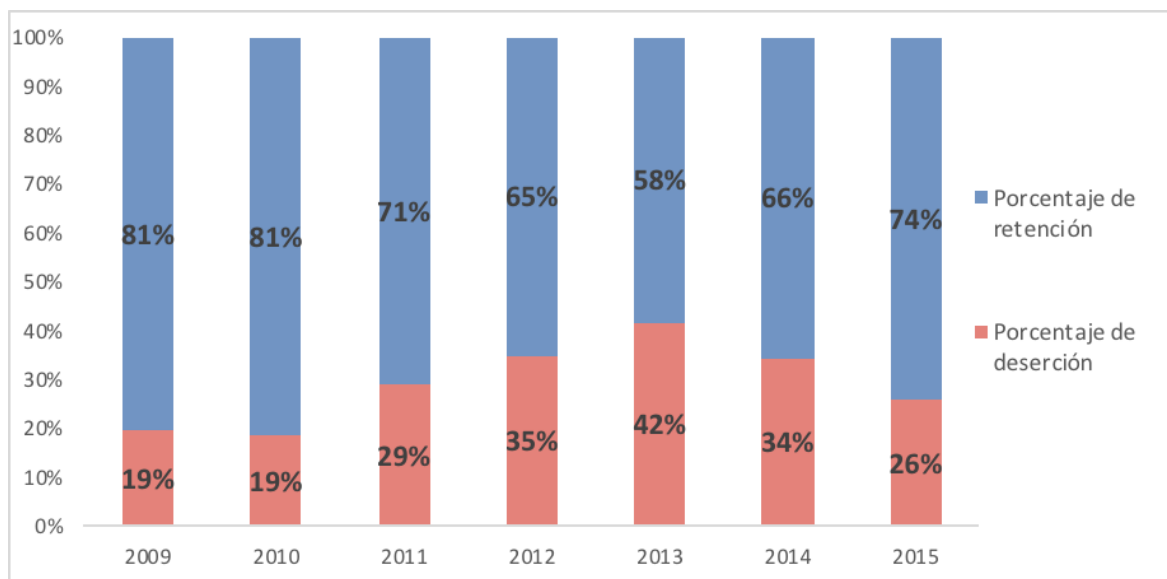


Figura 1.3: Porcentajes acumulados de deserción y retención en el segundo año académico, en el Campus Santiago

En la figura 1.4, se pueden observar los porcentajes de deserción y retención en el segundo año académico en el Campus Casa Central. El rango de los porcentajes de deserción varia entre el 22 % y el 37 %, alcanzándose el peak para el cohorte del año 2014 con un 37 %. El porcentaje de deserción promedio para este caso es del 28,25 %, por lo que también se encontraría por debajo del promedio nacional de para las carreras de Ingeniería Civil en universidades pertenecientes al Consejo de Rectores.

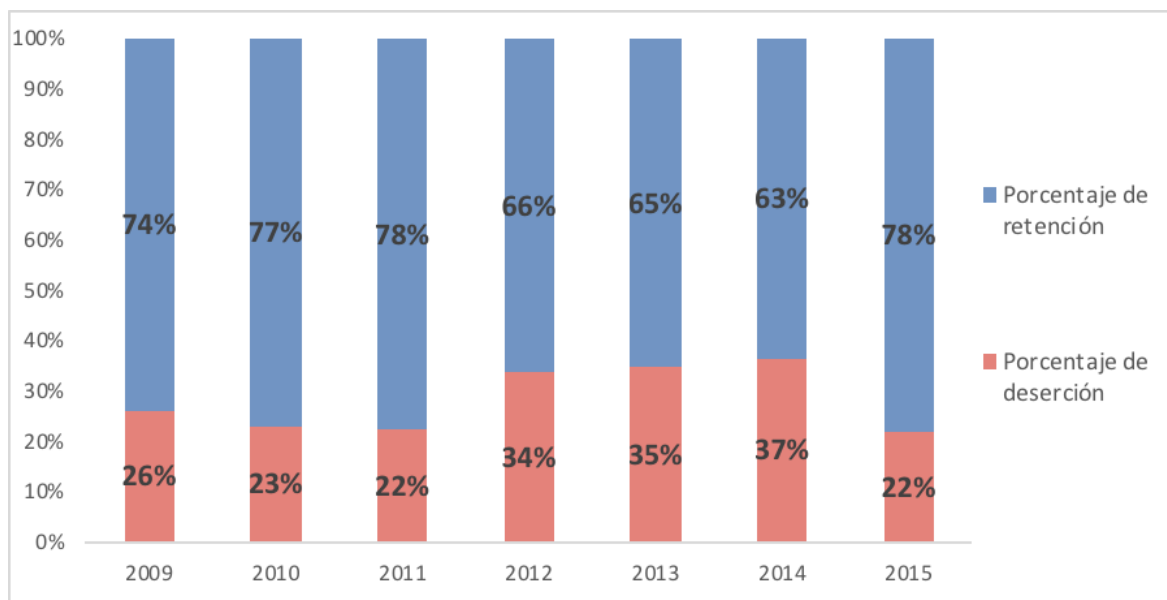


Figura 1.4: Porcentajes acumulados de deserción y retención en el segundo año académico por año de ingreso, en el Campus Casa Central

Según lo anterior, se puede afirmar, que la deserción en el Campus Santiago es un poco mayor que en el Campus Casa Central. Con relación a la deserción en el primer y segundo año académico se puede observar que los porcentajes de deserción promedios para ambos campus son bastante menores que los porcentajes promedio de deserción para las carreras de Ingeniería Civil de las universidades pertenecientes al Consejo de Rectores. Tomando lo anterior en cuenta se puede ver que la UTFSM se encuentra por un buen camino en cuanto a los porcentajes de deserción.

1.2. Alcances y objetivos de la memoria

En la confección de esta memoria, se buscara crear modelos capaces de predecir con gran calidad la deserción en las carreras de Ingeniería Civil de la UTFSM, y con esto le permita a la institución poder aplicar medidas a los estudiantes en riesgo de deserción antes de que el evento se produzca.

Objetivo general de la memoria: Creación de modelos capaces de predecir la deserción en

el primer y segundo año académico en las carreras de Ingeniería Civil en la UTFSM.

Como objetivo específico, identificar factores que más inciden en la deserción y retención en el primer y segundo año académico en las carreras de Ingeniería Civil en la UTFSM.

Capítulo 2

Estado del Arte

El problema de la deserción de estudiantes se ha manifestado en una gran cantidad de establecimientos educativos a lo largo del tiempo. Esto ha servido de motivación para dar origen a muchos trabajos que tienen como objetivo detectar tempranamente una posible situación de deserción, ya sea, estudiando y comprendiendo los factores que inciden directamente en la deserción o, la creación de herramientas automatizadas que la detecten tempranamente.

Actualmente existen dos comunidades de investigación llamadas *Learning Analytics* y *Educational Data Mining* (EDM) que se dedican a utilizar la creciente recolección de datos para la creación de herramientas que permitan evaluar situaciones de deserción [5]. Estas herramientas permiten clasificar a un estudiante entre los dos posibles casos que se pueden dar en un establecimiento educativo, que complete su proceso educativo o que deserte en el camino. Esto se logra a través de modelos que utilizan técnicas (algoritmos) de clasificación, para este problema en particular las técnicas más utilizadas son las de árboles de decisión, redes neuronales y redes bayesianas [6].

Los primeros trabajos relacionados con el tema, se focalizaron en estudiar los factores que inciden en la deserción de estudiantes. En uno de los más conocidos, se estudiaron los factores que inciden en la deserción de estudiantes en un college en los Estados Unidos, según el género, raza, clase social y desempeño [7]. En el estudio se detalla, que por cada 100 estudiantes (67 en programas de 4 años y 33 en programas de 2 años), solo el 59 % obtendrá un

título y el 41 % desertará. Tinto determina que al nivel del estudiante, sus objetivos educacionales y ocupacionales son críticos en la deserción. En cuanto al nivel institucional menciona que los factores que determinan la deserción son: la adaptación del estudiante tanto social como intelectual a la institución, la dificultad de adaptarse a los estándares académicos, y la incongruencia entre el estudiante y el establecimiento.

En [8] se condujo un estudio en la segunda institución con más ingenieros egresados en los Estados Unidos, cuyo objetivo era detectar las principales causas de la decisión de desertar en ingeniería. En el estudio, se menciona que la tasa de graduación en 6 años para la carrera de ingeniería en la institución es del 45 %. Para este estudio se utilizó la información de 113 estudiantes que desertaron de la carrera de ingeniería, la cual se obtuvo a partir de entrevistas y encuestas. Una vez finalizado el estudio, se concluyó que existen 3 factores que más influyen la decisión del estudiante de desertar de ingeniería:

- Enseñanza y orientación pobres.
- Dificultad con la malla de ingeniería.
- Falta de sentido de pertenencia a la carrera de ingeniería.

Entre los 3 factores se destaca que el más influyente por sobre los otros es la falta de sentido de pertenencia a la carrera de ingeniería, que se puede ver relacionado con el pensamiento de la falta de eficacia en la carrera de ingeniería por parte del estudiante, que además puede deberse a eventos circunstanciales en la carrera.

En [9] se realizó un análisis de supervivencia sobre 100.179 estudiantes de ingeniería (con diferentes características) en programas de 6 años, de 9 universidades diferentes en los Estados Unidos, con el objetivo de detectar la asociación de la deserción con diferentes factores. Según el estudio realizado, se detectaron los siguientes factores según diferentes características de los estudiantes, que inciden en el riesgo de deserción:

- Los puntajes de las pruebas de admisión a la universidad (SAT) son malos indicadores para una deserción a largo plazo, pero son buenos indicadores para la deserción a corto plazo (hasta tercer semestre).

- Las estudiantes mujeres poseen un mayor riesgo de desertar en el tercer y quinto semestre, en comparación a los hombres. El riesgo se mantiene igual para ambos géneros en los otros semestres.
- Con excepción de los estudiantes con puntajes superiores a 550 en la prueba de admisión (SAT), existe un mayor riesgo de deserción en el tercer semestre académico.
- La prueba de admisión de matemáticas predice mejor el riesgo de deserción que la prueba verbal de admisión. Entre más bajo sea el puntaje en la prueba de admisión de matemáticas, existe más riesgo que el estudiante deserte.
- Estudiantes con puntajes en la prueba de admisión verbal entre 200 y 500, son menos propensos a desertar que los estudiantes con puntajes entre 500 y 600. Esto se puede explicar con que los estudiantes con mejores puntajes en la prueba verbal tienden a cambiarse más de disciplina que requieren más habilidades verbales que los estudiantes con menos puntajes en la misma prueba.

En [10] se realizó un estudio con el fin de entender los factores que influyen la retención de estudiantes en la carrera de ingeniería, con el objetivo de mejorar la retención. Para el estudio se utilizaron los datos de múltiples universidades en los Estados Unidos, con información referente al pre ingreso a las instituciones de educación superior (información demográfica y académica), con la cual se estudió la dependencia de los factores raza, género, nota promedio de enseñanza media, prueba de admisión SAT de matemáticas, prueba de admisión SAT verbal y la ciudadanía. Dentro de las conclusiones del estudio, el promedio de la nota de enseñanza media y el puntaje de la prueba de admisión de matemáticas están positivamente correlacionadas con la retención del estudiante, es decir, entre más alto ambos factores, mejor es la probabilidad de retención, caso contrario con la prueba de admisión verbal, que se encuentra negativamente correlacionada con la retención (entre mayor el puntaje, mayor es la probabilidad de deserción). Los otros factores estudiados, mostraron influencia en la retención de estudiantes, pero no se probó consistentemente, si existe una correlación positiva o negativa.

Trabajos posteriores se enfocan en la creación de modelos que predicen la deserción académica en forma temprana, mediante el uso de métodos estadísticos o minería de datos, como se

realiza en [1], trabajo cuyo objetivo es crear un modelo para predecir la deserción en el segundo semestre en una universidad ubicada en una región en el medio oeste de los Estados Unidos, con aproximadamente 23000 estudiantes y que posee una deserción promedio del 21,03 % de estudiantes nuevos en el segundo semestre. En la tabla 2.1 se pueden observar los 39 factores utilizados para la creación del modelo predictivo en este trabajo, los cuales son de tipo académicos, financieros y demográficos. Se utilizaron 3 algoritmos de clasificación diferentes en este trabajo para la construcción de modelos, redes neuronales artificiales, regresión logística y árboles de decisión, los cuales fueron validados mediante una validación cruzada de 10 subconjuntos. El mejor ajuste fue logrado con el algoritmo de red neuronal artificial, logrando clasificar correctamente el 68,55 % de los estudiantes clasificados como desertores y el 93,83 % de los estudiantes clasificados como no desertores, con un 81,19 % del total de estudiantes clasificados correctamente.

Factor	
College	Puntaje de prueba SAT de ingles
Grado de college	Puntaje de prueba SAT de lectura
Major de college	Puntaje de prueba SAT de matemáticas
Concentración de college	Puntaje de prueba SAT de ciencias
Horas de cursos en el primer semestre	Edad
Horas de cursos aprobados en el primer semestre	Promedio de enseñanza media
Promedio del primer semestre	Año y mes de graduación de enseñanza media
Promedio acumulado del primer semestre	Puntaje de TOEFL
Horas de cursos en el segundo semestre	Horas convalidadas desde otra institución de educación superior
Horas de cursos aprobados el segundo semestre	Horas convalidadas desde la enseñanza media
Promedio del segundo semestre	Tipo de admisión
Promedio acumulado del segundo semestre	Estado de residencia en los Estados Unidos
Raza	Ayuda financiera recibida para el primer semestre
Género	Ayuda financiera recibida para el segundo semestre
Código de residencia	Crédito financiero del primer semestre
Estado civil	Beca del primer semestre
Puntaje de prueba SAT de comprensión	Trabajo federal para financiamiento del primer semestre
Beca del segundo semestre	Crédito financiero del segundo semestre
Trabajo federal para financiamiento del segundo semestre	

Tabla 2.1: Factores utilizados para predecir la deserción en [1].

En el trabajo [2], se busca crear un modelo para predecir la probabilidad de deserción de

estudiantes de primer año, en la institución *Institute of Engineering and Technology of VBS Purvanchal University*, ubicada en India, con el fin de poder destinar los limitados recursos, a programas para ayudar a los estudiantes con más probabilidades de desertar, debido a que han investigado que estos programas ayudan en gran manera a estudiantes en peligro de deserción. Los factores utilizados en este modelo se puede observar en la figura 2.2. En este trabajo se utiliza un algoritmo clasificador bayesiano ingenuo, el cual se basa en el teorema de bayes, para encontrar la probabilidad de que un estudiante deserte. El modelo construido fue validado mediante una validación cruzada de 10 subconjuntos, con la cual se clasifican correctamente el 91,7 % de los casos clasificados como no desertores y el 92,4 % de los reales no desertores (total de casos de no desertores utilizados en la validación). Para el caso de los desertores, se clasificaron correctamente el 69,7 % de los casos clasificados como desertores y el 67,6 % de los reales desertores.

Factor
Género
Promedio final en secundaria
Promedio final en secundaria superior
Tipo de admisión
Estudiante vive en un hostel
Tamaño de la familia del estudiante
Situación de padres (casados o divorciados)
Ingreso familiar
Calificaciones del padre
Calificaciones de la madre
Ocupación del padre
Ocupación de la madre

Tabla 2.2: Factores utilizados para predecir la deserción en [2].

En [11], se busca crear un modelo para predecir la deserción de estudiantes en el tercer semestre en los Estados Unidos. A diferencia de los otros trabajos, el problema de la predicción

de deserción de estudiantes se aborda como un problema de la clasificación no balanceada, es decir, existen más casos de estudiantes que no desertan que los que si lo hacen. Esto afecta del modelo predictivo el desempeño al clasificar la clase minoritaria (estudiantes que si desertan). Actualmente existen técnicas que son capaces de balancear los sets de datos, con el fin de mitigar este problema. El estudio, se realizó sobre una muestra de 21654 estudiantes nuevos, en los cuales se presentó una deserción del 21,3 % en el tercer semestre. En este trabajo, se ocupan 3 técnicas de balanceo distintas *random under-sampling* (se excluyen aleatoriamente casos de la clase mayoritaria), *random over-sampling* (se replican casos existentes de la clase minoritaria) y *synthetic minority over-sampling technique* (se crean nuevos casos de la clase minoritaria a partir de las características de la clase minoritaria existente), combinadas con varias técnicas de clasificación, regresión logística, árboles de decisión, redes neuronales artificiales y *support vector machine*. Los factores utilizados en este trabajo son los mismos que los utilizados en el trabajo de Dursun Delen (2011) [1], los cuales se pueden observar en la tabla 2.1. La validación de los modelos, fue realizada mediante una validación cruzada de 10 subconjuntos y el mejor modelo construido fue la combinación de las técnicas *synthetic minority over-sampling technique* para el balanceo de la muestra y *support vector machine* como técnica de clasificación, clasificando correctamente el 95,4 % de los casos clasificados como no desertores y el 84,9 % de los casos reales de no desertores. Para el caso de los desertores se clasificaron correctamente el 86 % de los casos clasificados como desertores y el 95,8 % de los casos totales de desertores.

Gerben W. Dekker *et al.* [12] construyeron modelos para predecir si un estudiante de Ingeniería Eléctrica de Eindhoven University of Technology obtendría su primer grado (que se entrega después de haber aprobado el primer año por completo) en un periodo de tres años, lo cual de no cumplirse estaría ligado fuertemente a la deserción. En este departamento existe una tasa del 40 % de deserción en estudiantes nuevos. Para la clasificación se construyeron modelos utilizando varias técnicas de clasificación, árboles de decisión con distintos algoritmos (CART y C4.5), red bayesiana, regresión logística, clasificador en base a reglas (JRip) y *random forest*. Los factores utilizados en los modelos son del tipo de pre ingreso (Información del estudiante previo al ingreso a la universidad) y de desempeño académico en la universidad. En los resultados se muestra que los modelos construidos solo con la información del desempeño académico en la universidad poseen un desempeño casi idéntico que a

los construidos con ambos grupos de factores. El mejor modelo construido fue aquel, en el cual se utilizó el algoritmo C4.5 (árbol de decisión), clasificando correctamente el 80 % de los casos para ambas clases (obtienen o no el su grado en un periodo 3 años).

2.0.1. Deserción en Chile y en UTFSM

El problema de la deserción en las instituciones universitarias en Chile es una realidad, según el Servicio de Información de Educación Superior, la tasa promedio de deserción en el primer año para las instituciones acreditadas es del 22,44 %, para los cohortes entre los años 2012 y 2016 [13]. Para el caso de las carreras de Ingeniería Civil impartidas en universidades pertenecientes al Consejo de Rectores, la tasa de deserción en el primer año se sitúa en el 24,3 % y aumentaría a 37,6 % en el segundo año académico [4].

En el trabajo [4] se mencionan 3 factores de la deserción en primer año en universidades en Chile, los cuales fueron determinados a través de entrevistas presenciales con directivos universitarios y encuestas Web. El primer factor mencionado es la falta de claridad vocacional de los estudiantes, la cual contempla dos causas. La primera que el estudiante no haya quedado en la carrera de su preferencia, debido a insuficiencia de puntaje o debido a privilegiar una institución por sobre la carrera preferida. La segunda es que el estudiante no haya tenido acceso a información y la orientación necesaria acerca de la carrera o universidad a la que entró, lo que puede generar una incongruencia entre el estudiante y la carrera o universidad. Otro factor mencionado es la situación económica, en especial para los estudiantes que deben costear sus gastos trabajando, los cuales se deben enfrentar al problema de tener que elegir entre destinarle tiempo a trabajar o a estudiar. El último factor mencionado es el rendimiento académico, lo que se puede deber a varios determinantes, como a una baja motivación debido a problemas vocacionales, debilidades académicas previas a la carrera, dificultades en metodologías de enseñanza y aprendizaje (debido a la transición del colegio a la universidad) y estar insatisfecho con la carrera.

En el trabajo de Sergio Celis *et al.*, se desarrolla un modelo para predecir la doble reprobación de un curso en estudiantes de primer año de Plan Común de Ingeniería y Ciencias de la Universidad de Chile, en donde existe una tasa de deserción del 5 % en el primer año

académico. Con la doble reprobación de un curso, la cual ocurre al final del segundo semestre del primer año, un estudiante entra en causal de eliminación, lo que afecta negativamente a las tasas de retención de primer año. La tasa de esta doble reprobación, es del 17 % en promedio entre los años 2010 y 2014. Para la creación del modelo se utilizó la información personal y académica disponible, la cual se separa en 3 grupos:

- **Características individuales:** Género, tiempo desde el egreso de enseñanza media y región de proveniencia.
- **Factores de pre ingreso:** Tipo de establecimiento de enseñanza media, experiencias previas en educación superior, puntajes PSU, vía de ingreso, ranking de notas y promedio en la enseñanza media.
- **Factores de rendimiento académico:** Ratio de créditos aprobados versus reprobados, variación de notas del primer al segundo semestre tanto en ramos aprobados como reprobados, y diferencias con la nota mínima de aprobación.

Para la selección de factores se utilizaron dos enfoques, el primero *forward feature selection*, en donde se parte sin ningún factor en el modelo y se va agregando el factor que mejora en mayor medida cierta métrica, hasta el punto que no queden factores que mejoren el desempeño. Posterior al *forward feature selection* se utilizó *backward elimination* en donde se empieza con los factores seleccionados con el enfoque anterior y se van eliminando los factores que al ser eliminados mejoren el desempeño de cierta métrica. Los factores que fueron seleccionados son:

- Género.
- Tipo de establecimiento de enseñanza media.
- Ratio de créditos reprobados.
- Promedios de controles 1 del segundo semestre menores al promedio final del primer semestre en cursos reprobados.

- Diferencia con nota mínima del peor promedio actual en cursos reprobados en el segundo semestre.
- Promedios de controles 1 del segundo semestre menores al promedio final del primer semestre en cursos no reprobados.
- Peor promedio de controles 1 del segundo semestre menores que el peor promedio final del primer semestre en cursos no reprobados.

Como técnica de clasificación se utilizó una regresión logística, la cual fue validada con los estudiantes del último año del set de datos (2014), en sus resultados se detalla que el 86 % de los estudiantes sin doble reprobación fueron clasificados correctamente y el 37,5 % de los estudiantes clasificados que tendrán doble reprobación fueron clasificados correctamente.

En el trabajo de título de Ignacio Ureta (2016) [3] se estudiaron los factores que inciden en la deserción en la UTFSM, utilizando la información que la universidad posee del estudiante después de la matrícula. En este trabajo se realizó una regresión lineal para estudiar las relaciones lineales que existen entre los diversos factores y la deserción e retención de estudiantes, en sus resultados se determinó que existen factores que inciden en la deserción y otros que inciden en la retención. En la tabla 2.3 se puede observar el ranking de factores que inciden en la retención, de mayor a menor son: el puntaje Nem, puntaje PSU de matemáticas, carrera del estudiante, puntaje PSU de lenguaje y si el estudiante se cambió de carrera. En la tabla 2.4 se detallan los factores que mejor explican la deserción de mayor a menor. El puntaje PSU de postulación a la UTFSM, puntaje PSU de ciencias, PSU de matemáticas, puntaje Nem y puntaje PSU de historia son los que más inciden en la deserción.

Ranking n°	Factor (Retención)
1	Puntaje NEM
2	Puntaje PSU de Matemáticas
3	Carrera
4	Puntaje PSU de Lenguaje y Comunicación
5	Cambio de Carrera
6	Puntaje PSU de Ciencias
7	Preferencia de carrera en postulación
8	Campus
9	Género
10	Región de origen
11	Educación de la Madre
12	Educación del Padre
13	Puntaje PSU de Historia, Geografía y Ciencias Sociales
14	Puntaje de postulación

Tabla 2.3: Ranking de factores que inciden en la retención en la UTFSM [3].

Ranking n°	Factor (Deserción)
1	Puntaje de postulación
2	Puntaje PSU de Ciencias
3	Puntaje PSU de Matemáticas
4	Puntaje NEM
5	Puntaje PSU de Historia, Geografía y Ciencias Sociales
6	Carrera
7	Puntaje PSU de Lenguaje y Comunicación
8	Región de origen
9	Educación de la madre
10	Género
11	Preferencia de carrera de postulación
12	Cambio de carrera
13	Educación del padre
14	Campus

Tabla 2.4: Ranking de factores que inciden en la deserción en la UTFSM [3].

2.1. Marco Teórico

2.1.1. Minería de datos

En [14] se define la minería de datos como el proceso de descubrir patrones interesantes y conocimientos a partir de grandes cantidades de datos. Los datos pueden provenir de bases de datos, almacenes de datos, la Web, otros repositorios de información o de una fuente en tiempo real.

Según [14] existen múltiples funcionalidades para la minería de datos, dentro de ellas se incluyen la discriminación y caracterización, minería de los patrones más frecuentes dentro de

las cuales están las asociaciones y correlaciones, clasificación y regresión, análisis de agrupaciones (clusters) y por último, análisis de valores atípicos (outliers). Las funcionalidades anteriormente mencionadas se pueden clasificar en dos categorías: la primera es minería descriptiva tiene como trabajo caracterizar propiedades de datos en un set de datos objetivos, y la segunda es la minería predictiva, cuyo trabajo es realizar una inducción en una fuente de datos para realizar predicciones.

Clasificación para análisis predictivo

Para este documento se utiliza la funcionalidad clasificación de la minería de datos, que en [14] se define como el proceso de encontrar un modelo (función o algoritmo) que describa y distinga las clases o conceptos de una fuente de datos. El modelo se deriva basado en un análisis de un set de entrenamiento (set de datos en el que se conoce la etiqueta de la clase de los datos). El modelo después es utilizado para predecir la etiqueta de la clase de los datos, en un set de datos en que dicha etiqueta no es conocida.

2.1.2. Metodología CRISP-DM

Una de las metodologías más utilizadas para realizar el proceso de minería de datos es la metodología CRISP-DM, según [15] consta de 6 fases, las cuales serán detalladas a continuación:

- **Comprensión del negocio:** Esta fase se focaliza en comprender los objetivos y requerimientos del proyecto desde una perspectiva del negocio, y después convertir este conocimiento en un problema de minería de datos por definición; además se detalla un plan de diseño preliminar para alcanzar los objetivos.
- **Comprensión de datos:** En esta fase se comienza con la recolección inicial de datos y se procede a realizar actividades para familiarizarse con ellos; se identifican además la calidad de los datos; también se descubren las primeras pistas en los datos, o detectar muestras interesantes de datos para formar hipótesis a partir de información oculta.

- **Preparación de datos:** Esta fase cubre todas las actividades para construir el set de datos final (El set de datos que será utilizado en la herramienta de modelado) desde el set de datos inicial. Estas actividades se pueden realizar múltiples veces, y no necesariamente en un orden determinado. Dentro de las actividades se contemplan la selección de tablas, registros y atributos, además de la limpieza de datos para la herramienta de modelado.
- **Modelado:** En esta fase se seleccionan y aplican varias técnicas de modelado, y sus parámetros son calibrados a valores óptimos. Normalmente, existen varias técnicas de modelado para un mismo problema de minería de datos.
- **Evaluación:** En esta etapa del proyecto ya se tiene construido un modelo (o modelos) que aparentan ser de gran calidad, desde la perspectiva de un análisis de datos. Antes de proceder a la fase de implementación del modelo, es de suma importancia evaluar el modelo, y revisar las etapas de la construcción del modelo, para estar seguros de que cumple con los objetivos del negocio. En el final de esta etapa se debe tomar una decisión en el uso de los resultados finales del proyecto de minería de datos.
- **Implementación:** Normalmente la creación de un modelo no es el final de un proyecto. Si el propósito del proyecto era incrementar el conocimiento de los datos, el conocimiento ganado deberá ser organizado y presentado en la forma de que el cliente lo pueda utilizar. Dependiendo de los requerimientos, la fase de implementación puede ser tan simple como generar un reporte o tan complejo como la implementación de un proceso de minería de datos. En muchos casos será el cliente y no el analista de datos, quien estará a cargo de las fases de implementación. Es importante que el analista de datos entienda al cliente las acciones que deberán llevarse a cabo para utilizar los modelos creados.

En la figura 2.1 se pueden observar gráficamente las etapas del proceso CRISP-DM.

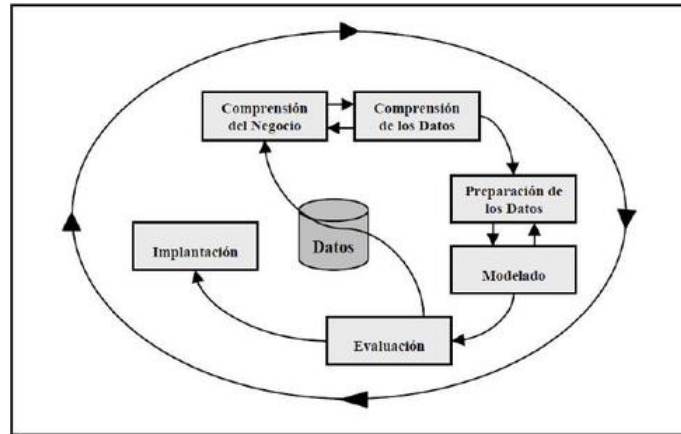


Figura 2.1: Etapas del proceso CRISP-DM [15].

Capítulo 3

Propuesta de solución

3.1. Comprensión del negocio

La deserción en la UTFSM se encuentra por debajo de los promedios nacionales para las carreras de Ingeniería Civil en universidades pertenecientes al Consejo de Rectores. Aun así estas cifras de retención pueden mejorar más. Una forma de lograr esto es poder aplicar medidas a los grupos de estudiantes en riesgo de deserción, identificándolos antes de que sea demasiado tarde para aplicar estas medidas. En este trabajo se tendrá como objetivo construir modelos que permitan clasificar a estudiantes, entre estudiantes que no estén riesgo deserción y los que sí estén, y en el camino determinar los factores que más inciden en que un estudiante esté en riesgo de deserción.

3.2. Propuesta

Como propuesta de solución a la problemática definida, se realizará un proyecto de minería de datos utilizando la metodología CRISP-DM, en el cual se construirán modelos capaces de predecir si un estudiante desertará o no en el primer o segundo año académico, a partir de datos entregados por diferentes unidades de la UTFSM. Como herramienta de modelado se

utilizará la librería scikit-learn del lenguaje de programación Python[16], la cual es una de las librerías aprendizaje automático más utilizadas actualmente. La librería cuenta con una amplia variedad de algoritmos con implementaciones del estado del arte y que además está fuertemente integrada con el lenguaje Python.

3.3. Plan de proyecto

A continuación se mostrará de forma resumida las tareas que se realizarán en las fases pendientes de la metodología CRISP-DM.

Comprensión de datos

- Se detallarán las fuentes de datos, explicando sus estructuras y proveniencias.

Preparación de datos

- Se integrarán todas las fuentes de datos, en una sola base de datos.
- Se crearán nuevos factores a partir de los datos integrados.
- Se explorarán los datos, con el fin de encontrar pistas ocultas en los datos.
- Se realizará la limpieza de datos producto de valores nulos.
- Se realizará la limpieza de datos producto de valores erróneos.
- Se realizará la limpieza de datos producto de las pistas encontradas en la exploración de los datos.
- Se cambiará el formato de algunos factores, de manera de adaptarlos a la herramienta de modelado.

Fase de Modelado

- Se mostrarán todos los algoritmos que se utilizarán para la construcción de los modelos.
- Se realizará una breve descripción de los algoritmos que se utilizarán.
- Se definirán las métricas necesarias para el plan de prueba para la evaluación de los modelos.
- Se definirá el criterio de desempeño de los modelos.
- Se realizará un proceso de selección de factores para ser utilizados en los diferentes modelos.
- Se realizará un proceso de ajuste de parámetros para cada uno de los diferentes modelos.

Fase de evaluación

- Se mostrarán los resultados de los modelos construidos.
- Se seleccionará el mejor modelo para cada una de las predicciones objetivas, según el criterio de desempeño.

Fase de implementación (Conclusión)

- Se mencionarán los comentarios acerca de la implementación de los modelos en la realidad.
- Se mencionarán los aspectos de destacar del trabajo.
- Se mencionarán los aspectos con posibilidades de mejora.

Capítulo 4

Implementación de la solución

4.1. Comprensión de los datos

4.1.1. Recolección y descripción de datos iniciales

Para este trabajo se utilizaron 4 fuentes de datos proporcionadas por diferentes unidades de la UTFSM, la primera contiene los datos referentes al pre ingreso a la universidad, la segunda y tercera contienen la información académica de los estudiantes, dentro de las cuales se encuentran las notas de primer año. Por último, se tiene la fuente que contiene la información referente a las becas de los estudiantes. Cada fuente de datos posee registros por estudiantes identificados por sus respectivos identificadores (Rut o identificador dependiendo el caso).

La Dirección de Admisión, proporcionó los datos referentes al pre ingreso a la universidad. Los nombres y descripciones de los factores que contienen los datos de la fuente antes mencionada, se pueden observar en la tabla 4.1.

Factor	Descripción
Carrera	Carrera de admisión del estudiante.
Sede	Sede de la UTFSM a la cual ingresó el estudiante.
Género	Género del estudiante, valor 1 para masculino y 2 para femenino.
Fecha de nacimiento	Fecha de nacimiento del estudiante.
Tramo de ingreso bruto familiar	Tramo de ingreso bruto familiar de la familia del estudiante, en donde 1 es el más bajo y 12 el más alto.
Cobertura de salud	Cobertura de salud del estudiante, en donde el valor 1 corresponde a Fonasa, valor 2 a Isapre, valor 3 otro, valor 4 Dipreca y valor 5 Capredena.
Rama educacional del colegio	Rama educacional del colegio del estudiante, los valores se adjuntan en la tabla 5.5 en el anexo.
Tipo de dependencia del colegio	Tipo dependencia del colegio de enseñanza media del estudiante, en donde valor 1 corresponde a Particular Pagado, valor 2 a Particular Subvencionado y valor 3 Municipal.
Número de región del colegio	Número de región del colegio del estudiante.
Puntaje NEM	Puntaje NEM del estudiante.
Puntaje PSU de Lenguaje y Comunicación	Puntaje de postulación de PSU de Lenguaje y Comunicación del estudiante.
Puntaje PSU de Matemáticas	Puntaje de postulación de PSU de Matemáticas del estudiante.
Puntaje Ranking	Puntaje Ranking del estudiante.
Puntaje PSU de Historia, Geografía y Ciencias Sociales	Puntaje de postulación de PSU de Historia, Geografía y Ciencias Sociales del estudiante.
Puntaje PSU de Ciencias	Puntaje de postulación de PSU de Ciencias del estudiante.
Puntaje anterior de PSU de Lenguaje y Comunicación	Puntaje anterior de PSU de Lenguaje y Comunicación del estudiante.
Puntaje anterior de PSU de Matemáticas	Puntaje anterior de PSU de Matemáticas del estudiante.
Puntaje anterior de PSU de Historia, Geografía y Ciencias Sociales	Puntaje anterior de PSU de Historia, Geografía y Ciencias Sociales del estudiante.
Puntaje anterior de PSU de Ciencias	Puntaje anterior de PSU de Ciencias del estudiante.

Tabla 4.1: Nombres y descripciones de los factores de los datos proporcionados por Dirección de Admisión.

La Dirección de Estudios de la UTFSM proporcionó 2 planillas de datos, la primera contiene la información que se puede observar en la tabla 4.2. Se omiten los datos ya repetidos en la tabla 4.1. La segunda planilla de datos contiene las notas correspondientes a los cursos de primer año en la malla curricular, en cada registro se tiene la estructura que se puede observar en la tabla 4.3.

Factor	Descripción
Dirección familiar	Dirección de la residencia familiar del estudiante.
Dirección académica	Dirección de la residencia académica del estudiante.
Año del primer semestre	Año del primer semestre académico del estudiante.
Año del tercer semestre	Año del tercer semestre académico del estudiante.
Año del quinto semestre	Año del quinto semestre académico del estudiante.
Posee cursos inscritos en el primer semestre	Indica si el estudiante posee cursos inscritos en el primer semestre académico con el valor 1, en caso contrario 0.
Posee cursos inscritos en el tercer semestre	Indica si el estudiante posee cursos inscritos en el tercer semestre académico con el valor 1, en caso contrario 0.
Posee cursos inscritos en el quinto semestre	Indica si el estudiante posee cursos inscritos en el quinto semestre académico con el valor 1, en caso contrario 0.

Tabla 4.2: Nombres y descripciones de los factores de la primera fuente de datos proporcionada por Dirección de Estudios.

Factor	Descripción
Nombre del curso	Nombre del curso que cursó el estudiante.
Período del curso	Año y semestre en el cual el estudiante cursó el curso.
Sede del curso	Sede en la cual se impartió el curso cursado por el estudiante.
Paralelo del curso	Paralelo del curso cursado por el estudiante.
Nota final del curso	Nota final obtenida por el estudiante en el curso cursado.
Vez que se cursa el curso	Vez que el estudiante esta cursando el curso.

Tabla 4.3: Nombres y descripciones de los factores de las notas proporcionadas por Dirección de Estudios.

Por último, se tienen los datos entregados por Relaciones Estudiantiles, consistentes en los montos de los beneficios (becas, ayudas, etc) de los estudiantes en su primer año. Para cada cohorte se tenía un archivo que contenía las diversas ayudas en montos por cada estudiante de la respectiva cohorte, la estructura se mantiene en cierta medida entre los años 2009 y 2015. El 2016 entró a funcionar la gratuidad , por lo que para ese año se tenían 2 archivos, uno con las becas y beneficios diferentes a la gratuidad y otro con los estudiantes con la gratuidad.

4.2. Preparación de los datos

4.2.1. Integración de datos

Para este trabajo, los factores más importantes son los que se deben predecir, estos corresponden a los que indican si el estudiante posee cursos inscritos en el tercer o quinto semestre académico. En el caso de no tenerlos, se asumirá deserción por parte del estudiante, en los años correspondientes a aquellos semestres. Tomando lo anterior en cuenta, se empezará con el archivo referido en la tabla 4.2 como base para integrar las bases de datos.

Incorporación de archivo de notas

Como fue descrito en la sección 4.1.1, se posee un archivo que contiene las notas de los cursos de primer año. Por cada estudiante, pueden existir múltiples registros en el archivo de notas; estos registros se conectan al estudiante a través del Rut. Por cada registro de estudiante en el archivo base se buscan las notas correspondientes en el archivo de notas. Utilizando ambos archivos se crearon nuevos factores, correspondiente a la nota del curso. También se creó un nuevo factor que corresponde al ranking de la nota final del estudiante en el curso, considerando el año, semestre y campus para su creación, donde 1 es el mejor, 2 el segundo mejor y así. En el caso de compartir la misma nota se comparte el ranking.

Incorporación de archivo con información de pre ingreso

Al igual que en la sección 4.2.1, se integró la información correspondiente al pre ingreso a la UTFSM, identificando al estudiante por el Rut. En este proceso se dió cuenta de que existían estudiantes que figuraban en más de un registro, lo que se explicaba por un reingreso a la universidad. Se procedió a eliminar a estos estudiantes por el posible ruido que pudieran generar en los datos. También se eliminaron a aquellos estudiantes que no figuraban en el archivo debido a que no se contaría con sus factores de pre ingreso; esto se debe a un ingreso especial a la UTFSM por parte de los estudiantes.

Incorporación de archivos con información de beneficios

Como se mencionó en la sección 4.1.1 se tenía un archivo por cohorte, con excepción para el año 2016. Por cada archivo se extrajeron y sumaron todos los beneficios que poseía cada estudiante; luego se creó un nuevo factor correspondiente al porcentaje de beca del estudiante con respecto al arancel de referencia, utilizando el monto de beneficios y el arancel de referencia del respectivo año. Se realizó el mismo procedimiento anterior para la cohorte 2016, con la excepción del caso de los estudiantes con la gratuidad, a los cuales se les asignó un porcentaje de beca del 100 %.

4.2.2. Construcción de datos

En esta fase se hablará sobre los nuevos factores contruidos con la información ya existente. Como se mencionó anteriormente se construyó los factores correspondientes a los rankings de las notas de los estudiantes en sus cursos. También se mencionó que se construyó el factor correspondiente al porcentaje de beca del arancel del estudiante.

Otro factor construido fue la de la edad aproximada; para ello se utilizó el año del primer semestre del estudiante restando el año de su fecha de nacimiento.

También, se construyó un factor binario para indicar si el estudiante se trasladó de domicilio para asistir a la universidad; para ello se comparó la dirección familiar y la dirección académica (tabla 4.2), en el caso de que fueran iguales se le asignó un valor 0 al factor, en caso contrario se le asignó el valor 1.

Se construyeron otros 4 factores binarios para indicar si un estudiante aprobó en el primer semestre cada uno de los 4 cursos contemplados en este trabajo (Programación, Química y Sociedad, Matemáticas I y Introducción a la Física). Para ello se utilizaron los factores correspondientes a las notas de cada curso, en el caso de que la nota fuera mayor o igual a 55 (nota de aprobación) se asignó el valor 1 al respectivo factor de aprobación para dicho curso, en caso contrario se asignó un 0. Para el caso de las asignaturas de Programación y Química y Sociedad, el valor 0 podría representar que no se cursó el curso.

Por último, se construyeron 3 factores para indicar los 5 quintiles de los puntajes PSU de Lenguaje y Comunicación, PSU de Matemáticas y Puntaje Nem, no se consideraron los puntajes de ranking, PSU de Historia, Geografía y Ciencias Sociales y PSU de Ciencias debido a que existían muchos valores faltantes para estos factores. Para la confección de los quintiles de las PSU de Lenguaje y Comunicación y Matemáticas se consideró el mejor puntaje PSU entre el postulación y el anterior en el caso de que exista, el mejor quintil sería el 5 y el peor sería el 1. Los quintiles de PSU de matemáticas, PSU de lenguaje y Nem se pueden observar en la tabla 4.4.

Factor \ Quintil	1	2	3	4	5
Puntaje PSU de Lenguaje y Comunicación	[344, 591)	[591, 624)	[624, 653)	[653, 691)	[691, 841]
Puntaje PSU de Matemáticas	[244, 661)	[661, 688)	[688, 713)	[713, 745)	[745, 850]
Puntaje NEM	[414, 624)	[624, 661)	[661, 702)	[702, 731)	[731, 826]

Tabla 4.4: Quintiles de puntaje PSU de Lenguaje y Comunicación, puntaje PSU de Matemáticas y puntaje NEM.

4.2.3. Exploración de los datos

Una vez integradas las bases de datos quedaron 11156 registros de estudiantes que ingresaron a la UTFSM entre los años 2009 y 2016, pertenecientes a las carreras que se pueden observar a continuación:

- Ingeniería Civil
- Ingeniería Civil Ambiental
- Ingeniería Civil de Minas
- Ingeniería Civil Eléctrica
- Ingeniería Civil Electrónica
- Ingeniería Civil en Informática
- Ingeniería Civil Industrial
- Ingeniería Civil Matemática
- Ingeniería Civil Mecánica
- Ingeniería Civil Metalúrgica
- Ingeniería Civil Común

- Ingeniería Civil Química
- Ingeniería Civil Telemática

En este estudio inicialmente se tenía la sospecha, fundamentada con la literatura, que los factores que más incidían en la deserción son los que correspondan a los referentes al desempeño académico, el cual es medido mediante las notas finales de los cursos, sus respectivos rankings y número de cursos reprobados.

Dentro de los cursos de primer año, existen dos que los estudiantes de las carreras de Ingeniería Civil tendrán que dar el primer semestre a diferencia de los otros dos cursos que varían según la malla, estos son los cursos de Introducción a la Física y Matemáticas I. En la figura 4.1 se puede observar las tasas de reprobación en el curso de Introducción a la Física por año de ingreso, lo mismo para el curso de Matemáticas I en la figura 4.2. Al observar las figuras se puede ver que en los años 2011 y 2015 disminuyen drásticamente las tasas de reprobación. Esto se debe a que en esos años el estudiantado de la UTFSM se mantuvo paralizado por un largo tiempo. Debido a las repercusiones de estas largas paralizaciones, se dio la posibilidad de des-inscribir cursos, lo que no se puede hacer con cursos de primer año en situaciones normales.

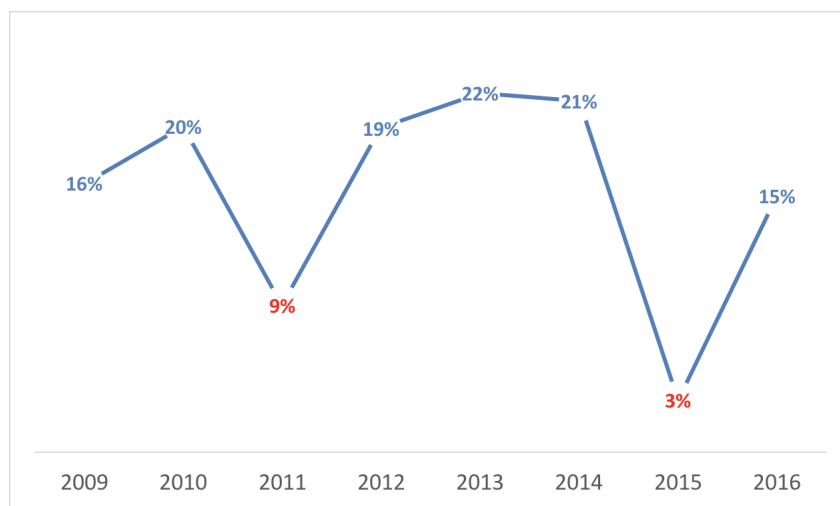


Figura 4.1: Tasas de reprobación en el curso de Introducción a la Física por cohorte.

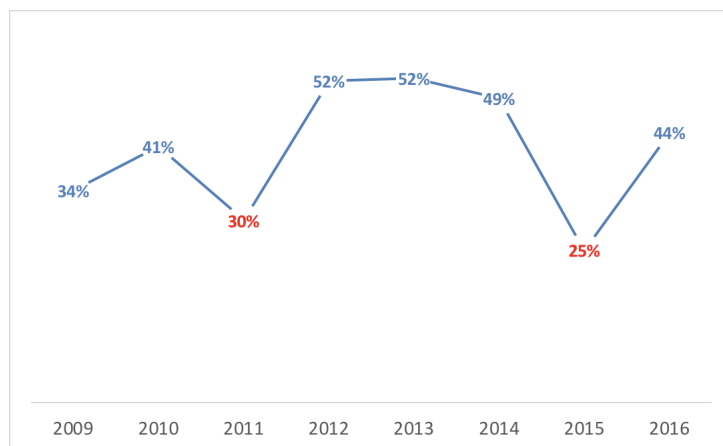


Figura 4.2: Tasas de reprobación en el curso de Matemáticas por cohorte.

En cuanto al número de cursos reprobados en el primer semestre, en la tabla 4.5 se pueden observar los porcentajes de retención y deserción en el primer año académico, según el número de cursos reprobados en el primer semestre. Se puede ver que a medida de que esté número aumenta, aumenta notablemente el porcentaje de deserción en el primer año. En la tabla 4.6 se puede observar lo mismo, pero para los porcentajes de deserción y retención en el segundo año académico. Analizando ambas tablas, se puede observar que el número de cursos reprobados en el primer semestre, será un factor predictivo de gran calidad, puesto a que en todos los casos existe un porcentaje mayor al 50 % para alguno de los dos posibles resultados (deserción o retención); esto se acentúa bastante en los extremos de las tablas. En el caso de la tabla 4.5 el 96 % de los estudiantes que no reprueban cursos en el primer semestre no desertan en el primer año y solo el 4 % si lo hace. Al segundo año (tabla 4.6), el porcentaje de estudiantes sin cursos reprobados en el primer semestre que siguen estudiando es de un 88 %. Se puede observar que el 90 % de los estudiantes que reprueban 3 cursos desertaron al cabo del segundo año y solo el 10 % no desertó.

Número de cursos reprobados en el primer semestre	Porcentaje de deserción	Porcentaje de retención
0	4 %	96 %
1	14 %	86 %
2	31 %	69 %
3	67 %	33 %

Tabla 4.5: Porcentajes de deserción y retención en el primer año, según el número de cursos reprobados en el primer semestre.

Número de cursos reprobados en el primer semestre	Porcentaje de deserción	Porcentaje de retención
0	12 %	88 %
1	34 %	66 %
2	57 %	43 %
3	90 %	10 %

Tabla 4.6: Porcentajes de deserción y retención en el segundo año, según el número de cursos reprobados en el primer semestre.

4.2.4. Limpieza de datos

En esta sección se hablará sobre la limpieza de datos, ya sea la eliminación de registros o factores. Para este trabajo se decidió eliminar los registros de estudiantes correspondientes a los años 2011 y 2015. En la sección 4.2.3, se muestran las tasas de reprobación de los cursos de Matemáticas I y Introducción a la Física afectadas por la posibilidad de des-inscribir asignaturas en el primer semestre académico. Esta eliminación, deja un total de 8594 registros. También en la sección 4.2.1, referente a la integración de los datos de pre ingreso, se mencionó que se eliminaron aquellos estudiantes que tuvieran asociado más de un registro

en los datos o que no tuvieran ninguno asociado.

Tratamiento de valores nulos

La herramienta que se utilizará para construir los modelos predictivos no soporta valores nulos, por lo que se pueden tomar dos medidas: la primera consiste en la eliminación de los registros que posean valores nulos para cierto factor, la segunda consiste en la eliminación del factor que posea valores nulos.

Para el primer caso descrito se eliminaron tan solo 4 registros que contenían valores nulos para ciertos factores.

Para el segundo caso se tienen 7 factores que fueron eliminados debido a la gran cantidad de valores nulos presentes en ellas. Los primeras dos corresponden a los factores de notas y rankings de los cursos de Programación y Química y sociedad, como se explicó anteriormente, estos cursos son dictados en el primer semestre dependiendo la malla curricular de la carrera del estudiante, solo existen 40 registros de estudiantes que dieron ambos cursos en el primer semestre, por lo que existen una gran cantidad de valores nulos para ambos factores, de conservarlas se deberá borrar aproximadamente el 9,4 % de los registros, por lo que se decidió borrar ambos factores.

También para el segundo caso se eliminaron los factores correspondientes al puntaje PSU de Historia, Geografía y Ciencias Sociales, puntaje PSU de Ciencias y puntaje Ranking, que poseían más del 50 % de sus valores nulos. Después de ésta limpieza quedaron 8590 registros.

Tratamiento de valores erróneos

Para el caso de los factores de cobertura de salud, tipo de dependencia del colegio, número de región del colegio, puntaje NEM y puntaje PSU de Lenguaje y Comunicación, existían registros que poseían valores iguales a 0 para aquellos factores, lo que debido a sus características no era posible, por lo que se eliminaron aquellos registros, quedando un total de

8569 registros.

Eliminación de factores

Por último, se eliminaron los factores correspondientes a las notas de Matemáticas I y Introducción a la Física, debido a que en algunos años se realizaron pruebas distintas para los diferentes paralelos dictados para un mismo curso, por lo que no serían comparables las notas de estudiantes de diferentes paralelos en un mismo año. La información de dichos factores es utilizada en los factores de ranking para ambos cursos. Estos factores si permiten comparar el desempeño de los estudiantes en un mismo año.

4.2.5. Cambio de formato de datos

Escalamiento de factores continuos

El algoritmo *support vector machine* requiere que los factores de tipo numéricos y continuos se encuentren en una escala en torno al 0; para ello se utilizó una técnica de escalamiento que remueve la media de los factores y los escala en un rango entre -1 y 1. Los factores a los cuales se les aplicó este procedimiento se pueden observar en 4.7.

Factor
Ranking en curso Introducción a la Física
Ranking en curso Matemáticas I
Edad
Puntaje NEM
Puntaje PSU de Lenguaje y Comunicación
Puntaje PSU de Matemáticas
Porcentaje de beca
Número de cursos reprobados
Quintil de puntaje PSU de Matemáticas
Quintil de puntaje PSU de Lenguaje y Comunicación
Quintil de puntaje Nem

Tabla 4.7: Factores escalados para ser utilizados en el algoritmo *support vector machine*.

Codificación de factores categóricos

La herramienta utilizada no acepta valores en el formato de texto, por lo que se deben codificar los factores en este formato utilizando números, los factores que deben ser codificados son la carrera del estudiante y la rama educacional del colegio.

Los factores anteriormente codificados, en adición de otros, no pueden ser utilizados directamente en los algoritmos clasificadores, puesto a que considerarían que los factores poseen un orden lógico (que se encuentran ordenados de menor a mayor o vice versa). La solución seleccionada para esta problemática es una codificación *One-Hot*[17], la cual consiste en crear un nuevo factor binario por cada uno de los valores de los factores de tipo categóricos, si el valor se encuentra presente en el registro se asignará el valor 1 y en caso contrario el valor 0. Los factores a los cuales hay que aplicarles este procedimiento se pueden observar en la tabla 4.8.

Factor
Carrera del estudiante
Rama educacional del colegio
Número de región del colegio
Tipo de dependencia del colegio
Género del estudiante
Cobertura de salud del estudiante

Tabla 4.8: Factores codificados con codificación *One-Hot*.

4.3. Fase de Modelado

En esta sección se hablará sobre los algoritmos de clasificación a utilizar, la selección de factores y el ajuste de parámetros.

4.3.1. Selección de técnicas de modelado

Para este problema de clasificación se seleccionaron varios algoritmos de clasificación, con el objetivo de poder comparar los resultados y seleccionar el con mejor desempeño según un plan de prueba. Las técnicas seleccionadas fueron escogidas mediante a reiterativas iteraciones de los modelos a construir, además de lo visto en la literatura. Los algoritmos de clasificación a utilizar son *random forest*, *k-nearest neighbors*, *support vector machine* con kernel lineal y *gradient boosting*.

- ***Random forest***: Es un algoritmo de clasificación desarrollado por Leo Breiman, que pertenece a la familia de los métodos ensemble en la Minería de datos, los cuales hacen referencia a los métodos que utilizan la combinación de varios modelos para obtener un mejor resultado. Este algoritmo funciona a partir de varios árboles de decisión que son contruidos con muestras aleatorias del set de entrenamiento, y que a diferencia de los árboles de decisión tradicionales, en vez de elegir la mejor variable para realizar

la separación en los nodos, se utiliza la mejor variable de un set de variables candidatas que son seleccionadas de forma aleatoria del total de variables del set de datos [18]. Este algoritmo de clasificación utiliza el método *bagging*, que hace referencia a la creación de modelos independientes, los cuales son entrenados con muestras del mismo tamaño (con repetición) del set de datos de entrenamiento. Una vez contruidos los árboles de decisión, para realizar una clasificación, a diferencia del algoritmo original, la herramienta utilizada promedia las probabilidades de clasificación de todos los árboles de decisión, la cual es utilizada como la probabilidad de clasificación.

- ***K-nearest neighbors***: En este algoritmo de clasificación se utilizan los vecinos más cercanos para determinar la clase de un caso a clasificar. El algoritmo cuenta de dos partes, la primera es determinar los K vecinos más cercanos al caso de clasificación, para ello se utilizan las características (variables) del caso, las cuales son normalizadas en un rango entre 0 y 1, y luego utilizando una métrica de distancia se encuentran sus K vecinos más cercano, basado en la cercanía a las características normalizadas. La segunda parte consta de la votación de los K vecinos más cercanos, para determinar la clase que se le asignará al caso de clasificación, para ello se realiza la votación considerando el peso de la distancia de cada vecino (entre más cercano, más peso tiene su votación), y se asigna la clase más votada [19].
- ***Support vector machine***: Es un algoritmo perteneciente a la familia de algoritmos lineales generalizados. Este algoritmo clasifica mediante la construcción de un hiperplano, en el cual se proyectan los casos de entrenamiento (los cuales son considerados vectores en el hiperplano). El hiperplano se divide en dos partes de tal forma que todos los casos de entrenamiento, se encuentren a la mayor distancia posible del límite de separación y que cada parte corresponda a una clase de clasificación (de los dos casos posibles). Los puntos más cercanos a la separación son llamados vectores de soporte y el criterio de clasificación corresponde a la parte que se proyecte el vector del caso a clasificar, y entre más distante se encuentre del límite de separación mejor será la probabilidad de clasificación de la clase a clasificar [20].
- ***Gradient boosting***: Este algoritmo de clasificación pertenece a la familia de métodos ensemble al igual que *random forest*, la principal diferencia con respecto al algoritmo

random forest es que en vez de utilizar el método *bagging* para combinar árboles de decisión se utiliza el método *boosting*. En el método *boosting* los modelos no son creados de forma independiente a diferencia de *bagging*; en el caso de *boosting* se van creando nuevos modelos, los cuales son influenciados por los modelos anteriores y modificados de forma de que sean expertos en los casos que los modelos anteriores no pudieron clasificar bien. Otra diferencia es que *boosting* a diferencia de *bagging* le asigna pesos a la contribución de los modelos anteriores a los nuevos según sus desempeños [21].

4.3.2. Generación de plan de prueba

Con el fin de poder demostrar la validez y calidad de los modelos, se definirán ciertos conceptos y métricas a continuación.

- **Verdaderos Positivos (VP):** Número de estudiantes que no desertan en el primer o segundo año académico (dependiendo el caso) y que son clasificados no desertores por el modelo predictivo.
- **Falsos Positivos (FP):** Número de estudiantes que si desertan en el primer o segundo año académico (dependiendo el caso) y que son clasificados como no desertores por el modelo predictivo.
- **Verdaderos Negativos (VN):** Número de estudiantes que si desertan en el primer o segundo año académico (dependiendo el caso) y que son clasificados como desertores por el modelo predictivo.
- **Falsos Negativos (FN):** Número de estudiantes que no desertan en el primer o segundo año académico (dependiendo el caso) y que son clasificados como desertores por el modelo predictivo.

Métricas de evaluación

- **Accuracy:** Esta métrica corresponde a la razón entre las predicciones correctas y el

total de predicciones, y se encuentra descrita a través de la ecuación 4.1.

$$\frac{(VP + VN)}{(VP + VN + FP + FN)} \quad (4.1)$$

- **Precision:** Esta métrica corresponde a la razón entre las predicciones correctas de una clase y el total de predicciones de dicha clase. En este caso se tienen dos tipos de clases, los estudiantes que desertan y los estudiantes que no desertan, para el caso de los estudiantes que no desertan la métrica de precisión (*precision+*) se describe a través de la ecuación 4.2.

$$\frac{(VP)}{(VP + FP)} \quad (4.2)$$

Para el caso de los estudiantes que si desertan la métrica de *precision* (Precision-) se encuentra descrita por la ecuación 4.3

$$\frac{(VN)}{(VN + FN)} \quad (4.3)$$

- **Recall:** Esta métrica corresponde a la razón entre las predicciones correctas de una clase y el total de casos de dicha clase. Para el caso de la clase de estudiantes que no desertan la métrica *recall* se describe por la ecuación 4.4.

$$\frac{(VP)}{(VP + FN)} \quad (4.4)$$

- **F1 Score:** Esta métrica corresponde a la media armónica entre las métricas *precision* y *recall*, la cual se encuentra descrita por la ecuación 4.5.

$$\frac{(2 + Precision + Recall)}{(Recall + Precision)} \quad (4.5)$$

- **Macro F1 Score:** Esta métrica corresponde al promedio de las métricas *F1 Score* de las clases, esta métrica es utilizada con el objetivo de optimizar los modelos en cuanto a las métricas *precision* y *recall*, en las fases de selección de variables y de parámetros.

Criterio de desempeño de modelos

Para selección del mejor modelo construido, se utilizará la métrica *F1 Score*, debido a que esta métrica es más completa que las demás mencionadas al momento de medir el desempeño de modelos de clasificación, esto se debe a que al utilizarla se contemplan dos métricas muy utilizadas en la literatura, *precision* y *recall*.

Separación de set de datos entre entrenamiento y validación

Para la construcción y evaluación de los modelos se debe separar el set de datos en 2 partes, la primera corresponde al set de datos utilizado para entrenar los modelos (set de entrenamiento), y la segunda parte se utiliza para validar los modelos (set de validación). Estos sets de datos no pueden compartir registros debido a errores al momento de evaluar el desempeño de los modelos. Para el set de validación se escogió utilizar los registros de estudiantes cuyos cohortes sean las más actuales, debido a que sería el caso real de utilización de estos modelos. Según lo anterior se utilizarán los registros de los años de ingreso 2014 y 2016 para la validación de los modelos que predicen la deserción en el primer año, y los registros de los años de ingreso 2013 y 2014 para la validación de los modelos que predicen la deserción en el segundo año. Para entrenar los modelos que predicen la deserción en el primer año, se utilizarán los registros de los años de ingreso 2009, 2010, 2012 y 2013, y para los modelos que predicen la deserción en el segundo año, se utilizarán los registros de los años de ingreso 2009, 2010 y 2012.

4.3.3. Selección de factores

Con el fin de disminuir el posible ruido existente que pueden generar ciertos factores, se realizó un proceso de selección. Este proceso consta de dos partes, la primera es determinar la dependencia entre cada una de las factores existentes y los factores a predecir por los modelos (deserción en el primer o segundo año académico), y la segunda parte consiste en elegir el mejor percentil de factores a utilizar en cuanto a la dependencia existente.

Dependencia de factores

Para determinar la dependencia de los factores con los factores objetivos (deserción en el primer o segundo año académico) se utilizó un algoritmo llamado información mutua (*mutual information*), el cual es capaz de determinar cualquier tipo de dependencia (lineal, cuadrática, etc.) entre dos factores [22]. Para explicar el grado de dependencia entre dos factores se utiliza un puntaje que va de 0 cuando no existe dependencia a 1 donde existe mucha dependencia. En la tabla 4.9 se puede observar el ranking de factores con más dependencia con el factor que indica deserción o retención en el primer año. Para el caso anterior se puede observar que los factores con más dependencia con la deserción o retención en el primer año son los factores de desempeño académico. En el primer lugar se encuentra el número de cursos reprobados en el primer semestre, luego se encuentra el ranking en el curso de Matemáticas I, seguido por el factor binario que indica si el estudiante aprobó Química y Sociedad; después se encuentra el factor que indica la aprobación del curso de Programación, luego se encuentra el ranking en curso de Introducción a la Física, y por último, se tienen los factores correspondientes aprobaciones de los cursos Matemáticas I y Introducción a la Física en el sexto y séptimo lugar respectivamente.

En la tabla 4.10 se observa lo mismo pero para la dependencia con la deserción y retención en el segundo año. En este caso el ranking de dependencia se ve también encabezado por los factores de desempeño académico, en mismo orden que para la dependencia de la retención y deserción en el primer año.

Factor	Puntaje
Número de cursos reprobados	0,099
Ranking en curso Matemáticas I	0,083
Aprobación de curso Química y Sociedad	0,073
Aprobación de curso de Programación	0,071
Ranking en curso Introducción a la Física	0,058
Aprobación de curso Matemáticas I	0,052
Aprobación de curso Introducción a la Física	0,048
Puntaje de PSU de Matemáticas	0,027
Porcentaje de beca	0,008
Puntaje de PSU de Lenguaje y Comunicación	0,006
Quintil de puntaje NEM	0,005
Estudiante pertenece a carrera Ingeniería Civil Industrial	0,005
Estudiante pertenece a carrera Ingeniería Civil en Informática	0,004
Quintil de puntaje de PSU de Matemáticas	0,004
Puntaje NEM	0,004
Edad	0,003
Colegio del estudiante se localiza en la III Región	0,003

Tabla 4.9: Top 20 de dependencia de factores con la deserción en el primer año académico.

Factor	Puntaje
Número de cursos reprobados	0,111
Ranking en curso Matemáticas I	0,092
Aprobación de curso Química y Sociedad	0,089
Aprobación de curso de Programación	0,075
Ranking en curso Introducción a la Física	0,073
Aprobación de curso Matemáticas I	0,059
Aprobación de curso Introducción a la Física	0,055
Puntaje de PSU de Matemáticas	0,021
Quintil de puntaje NEM	0,016
Porcentaje de beca	0,014
Puntaje NEM	0,008
Estudiante pertenece a carrera Ingeniería Civil Industrial	0,006
Estudiante pertenece a carrera Ingeniería Civil Telemática	0,005
Estudiante pertenece a carrera Ingeniería Civil en Informática	0,005
Puntaje de PSU de Lenguaje y Comunicación	0,005
Edad	0,004
Colegio del estudiante se localiza en la VIII Región	0,004

Tabla 4.10: Top 20 de dependencia de factores con la deserción en el segundo año académico.

Selección de mejor percentil de factores

Para la segunda parte de la selección de factores se procedió a escoger el mejor percentil de factores según los puntajes de dependencia, para cada uno de los modelos a construir. Este proceso se realizó mediante una validación cruzada con un esquema *k-fold* utilizando solo el set de entrenamiento. Con percentiles de 5 % en 5 % (hasta llegar al 100 %) se realizó el procedimiento de la figura 4.3. Por cada uno de los percentiles se construyeron modelos, los cuales fueron validados solo con registros de un determinado año de ingreso; el resto de los registros son utilizados para entrenamiento. Este proceso se repite por cada uno de los años de ingreso del set de entrenamiento. Una vez contruidos y validados todos los modelos para

un percentil, se genera valor promedio de la métrica *F1 Score Macro*, considerando todos los modelos construidos para ese percentil. Por último se utiliza el percentil con mejor valor para la métrica *F1 Score Macro*. Los percentiles utilizados para los modelos que predicen la deserción en el primer año académico se pueden observar en la tabla 4.11, para el caso de los modelos que predicen la deserción en el segundo año, se pueden ver en la tabla 4.12.

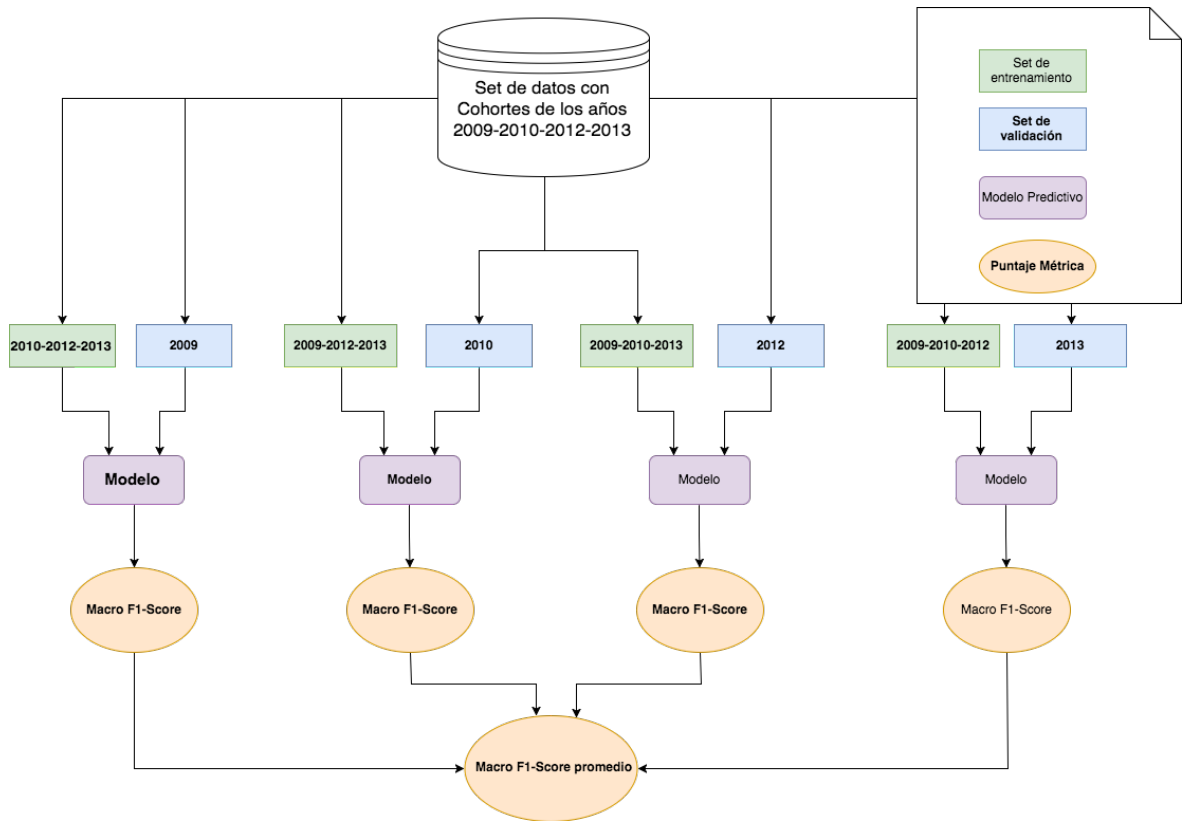


Figura 4.3: Esquema *k-fold* utilizado para la selección de percentil, para modelos de predicción de deserción en el primer año académico.

Técnica de clasificación	Percentil utilizado
<i>Random forest</i>	35 %
<i>K-nearest neighbors</i>	5 %
<i>support vector machine</i>	85 %
<i>Gradient boosting</i>	80 %

Tabla 4.11: Percentiles utilizados para modelos que predicen la deserción en el primer año académico.

Técnica de clasificación	Percentil utilizado
<i>Random forest</i>	85 %
<i>K-nearest neighbors</i>	10 %
<i>Support vector machine</i>	75 %
<i>Gradient boosting</i>	85 %

Tabla 4.12: Percentiles utilizados para modelos que predicen la deserción en el segundo año académico.

4.3.4. Ajuste de parámetros de los algoritmos

Para el ajuste de parámetros se utilizó el mismo esquema utilizado para la selección de percentil (como se muestra en la figura 4.3). Utilizando una amplia lista de diferentes combinaciones de parámetros (dependiendo del algoritmo de clasificación utilizado), se procedió a buscar la combinación de parámetros que tuviera un mejor *F1 Score Macro*. Los parámetros seleccionados para los diferentes modelos se pueden observar en las tablas 5.6, 5.7, 5.8 y 5.9.

Capítulo 5

Evaluación de la solución

Una vez realizadas todas las fases anteriores, se procede a evaluar el desempeño de los modelos. Se construyeron modelos utilizando solo registros del Campus Santiago y otros solo utilizando registros del Campus Casa Central, pero al final se obtuvieron mejores resultados utilizando todos los registros.

5.0.1. Evaluación de modelos que predicen la deserción en el primer año académico

En la tabla 5.1 se pueden observar los resultados con los diferentes algoritmos de clasificación, para la predicción de la deserción en el primer año académico. Las métricas construidas con los resultados se pueden observar en la tabla 5.2.

Considerando el criterio de desempeño (sección 4.3.2), el modelo con el mejor valor para la métrica *F1 Score* es el aquel construido con el algoritmo *random forest*, con un valor del 92,4 % (0,924). Adicionalmente, este modelo es el que posee mejores valores para las métricas *accuracy*, *precision-* y *recall*. En la única métrica que se ve superado este modelo por todos los restantes es en la *precision+*, el modelo con el mejor valor para esta métrica es el construido con el algoritmo *support vector machine*, y supera al modelo construido con *random forest* en un 8,6 % en la métrica *precision+*,.

Algoritmo de clasificación	VP	FP	VN	FN
<i>Random forest</i>	2185	253	215	88
<i>K-nearest neighbors</i>	2050	199	269	223
<i>Support vector machine</i>	1230	29	439	1043
<i>Gradient boosting</i>	2092	206	262	181

Tabla 5.1: Resultados de modelos que predicen la deserción en el primer año académico.

Algoritmo de clasificación	<i>Accuracy</i>	<i>Precision+</i>	<i>Precision-</i>	<i>Recall</i>	<i>F1 Score</i>
<i>Random forest</i>	0,869	0,891	0,710	0,960	0,924
<i>K-nearest neighbors</i>	0,846	0,912	0,547	0,902	0,907
<i>Support vector machine</i>	0,609	0,977	0,296	0,541	0,696
<i>Gradient boosting</i>	0,859	0,910	0,591	0,920	0,915

Tabla 5.2: Métricas de los resultados de modelos que predicen la deserción en el primer año académico.

5.0.2. Evaluación de modelos que predicen la deserción en el segundo año académico

En la tabla 5.3 se pueden observar los resultados con los diferentes algoritmos de clasificación, para la predicción de la deserción en el segundo año académico. Las métricas construidas con los resultados se pueden observar en la tabla 5.4.

En este caso el modelo con el mejor desempeño según la métrica *F1 Score* es aquel construido con el algoritmo *gradient boosting*, con un valor del 92,9 % (0,929). El mejor modelo seleccionado solo se ve superado en dos métricas por dos algoritmos distintos, la primera es la *precision+*, en la cual se ve superado por el modelo construido con algoritmo *support vector machines* en un 1,1 %, la segunda métrica es el *recall* en donde se ve superado por un 1,5 % por el modelo construido con el algoritmo *random forest*.

Algoritmo de clasificación	VP	FP	VN	FN
<i>Random forest</i>	1629	491	501	78
<i>K-nearest neighbors</i>	1516	426	566	191
<i>Support vector machine</i>	1452	108	884	255
<i>Gradient boosting</i>	1603	120	872	104

Tabla 5.3: Resultados de modelos que predicen la deserción en el segundo año académico.

Algoritmo de clasificación	<i>Accuracy</i>	<i>Precision+</i>	<i>Precision-</i>	<i>Recall</i>	<i>F1 Score</i>
<i>Random forest</i>	0,789	0,768	0,865	0,954	0,851
<i>K-nearest neighbors</i>	0,771	0,781	0,748	0,888	0,831
<i>Support vector machine</i>	0,866	0,931	0,776	0,851	0,889
<i>Gradient Boosting</i>	0,910	0,920	0,893	0,939	0,929

Tabla 5.4: Métricas de los resultados de modelos que predicen la deserción en el segundo año académico.

Conclusiones

Con respecto al objetivo general de la memoria, se lograron construir modelos para predecir la deserción y retención en el primer y segundo año académico en las carreras de Ingeniería Civil de la UTFSM. Los modelos construidos resultaron ser de gran calidad al compararlos con los resultados de modelos similares en el estado del arte, además de superar con creces los modelos similares en Chile.

Para el caso del mejor modelo que predice la deserción y retención en el primer año académico, construido con el algoritmo *random forest*, se logró clasificar correctamente el 86,9 % de todos los casos (*accuracy*) , el 89,1 % de todos los casos clasificados como no desertores (*precision+*), el 71 % de los casos clasificados como desertores (*precision-*), el 96 % de todos los casos de no desertores (*recall*) y por último, se obtuvo un valor del 92,4 % para la métrica *F1 Score*.

Para la predicción de la deserción y retención en el segundo año académico, el mejor modelo fue aquel construido con el algoritmo *gradient boosting*, con el cual se clasificó correctamente el 91 % de todos los casos clasificados, el 92 % de los casos clasificados como no desertores, el 89,3 % de los casos clasificados como desertores, el 93,9 % del total de casos de no desertores y se obtuvo un valor del 92,9 % para la métrica *F1 Score*. Los resultados para este modelo son ligeramente mejores que el mejor modelo que predice la deserción en el primer año académico.

Con respecto al objetivo específico, el cual era identificar los factores que más inciden en la deserción y retención en el primer y segundo año académico, en la sección 4.3.3 se pudo

observar que los factores más importantes sin lugar a dudas son los referentes al desempeño académico en el primer semestre. Estos factores superaron en nivel de importancia al predecir la deserción y retención, a todos los otros factores referentes a la información del preingreso a la UTFSM. Para ambos casos de predicción se repitieron los 7 primeros factores, en mismo orden de importancia, el más importante fue el número de cursos reprobados en el primer semestre académico, lo que probó la hipótesis sobre su importancia en la sección 4.2.3, en donde se exploraron los datos en búsqueda de pistas. El segundo factor más importante resultó ser el ranking de la nota final del estudiante en el curso de Matemáticas I, por lo que esto sustenta la decisión de haber excluido los registros de cohortes cuyas tasas de reprobación se vieron disminuidas por las medidas tomadas por la UTFSM debido a las largas paralizaciones estudiantiles. El tercer factor más importante fue el factor binario que indica la aprobación del curso de Química y Sociedad en el primer semestre académico, seguido por el cuarto factor más importante correspondiente al factor que indica la aprobación del curso de Programación en el primer semestre académico. El quinto factor más importante fue el ranking de la nota final del estudiante en el curso de Introducción a la Física, seguido por el sexto y séptimo que corresponden a los factores que indican la aprobación en los cursos de Matemáticas I y Introducción a la Física respectivamente. En cuanto a los factores referentes al pre ingreso a la UTFSM, se comparte entre ambos casos que el más importante es el puntaje de PSU de Matemáticas. Para el caso de la importancia de factores en la predicción de la deserción y retención en el primer año académico, el factor que le sigue en importancia al puntaje de la PSU de Matemáticas, es el porcentaje de beca del estudiante con respecto al arancel de referencia, pero según el algoritmo utilizado para detectar la dependencia este factor es aproximadamente 3 veces menos importante que el puntaje PSU de Matemáticas (desde ese punto disminuye mucho el nivel de importancia de los factores). Para el caso de la deserción y retención en el segundo año académico, el factor que le sigue en importancia a el puntaje de la PSU de Matemáticas es el Quintil de puntaje NEM del estudiante, seguido por el porcentaje de beca del estudiante, en este caso no se ve una disminución tan notoria en el nivel de importancia en ambos factores con respecto al factor correspondiente puntaje de la PSU de Matemáticas.

En cuanto a la implementación real de los modelos en el futuro, se concluye que es indispensable tener un proceso de gran calidad para la obtención de los datos requeridos para la

construcción de los modelos. El trabajo y la forma de cómo se integraron las fuentes de datos, además de la creación de factores nuevos, no es lo suficientemente eficiente para efectuarlo año a año, para la predicción de la deserción de estudiantes nuevos en el primer y segundo año académico. La creación de un procedimiento de mayor calidad para la generación de los datos o la creación de una base de datos centralizada sería un muy buen punto de partida para la implementación real de los modelos, una vez realizado esto ,permitirá el desarrollo de este proyecto y de muchos otros posibles.

Como aspectos a destacar de este trabajo, se destaca el uso de la librería de modelado scikit-learn, la elección de esta librería fue el origen de la búsqueda de la mejor herramienta posible para este trabajo, sin importar la dificultad de aprendizaje que representaba utilizarla. La utilización de esta librería en adición de las múltiples librerías con las que cuenta el lenguaje de programación Python, permitió realizar un proyecto de gran calidad en cuanto a sus resultados.

En este trabajo se tuvo como objetivo específico encontrar el mejor algoritmo de clasificación para cada una las predicciones que se querían realizar (deserción en el primer y segundo año académico). Los cuatro algoritmos mostrados en este trabajo son el producto de reiterativas iteraciones de los modelos, en donde fueron utilizados bastantes más algoritmos de la librería utilizada (los cuales no se mostraron debido a que se consideró que era suficiente con los cuatro mostrados). Dentro de los algoritmos no utilizados se encuentran los pertenecientes a la familia Deep Learning, que en la actualidad son una de las familias de algoritmos con mejor desempeño, además de que Google se encuentra constantemente potenciando su librería llamada *TensorFlow*[23], que se especializa en este tipo de algoritmos y actualmente es la más utilizada para el desarrollo de proyectos de minería de datos de clasificación. Por lo que se ve un posible espacio para la mejora realizando el mismo proyecto con la librería *TensorFlow*.

Por último, con respecto a la metodología CRISP-DM, utilizada para la estructuración de este documento sobre un proyecto de minería de datos, Inicialmente se empezó este documento sin ninguna estructura en particular, lo que provoco que fuera confuso para el lector, por lo que se decidió utilizar la metodología CRISP-DM. Se concluye que fue de suma utilidad la utilización de la metodolología para ordenar el documento de una forma que se pudiera

transmitir al lector de una manera ordenada y concisa, este documento.

Bibliografía

- [1] Dursun Delen, “Predicting student attrition with data mining methods,” *Journal of College Student Retention*, vol. 13 (1), pp. 17–35, 2011.
- [2] Dr. Saurabh Pal, “Mining educational data using classification to decrease dropout rate of students,” *International Journal Of Multidisciplinary Sciences And Engineering*, vol. 3 (5), pp. 35–39, 2012.
- [3] Ignacio Ureta, “Determinación de factores de riesgo para la deserción académica en la UTFSM mediante análisis de supervivencia, Memoria de título de Ingeniero Civil en Informática, UTFSM,” 2016.
- [4] Centro de Microdatos, “Estudio sobre causas de la deserción universitaria,” 2008.
- [5] George Siemens, Ryan S J.d. Baker, “Learning analytics and educational data mining: towards communication and collaboration,” in *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge*, pp. 252–254, 2012.
- [6] Cristóbal Romero, Sebastián Ventura, “Educational data mining: A review of the state of the art,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 40 (6), pp. 601 – 618, 2010.
- [7] Vincent Tinto, *Leaving College: Rethinking the Causes and Cures of Student Attrition*. University of Chicago Press, 1987.
- [8] Rose M. Marra, Kelly A. Rodgers, Demei Shen, Barbara Bogue, “Leaving engineering: A multi-year single institution study,” *Journal of Engineering Education*, vol. 101, pp. 6–27, 2012.
- [9] Youngkyoung Min, Guilli Zhang, Russell A. Long, Timothy J. Anderson, Matthew W. Ohland, “Nonparametric survival analysis of the loss rate of undergraduate engineering students,” *Journal of Engineering Education*, vol. 100 (2), pp. 349–373, 2011.
- [10] Guilli Zhang, Timothy J. Anderson, Mathew W. Ohland, Brian R. Thorndyke, “Identifying factors influencing engineering student graduation: A longitudinal and cross-institutional study,” *Journal of Engineering Education*, vol. 93(4), pp. 313–320, 2004.

- [11] Dech Thammasiri, Dursun Delen, Phayung Meesad, Nihat Kasap, “A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition,” *Expert Systems with Applications*, vol. 41 (1), p. 321–330, 2014.
- [12] Gerben Dekker, Mykola Pechenizkiy, Jan Vleeshouwers, “Learning analytics and educational data mining: towards communication and collaboration,” in *Proceedings of the 2nd International Conference on Educational Data Mining.*, pp. 41–50, 2009.
- [13] Servicio de Información de Educación Superior, “Informe retención de 1er año de pregrado cohortes 2012-2016,” 2018.
- [14] Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2011.
- [15] Pete Chapman, Randy Kerber, Julian Clinton, Thomas Khabaza, Thomas Reinartz, Rüdiger Wirth, “The crisp-dm process model,” 1999.
- [16] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, “Scikit-learn: Machine learning in python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [17] Kedar Potdar, Taher S. Pardawala, Chinmay D. Pai, “A comparative study of categorical variable encoding techniques for neural network classifiers,” *International Journal of Computer Applications*, vol. 175(4), pp. 7–9, 2017.
- [18] Ramón Díaz Uriarte, Sara Alvarez de Andrés, “Gene selection and classification of microarray data using random forest,” *BMC Bioinformatics*, vol. 7 (3), 2006.
- [19] Pádraig Cunningham, Sarah Jane Delany, “k-nearest neighbour classifiers,” 2007.
- [20] Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, *An Introduction to Information Retrieval*. Cambridge University Press, 2009.
- [21] Ian H. Witten, Eibe Franki, *Data Mining, Practical Machine Learning Tools and Techniques*, vol. 2. Morgan Kaufmann, 2005.
- [22] Alexander Kraskov, Harald Stoeckbauer, Peter Grassberger, “Estimating mutual information,” *Phys. Rev.*, vol. 69, 2004.
- [23] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, Xiaoqiang Zheng, “Tensorflow: A system for large-scale machine learning,” in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation.*, pp. 265–283, 2016.

5.1. Anexos

Valor	Significado
H1	Humanista Científico Diurno.
H2	Humanista Científico Nocturno.
H3	Humanista Científico – Validación de estudios.
H4	Humanista Científico – Reconocimiento de estudios.
T1	Técnico Profesional Comercial.
T2	Técnico Profesional Industrial.
T3	Técnico Profesional Servicios y Técnica.
T4	Técnico Profesional Agrícola.
T5	Técnico Profesional Marítima.

Tabla 5.5: Valores y significados para el factor rama educacional del colegio del estudiante.

Factor a predecir/Parámetro	min_samples_leaf	min_samples_split	max_depth	n_estimators	random_state
Deserción en primer año	1	10	4	20	22
Deserción en segundo año	5	15	4	20	4

Tabla 5.6: Parámetros utilizados en los modelos construidos con el algoritmo de clasificación *random forest*.

Factor a predecir/Parámetro	n_neighbors	weights	algorithm	p
Deserción en primer año	8	uniform	ball_tree	1
Deserción en segundo año	8	uniform	ball_tree	1

Tabla 5.7: Parámetros utilizados en los modelos construidos con el algoritmo de clasificación *k-nearest neighbors*.

Factor a predecir/Parámetro	C	class_weight	random_state
Deserción en primer año	0,01	balanced	22
Deserción en segundo año	0,01	balanced	4

Tabla 5.8: Parámetros utilizados en los modelos construidos con el algoritmo de clasificación *support vector machine*.

Factor a predecir/Parámetro	learning_rate	min_samples_leaf	min_samples_split	max_depth	n_estimators	random_state
Deserción en primer año	0,1	10	2	2	200	25
Deserción en segundo año	0,1	15	25	4	200	19

Tabla 5.9: Parámetros utilizados en los modelos construidos con el algoritmo de clasificación *gradient boosting*.

5.1.1. Código en Python utilizado en este trabajo

```

1  #Librerías utilizadas.
2  import pandas as pd
3  from sklearn.metrics import accuracy_score, f1_score, precision_score,
    recall_score
4  semilla=42
5  import numpy as np
6  import json
7  from sklearn.preprocessing import StandardScaler, MinMaxScaler
8  from sklearn.svm import SVC
9  from sklearn.neighbors import KNeighborsClassifier
10 from sklearn.ensemble import RandomForestClassifier,
    GradientBoostingClassifier
11 from itertools import product
12 import statistics as st
13 import random
14 from sklearn.feature_selection import SelectPercentile,
    mutual_info_classif
15
16
17 #Función de filtrado de factores

```

```

18 def filtering_features(dataframe , diccionario , modelo_clasificador):
19     year_list=set(dataframe[ 'Cohorte' ])
20     best=0
21     best_k=0
22     for k in diccionario["percentile_list"]:
23         fl_array=[]
24         for i in year_list:
25             data_test=dataframe.loc[ dataframe[ 'Cohorte' ]==i ]
26             data_train=dataframe.loc[ dataframe[ 'Cohorte' ]!=i ]
27             y_test=data_test["target"]
28             y_train=data_train["target"]
29             selector=SelectPercentile(mutual_info_classif,k)
30             data_train_l=selector.fit_transform( data_train[ diccionario["
31                 columnas_a_estudiar" ]], y_train)
32             data_test_l=selector.transform( data_test[ diccionario["
33                 columnas_a_estudiar" ]])
34             if(diccionario["clasificador"] == "SVL" or diccionario["
35                 clasificador"] == "SVR"):
36                 support = list(selector.get_support())
37                 usadas = []
38                 for i in range(0, len(support)):
39                     if (support[i]):
40                         usadas.append(diccionario["columnas_a_estudiar"][
41                             i])
42             lista = list(diccionario["columnas_a_escalas"])
43             for i in lista:
44                 if (not i in usadas):
45                     lista.remove(i)
46             escalador = StandardScaler()
47             dataframe_aux_train=data_train.copy()
48             dataframe_aux_test=data_test.copy()
49             dataframe_aux_train[ lista ] = escalador.fit_transform(
50                 dataframe_aux_train[ lista ])
51             dataframe_aux_test[ lista ] = escalador.transform(
52                 dataframe_aux_test[ lista ])
53             y_test_svm = dataframe_aux_test["target"]
54             y_train_svm = dataframe_aux_train["target"]

```

```

49         modelo_clasificador.fit(dataframe_aux_train[usadas],
                                   y_train_svm)
50         prediction = modelo_clasificador.predict(
                                   dataframe_aux_test[usadas])
51         f1_array.append(f1_score(y_test_svm, prediction, average=
                                   'macro'))
52     else:
53         modelo_clasificador.fit(data_train_1, y_train)
54         prediction=modelo_clasificador.predict(data_test_1)
55         f1_array.append(f1_score(y_test, prediction, average = '
                                   macro'))
56     pond=st.mean(f1_array)
57     print(k, pond)
58     if(pond>best):
59         best_k=k
60         best=pond
61     print("best_k:", best_k)
62     selector = SelectPercentile(mutual_info_classif, best_k)
63     target = dataframe["target"]
64     selector.fit(dataframe[diccionario["columnas_a_estudiar"]], target)
65     columnas_no_utilizadas=[]
66     columnas_utilizadas = []
67     support =list(selector.get_support())
68     print(support)
69     for i in range(0, len(support)):
70         if(support[i]):
71             columnas_utilizadas.append(diccionario["columnas_a_estudiar"
72             ][i])
73         else:
74             columnas_no_utilizadas.append(diccionario["
75             columnas_a_estudiar"][i])
76     return columnas_utilizadas, columnas_no_utilizadas, list(selector.
77     scores_) , diccionario["columnas_a_estudiar"], best_k
78
79 #Función de generación de matriz de confusión
80 def confusion_matrix(vector_test, vector_prediction):
81     fp=0
82     fn=0

```

```

80     tn=0
81     tp=0
82     for j in range(0,len(vector_test)):
83         if (vector_test[j]==vector_prediction[j]):
84             if (vector_test[j]==1):
85                 tp+=1
86             else:
87                 tn+=1
88         else:
89             if (vector_test[j]==1):
90                 fn+=1
91             else:
92                 fp+=1
93     print("True_Positive: ",tp)
94     print("True_Negative: ",tn)
95     print("False_Positive: ",fp)
96     print("False_Negative: ",fn)
97     return tp,fp,tn,fn
98
99 #Función de selección de parámetros
100 def tuning_of_params(dataframe, args, model, params):
101     presicion_score_pond= args["presicion_score_pond"]
102     fl_score_pond= args["fl_score_pond"]
103     arguments=[]
104     productory_list=[]
105     dict_list=[]
106     random_sample=args["random_sample"]
107     for i in params.keys():
108         arguments.append(i)
109         productory_list.append(params[i])
110     argument_list=list(product(*productory_list))
111     print("Número_de_parámetros_totales: ",len(argument_list))
112     if(len(argument_list) > random_sample):
113         argument_list = random.sample(argument_list, random_sample)
114     else:
115         print('No_se_limitan_parametros')
116     for i in argument_list:
117         diccionario=dict()

```

```

118         for j in range(0, len(i)):
119             diccionario[arguments[j]] = i[j]
120             dict_list.append((diccionario, [], []))
121     lista_años = set(dataframe["Cohorte"])
122     for i in lista_años:
123         print("año: ", i)
124         data_train = dataframe.loc[dataframe["Cohorte"] != i]
125         data_test = dataframe.loc[dataframe["Cohorte"] == i]
126         trgt_train = data_train["target"]
127         trgt_test = data_test["target"]
128         count = 0
129         for j in range(0, len(dict_list)):
130             model.set_params(**dict_list[j][0])
131             model.fit(data_train[arguments["columnas_clasificacion"]],
132                       trgt_train)
133             prediction = model.predict(data_test[arguments["columnas_clasificacion"]])
134             dict_list[j][1].append(f1_score(trgt_test, prediction, average='macro'))
135             dict_list[j][2].append(precision_score(trgt_test, prediction))
136             count += 1
137             if (count % 100 == 0):
138                 print("LLeva: ", count)
139
140     best_score = 0
141     for j in dict_list:
142         mean_f1 = st.mean(j[1])
143         mean_presicion = st.mean(j[2])
144         score = mean_presicion * precision_score_pond + mean_f1 * f1_score_pond
145         if (score > best_score):
146             best_score = score
147             print("best_mean_j: ", mean_f1)
148             best = j[0]
149     print(best)
150     return best
151
152 #Función de codificación One-Hot
153 def onehot_encoder(df_train, df_val, columns):

```

```

153     onehot_columns=[]
154     print(list(df_train))
155     for i in columns:
156         values=list(set((list(df_train[i]))))
157         for j in values:
158             if(type(j) is str):
159                 df_train[str(i)+"_"+j+'_o']=0
160                 df_val[str(i)+"_"+j+'_o'] = 0
161                 df_train.loc[df_train[i]==j, str(i)+"_"+j+'_o']=1
162                 df_val.loc[df_val[i] == j, str(i)+"_"+j+'_o'] =1
163                 onehot_columns.append(str(i)+"_"+j+'_o')
164             elif(type(j) in [int, float, np.float64]):
165                 df_train[i+'_'+str(j)] = 0
166                 df_val[i+'_'+str(j)] = 0
167                 df_train.loc[df_train[i] == j, i+'_'+str(j)] = 1
168                 df_val.loc[df_val[i] == j, i+'_'+str(j)] = 1
169                 onehot_columns.append(i+'_'+str(j))
170             else:
171                 print(type(j))
172         del df_train[i]
173         del df_val[i]
174     return df_train, df_val, onehot_columns
175
176
177
178
179 def model_experimenter(argsi, param_grid, resultados_dataframe, index):
180     resultados=dict()
181     args=dict(argsi)
182     dataframe = pd.pandas.read_excel(args["direccion_BD"], 'Hoja_1')
183     #Selección de Campus
184     print(set(dataframe["Carrera"]))
185     if(args["campus"]=="S"):
186         dataframe = dataframe.loc[dataframe["Campus"] == "S"]
187     else:
188         if(args["campus"]=="V"):
189             dataframe = dataframe.loc[dataframe["Campus"] == "V"]
190         else:

```

```

191         print("All_campus")
192
193     #Selección de target objetivo y separación entre set de validación y
    entrenamiento
194     if (args["target"]=="S3"):
195         dataframe["target"]=dataframe["S3"]
196         dataframe_train = dataframe.loc[(dataframe["Cohorte"] != 2015) &
            (dataframe["Cohorte"] != 2016) & (dataframe["Cohorte"] !=
            2011) & (dataframe["Cohorte"] != 2014)]
197         dataframe_test = dataframe.loc[(dataframe["Cohorte"] == 2014) | (
            dataframe["Cohorte"] == 2016)]
198     else:
199         if (args["target"]=="S5"):
200             dataframe["target"] = dataframe["S5"]
201             dataframe_train = dataframe.loc[(dataframe["Cohorte"] !=
                2014) & (dataframe["Cohorte"] != 2015) & (dataframe["
                Cohorte"] != 2016) & (dataframe["Cohorte"] != 2011) & (
                dataframe["Cohorte"] != 2013)]
202             dataframe_test = dataframe.loc[(dataframe["Cohorte"] == 2014)
                | (dataframe["Cohorte"] == 2013)]
203         else:
204             print("Bad_Trget")
205             return None
206
207     #Escalamiento de factores
208     if("Escalamiento" in args.keys() and (not(args["clasificador"] == "
        SVL") or args["clasificador"] == "SVR")):
209         print("escalar")
210         lista=list(args["columnas_a_escalar"])
211         for i in lista:
212             if(not i in args["columnas_clasificacion"]):
213                 lista.remove(i)
214         escalador=StandardScaler()
215         print(dataframe_train[lista])
216         dataframe_train[lista]=escalador.fit_transform(dataframe_train[
            lista])
217         dataframe_test[lista] = escalador.transform(dataframe_test[lista
            ])

```

```

218         print(dataframe_train[ lista ])
219
220     #Codificación de variables de tipo string y codificación One-Hot
221     if("one_hot" in args.keys()):
222         dataframe_train , dataframe_test , columnas_one_hot= onehot_encoder
223             (dataframe_train ,dataframe_test ,args["columnas_one_hot"])
224         args["columnas_a_estudiar"]= args["columnas_a_estudiar"] +
225             columnas_one_hot
226     print(args["columnas_a_estudiar"])
227
228     #Selección de clasificadores
229     if(args["clasificador"] == "SVL"):
230         clf=SVC(kernel="linear",max_iter =500000)
231         params=param_grid["SVL"]
232
233     elif(args["clasificador"] == "SVR"):
234         clf = SVC(kernel="rbf")
235         params = param_grid["SVR"]
236     elif(args["clasificador"] == "NN"):
237         clf = KNeighborsClassifier(n_jobs=-1)
238         params = param_grid["NN"]
239
240     elif(args["clasificador"] == "GB"):
241         clf=GradientBoostingClassifier()
242         params = param_grid["GB"]
243     elif(args["clasificador"]== "RF"):
244         clf = RandomForestClassifier()
245         params = param_grid["RF"]
246
247     else:
248         print("bad_classifier")
249         return None
250
251     #Selección de factores
252     if("Filtrar" in args.keys()):
253         print("Start_Filtering")

```

```

251     args["columnas_clasificacion"], resultados["
        columnas_rechazadas_en_filtro"], resultados["columnas_filtro"]
        , resultados["selector.pvalues_"], resultados["best_k"]=
        filtering_features(dataframe_train , args , clf)

252
253 #Escalamiento para algoritmos support vector machine
254 if ("Escalamiento" in args.keys() and ((args["clasificador"] == "SVL"
    ) or args["clasificador"] == "SVR")):
255     print("escalar")
256     lista = list(args["columnas_a_escalr"])
257     for i in lista:
258         if (not i in args["columnas_clasificacion"]):
259             lista.remove(i)
260     escalador = StandardScaler()
261     print(dataframe_train[lista])
262     dataframe_train[lista] = escalador.fit_transform(dataframe_train[
        lista])
263     dataframe_test[lista] = escalador.transform(dataframe_test[lista
        ])
264     print(dataframe_train[lista])
265
266 #Selección de parámetros
267 if("Tunning" in args.keys()):
268     print("Start_Tunning")
269     parameters = tuning_of_params(dataframe_train , args , clf , params)
270     resultados["parametros"]=parameters
271     clf.set_params(**parameters)
272
273 #Evaluación y generación de resultados
274 training_size=len(dataframe_train.index)
275 test_size=len(dataframe_test)
276 clf.fit(dataframe_train[args["columnas_clasificacion"]],
        dataframe_train["target"])
277 prediction=clf.predict(dataframe_test[args["columnas_clasificacion"
        ]])
278 tp, fp, tn, fn = confusion_matrix(list(dataframe_test["target"]),
        list(prediction))
279 accuracy = accuracy_score(dataframe_test["target"], prediction)

```

```

280     presicion = precision_score(dataframe_test["target"], prediction)
281     f1_score_1 = f1_score(dataframe_test["target"], prediction)
282     recall = recall_score(dataframe_test["target"], prediction)
283     resultados["columnas_clasificacion"]=args["columnas_clasificacion"]
284     print([args["clasificador"],tp,fp,tn,fn,presicion,f1_score_1,accuracy
        ,recall,training_size,test_size,json.dumps(resultados)])
285     resultados_dataframe.loc[index]=[args["clasificador"],tp,fp,tn,fn,
        presicion,f1_score_1,accuracy,recall,training_size,test_size,json.
        dumps(resultados),args["campus"],args["target"],args["
        f1_score_pond"],args["presicion_score_pond"]]
286     return resultados_dataframe
287
288
289 #Parámetros a probar en los algoritmos.
290 param_grid=dict()
291 param_grid["GB"] = {"learning_rate": [0.1,0.02,0.05], "min_samples_leaf"
    : [1,3,5,10,15,20,30,40,50,75,100], "min_samples_split"
    : [2,5,10,15,20,25,30,40,50,60,70,100], "max_depth"
    : [1,2,3,4,5,6,7,8,9,10], "n_estimators"
    : [10,20,30,40,50,60,70,90,100,200], 'random_state': list(range(1, 30,
    3))}
292 param_grid["SVL"] = {"C": list(range(1, 10000000, 50000)) + [0.01, 0.001,
    0.1, 1, 10, 100, 1000], 'random_state': list(range(1, 30, 3)) , '
    class_weight': ['balanced'], 'max_iter': [500000]}
293 param_grid["SVR"] = {"C": list(range(1, 10000000,50000))
    + [0.01,0.001,0.1,1,10,100,1000], "gamma": list(np.logspace(-15, 3, 15))
    + [0.002,0.1,100,1000,10000], 'random_state': list(range(1, 30, 3)) , '
    class_weight': ['balanced'], 'max_iter': [1000000]}
294 param_grid["NN"] = {"n_neighbors": list(range(1, 100)), "weights": ["
    uniform", "distance"],
295     "algorithm": ["ball_tree", "kd_tree"], "p": [1, 2], "n_jobs
        ": [-1]}
296
297 param_grid["RF"] = {
298     "min_samples_leaf": [1, 3, 5, 10, 15, 20,25],
299     "min_samples_split": [2, 5, 10, 15, 20, 25],
300     "max_depth": [1, 2, 3, 4],

```

```

301         "n_estimators": [ 20, 30, 40, 50, 60, 70, 90,
302                             100,200],
303         "n_jobs":[-1],
304         'random_state': list(range(1, 30, 3))}
305
306 resultados=pd.DataFrame(index=np.arange(0, 20), columns=('Modelo_
307     Clasificador', 'True_Positives', 'False_Positives', 'True_Negatives', '
308     False_Negatives', 'Precision', 'F1_SCORE', 'Accuracy', 'Recall', 'Registros
309     _utilizados_para_entrenamiento', 'Registros_utilizados_para_testing', '
310     columnas', 'campus', 'target', "f1_score", "precion_pond"))
311
312 diccionario=dict()
313 index=0
314
315 #Columnas a ser escaladas
316 diccionario["columnas_a_escalas"]=["Npsu_mat", "Npsu_nem", "Npsu_len", "
317     Reprob", "edad", "beca", 'MIIR', 'FIIR', 'psu_nem', 'psu_leng', 'beca', '
318     ingreso_fam']
319
320 #Dirección de la base de datos.
321 diccionario["direccion_BD"]='BD'
322
323 #Número de combinaciones de parámetros a utilizar en la fase de selección
324     de parámetros
325 diccionario["random_sample"] = 1000000
326
327 #Columnas a codificar con una codificación One-Hot
328 diccionario["columnas_one_hot"]=['salud', 'rama_col', 'tipo_col', 'Carrera',
329     'reg_colegio', 'Gender']
330
331 #Columnas utilizadas para clasificar
332 diccionario["columnas_clasificacion"] = diccionario["columnas_a_estudiar"
333     ] = [ 'FIIR', 'MIIR', 'edad', 'ingreso_fam', 'psu_nem', 'psu_leng', 'psu_mat'
334     , 'beca', 'cambiodecomuna', 'AP', 'AF', 'AM', 'AQ', 'Reprob', "Npsu_mat", "
335     Npsu_nem", "Npsu_len"]
336
337
338 #Opción de realizar tuning de parámetros (1) o no (0)
339 diccionario["Tunning"]=1
340
341 #Ponderaciones de las métricas para la selección de parámetros
342 diccionario["presicion_score_pond"]= 0
343
344 diccionario["f1_score_pond"]= 1
345
346 #Opción de realizar filtro de factores (1) o no (0)

```

```

327  diccionario["Filtrar"] = 1
328  #Opción de realizar codificación One-Hot de factores (1) o no (0)
329  diccionario["one_hot"] = 1
330  #Lista de percentiles a utilizar en el filtro de factores
331  diccionario["percentile_list"]=[5 ,10 , 15, 20,25, 30, 35,40, 45, 50,
    55,60, 65, 70, 75 , 80, 85 , 90, 95, 100]
332
333  #Modelos para predecir la deserción en el primer año académico
334  diccionario_1=dict(diccionario)
335  diccionario_1["campus"]="All"
336  diccionario_1["target"]="S3"
337
338
339  #Support vector machine
340  diccionario_1_1_2=dict(diccionario_1)
341  diccionario_1_1_2["clasificador"]="SVL"
342  diccionario_1_1_2["Escalamiento"]=1
343  resultados=model_experimenter(dict(diccionario_1_1_2),param_grid ,
    resultados ,index)
344  index+=1
345
346  #K nearest neighbor
347  diccionario_1_4=dict(diccionario_1)
348  diccionario_1_4["clasificador"]="NN"
349  resultados=model_experimenter(dict(diccionario_1_4),param_grid ,resultados
    ,index)
350  index+=1
351
352  #Random forest
353  diccionario_1_3=dict(diccionario_1)
354  diccionario_1_3["clasificador"]="RF"
355  resultados=model_experimenter(dict(diccionario_1_3),param_grid ,resultados
    ,index)
356  index+=1
357
358  #Gradient boosting
359  diccionario_1_2=dict(diccionario_1)
360  diccionario_1_2["clasificador"]="GB"

```

```

361 resultados=model_experimenter(dict(diccionario_1_2),param_grid,resultados
    ,index)
362 index+=1
363
364
365 #Modelos para predecir la deserción en el segundo año académico
366 diccionario_1=dict(diccionario)
367 diccionario_1["campus"]="ALL"
368 diccionario_1["target"]="S5"
369
370
371 #Support vector machine
372 diccionario_1_1_2=dict(diccionario_1)
373 diccionario_1_1_2["clasificador"]="SVL"
374 diccionario_1_1_2["Escalamiento"]=1
375 resultados=model_experimenter(dict(diccionario_1_1_2),param_grid,
    resultados,index)
376 index+=1
377
378 #Random Forest
379 diccionario_1_3=dict(diccionario_1)
380 diccionario_1_3["clasificador"]="RF"
381 resultados=model_experimenter(dict(diccionario_1_3),param_grid,resultados
    ,index)
382 index+=1
383
384 #K nearest neighbor
385 diccionario_1_4=dict(diccionario_1)
386 diccionario_1_4["clasificador"]="NN"
387 resultados=model_experimenter(dict(diccionario_1_4),param_grid,resultados
    ,index)
388 index+=1
389
390 #Gradient boosting
391 diccionario_1_2=dict(diccionario_1)
392 diccionario_1_2["clasificador"]="GB"
393 resultados=model_experimenter(dict(diccionario_1_2),param_grid,resultados
    ,index)

```

```
394 index+=1
395
396 #Dirección en donde guardar los resultados
397 writer = pd.ExcelWriter('Dirección_de_resultados')
398 resultados.to_excel(writer, 'Hoja_1', index=False)
399 writer.save()
```
