GUIA CRISP-DM

En el presente informe de Machine Learning se expone el proceso para la extracción de modelos computacionales que representan el conocimiento que existe en los datos académicos de la Carrera de Ingeniería en Sistemas y la Carrera de Computación; datos proporcionados por la Unidad de Telecomunicaciones e Información (UTI) de la Universidad Nacional de Loja (UNL). Este trabajo se pretende realizar ordenadamente, por ello, se utiliza la metodología CRISP-DM. A continuación, se desarrolla cada una de las fases con sus tareas correspondientes.

1. Comprensión del negocio

La UNL es una institución pública de educación superior que tiene como objetivo la formación de profesionales, con sólidas bases científicas y técnicas, pertinencia social y valores; la generación y aplicación de conocimientos científicos, tecnológicos y técnicos, que aporten al desarrollo integral del entorno y al avance de la ciencia[1].

La Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables (FEIRNNR) perteneciente a la UNL, forma profesionales con bases sólidas y enfoques científico-humanista en los niveles tecnológico-técnico-artesanal y de pregrado, acorde a la vanguardia tecnológica en los avances científicos. La Carrera Ingeniería en Sistemas prepara profesionales que contribuyan al desarrollo científico-tecnológico, con talento humano capaz de brindar soluciones informáticas y computacionales, eficientes y eficaces a las necesidades de la sociedad, aplicando programas de investigación, desarrollo e innovación. La Carrera de Computación forma profesionales con capacidades científico, técnicas y humanistas en los campos de estudio de las Matemáticas, Ciencias Agrarias, Ciencias de la Vida, Química, Física, Ciencias de la Tierra y del Espacio, y Ciencias Económicas [1].

1.1. Determinar los objetivos del negocio

El objetivo empresarial de las Carreras de Computación e Ingeniería en Sistemas es

preparar profesionales altamente capacitados que contribuyan al desarrollo, brindando soluciones eficaces a la sociedad. De esta manera, se busca mejorar la calidad en la educación, aumentando el número de graduados y reduciendo la deserción académica.

1.2. Evaluar la situación

Actualmente, el principal objetivo de la empresa en cuestión se ha visto comprometido, ya que el talento humano generado es poco, debido a que: los estudiantes tardan muchos años en concluir la Carrera profesionalizante, pierden considerablemente asignaturas y ciclos académicos; llegando a provocar en ocasiones el abandono de los estudios. Al inicio del proyecto, la UNL dispone de información de estudiantes que se encuentran cursando o que ya han terminado una titulación. Sin embargo, no existen soluciones o estudios a profundidad sobre el análisis del rendimiento académico mediante herramientas de machine learning que permitan predecir el éxito o fracaso estudiantil, y peor aún, que la institución se encuentre preparada para que en el futuro se implementen soluciones basadas en tecnologías emergentes.

1.3. Determinar los objetivos de la minería de datos

El proyecto "Machine Learning para el análisis del rendimiento académico de estudiantes de Ingeniería", tiene la finalidad de extraer conocimiento en forma de modelos que representen los patrones que cumplen ciertos grupos de estudiantes mientras cursan la Carrera de Ingeniería en Sistemas y la Carrera de Computación. Los modelos que se buscan deben ser fiables para poder predecir los resultados de los futuros periodos académicos. Concretamente, a través de la guía metodológica de CRISP-DM se busca lo siguiente:

- Encontrar modelos computacionales que representen el conocimiento cuando un estudiante aprueba o reprueba un periodo académico en base a las asignaturas dictadas en los primeros ciclos de la Carrera de Ingeniería de Sistemas y la Carrera de Computación de la universidad Nacional de Loja.

Los resultados permitirían identificar cuando un individuo o conjunto de individuos podrían tener fracasos académicos, ya sea por falta de preparación de los alumnos, de los profesores o cualquier otro factor. Y así también, que el personal administrativo (gestores, decanos, directores, etc.) de la Universidad puedan mejorar la planificación o tomar algunas medidas que reduzcan los problemas que no permiten mejorar la calidad de la educación.

1.4. Producir el plan de proyecto

Para cumplir con las posteriores fases de la metodología CRISP-DM, se utiliza el lenguaje de programación Python, ya que, como se concluye en [2], es el lenguaje más apropiado para desarrollar el presente trabajo.

Según KDnuggets [3], un conocido sitio web de discusión y aprendizaje para Business Analytics, Data Mining y Data Science, Python lidera las principales plataformas de la ciencia de datos y el aprendizaje automático.

La **Figura 1** presenta los resultados de las encuestas realizadas en el año 2019 y las relaciona con los resultados de las mismas encuestas, pero realizadas durante los años 2017 y 2018; encuestas acerca de Software de análisis, ciencia de datos y aprendizaje automático. Asimismo, en los trabajos [4] y [5] se menciona que Python es la herramienta más popular para aplicaciones de aprendizaje automático.

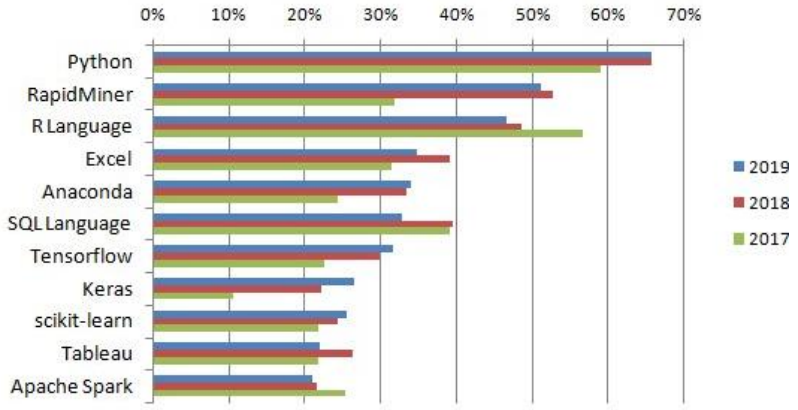


Figura 1 Comparativa del uso de las once principales plataformas para la machine learning.

En la ciencia de datos se requiere el uso de herramientas analíticas, tecnologías y lenguajes de programación que ayuden a extraer conocimiento valioso de los datos. En [6] y [7] se utiliza Jupyter Notebook como la herramienta principal para codificar en Python, además de ello, [8] trabaja con archivos de Python almacenados en Google Drive.

Una encuesta reciente del año 2019 realizada por Kaggle [9], revela que Python conjuntamente con el IDE Júpiter son los más populares, con un porcentaje mayor al 83% de uso regular. La **Figura 2** presenta los resultados de dichas encuestas.

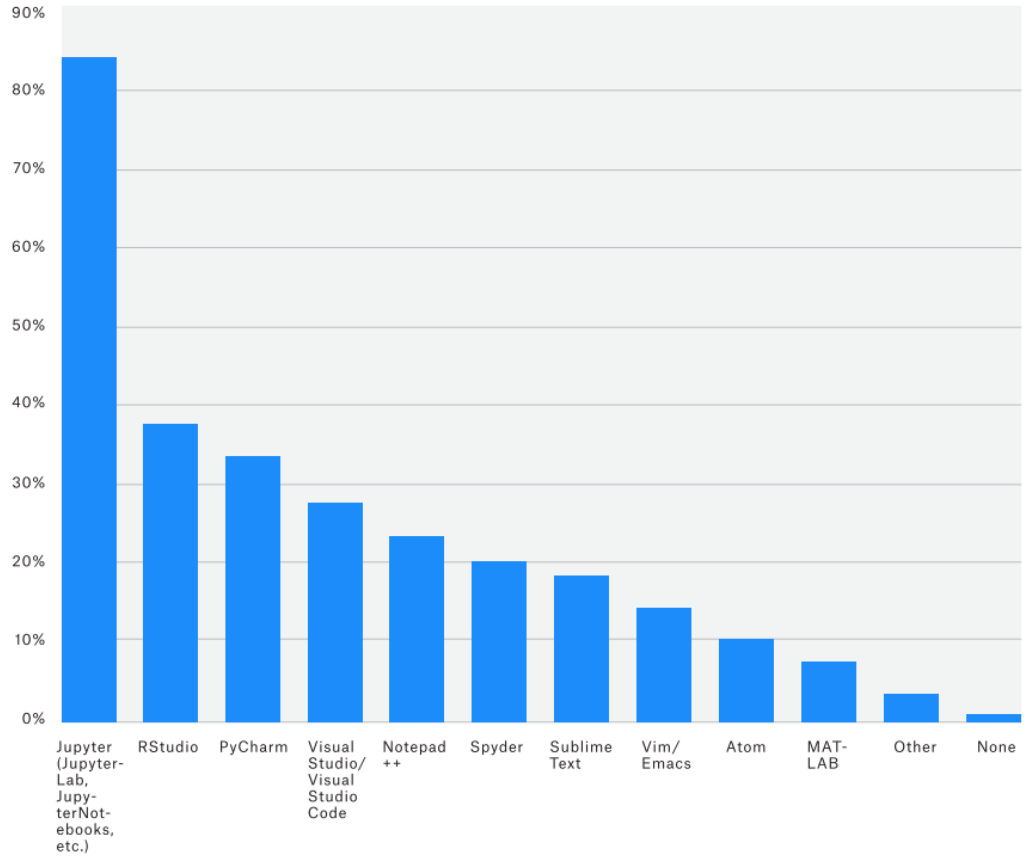


Figura 2 Entornos de desarrollo más populares en el machine learning

HemaMalini [5] en su investigación utiliza Scikit-learn debido a que es la biblioteca de aprendizaje automático más utilizada actualmente para Python. De la misma manera, [5] menciona que casi todas las demás bibliotecas y marcos desarrollados en los últimos años para el aprendizaje automático, son compatibles con Scikit-learn y muy fáciles de implementar.

En base al resumen ejecutivo de [9] obtenemos que el framework más utilizado es Scikit-learn, con más del 80%. Scikit-learn es un paquete de Python que contiene algoritmos populares de machine learning. La **Figura 3** resume lo mencionado.

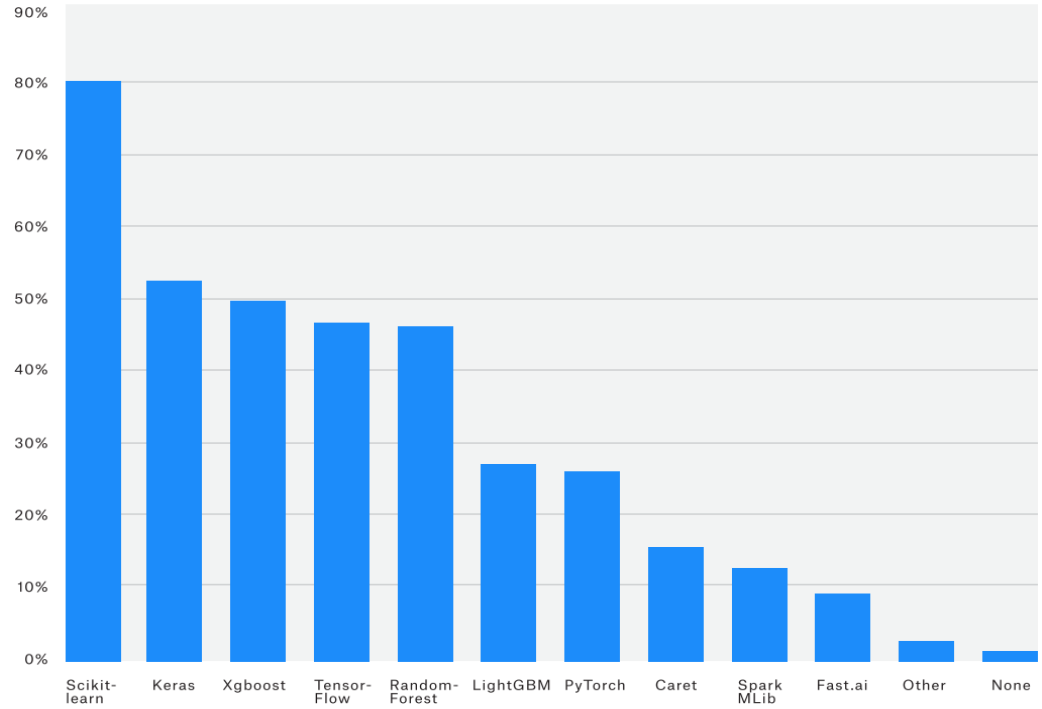


Figura 3 Framewoks más utilizados para el machine learning

Además, [9] también presenta información sobre los algoritmos más utilizados en machine learning, en donde se dice que: los métodos más comunes son la regresión lineal o logística, seguida de los árboles de decisión. En la **Figura 4** se presenta la gráfica correspondiente a dicho análisis.

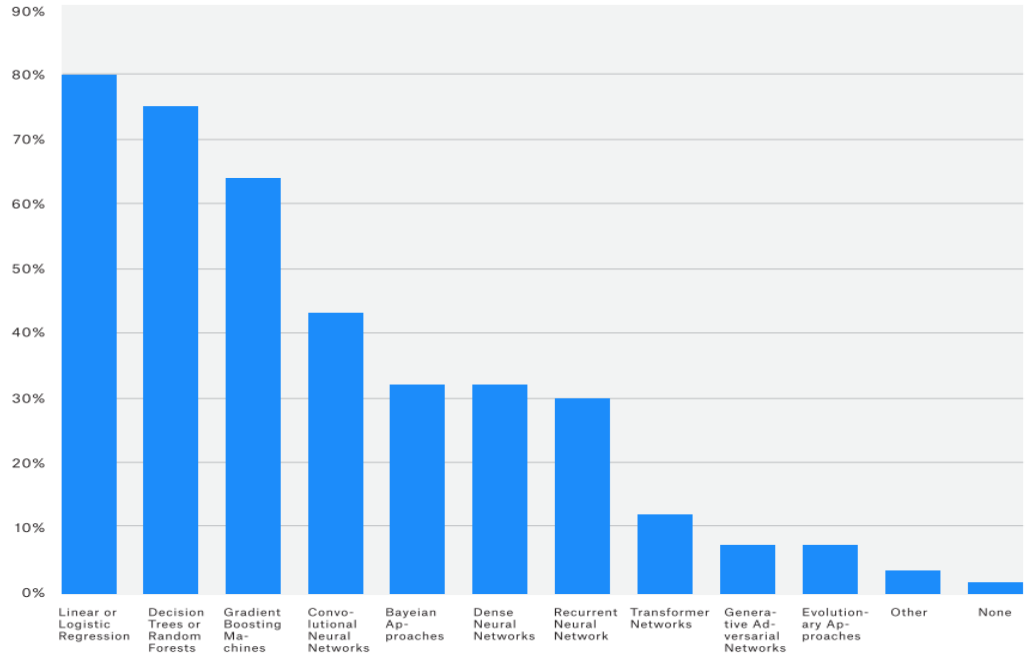


Figura 4 Algoritmos más utilizados en machine learning

Para dar cumplimiento a los objetivos de la presente actividad, se procede a detallar las etapas a desarrollarse con el fin de mantener el orden y una mejor organización del trabajo.

1. •Obtener los datos y seleccionar las herramientas y técnicas a utilizar.
2. •Vizualización y comprensión de los datos.
3. •Realizar el preprocesamiento del conjunto de datos.
4. •Seleccionar el algoritmo de machine learning que mejor de adapte para el proceso de extracción de conocimiento.
5. •Evaluar los modelos obtenidos a travez de las diversas técnicas
6. •Documentar los resultados finales.

Para tener un mayor éxito en los proyectos de machine learning se necesita una elevada potencia de cálculo y por ende, gran uso de CPU o GPU; por lo tanto, si no se dispone del hardware necesario, [8] recomienda utilizar Google Colab. La plataforma Google Colab es un entorno de Jupyter Notebook que no requiere ninguna configuración inicial y que facilita el acceso a GPU [7]. El servicio gratuito de Google Colab es un entorno de desarrollo interactivo que se ejecuta completamente en la nube, al que se puede acceder con un navegador web sin importar el sistema operativo. Las principales ventajas que brinda *Google Colab* son:

- No necesita una configuración previa.
- Permite el acceso gratuito a CPU, GPU y TPU.
- Facilita el trabajo colaborativo.

2. Comprensión de los datos

En esta fase se empieza a familiarizar con los datos y analizar su calidad; se presenta desde cómo se los recolectó, hasta describirlos gráficamente.

2.1. Recopilar los datos iniciales

Los datos que se utilizaran para cumplir con el siguiente trabajo se los obtuvo mediante una solicitud a la Unidad de Telecomunicaciones e Información (UTI) de la Universidad Nacional de Loja (UNL), la cual brindó el dataset correspondiente. Para más detalles acerca del proceso de recopilación de los datos, se sugiere revisar el ***Informe de obtención de datos***.

2.2. Describa los datos

El reporte acumulado de datos en formato ***csv***, consta de un total de 36362 registros (filas) con 51 atributos (columnas) que pertenecen a un total de 1814 estudiantes, entre los que han terminado, abandonado y los que se encuentran cursando las Carreras de Ingeniería en Sistemas y la Carrera de Computación en la UNL.

En la **Tabla 1** se detalla cada una de las columnas (features) del conjunto de datos:

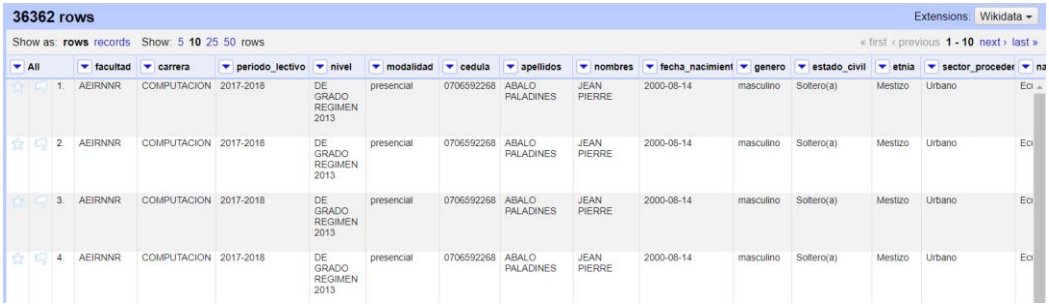
*Tabla 1 Descripción de los datos del dataset inicial***.**

CAMPO	TIPO DE DATO	DESCRIPCIÓN
facultad	TEXTO	Almacena las siglas de la facultad o área a la que pertenece la Carrera que el alumno estudió o se encuentra estudiando. En este caso, contiene únicamente el texto "AEIRNNR".
Carrera	TEXTO	Corresponde al nombre de la Carrera o especialidad que estudió o se encuentra estudiando el alumno.
periodo_lectivo	TEXTO	Representa el año inicial y el año final en la que el alumno llevó a cabo un ciclo académico.
nivel	TEXTO	Simboliza el régimen al que pertenece la Carrera que el alumno estudió o se encuentra estudiando.
modalidad	TEXTO	Contiene el nombre de la modalidad en que se desarrolla la Carrera profesional.
cedula	TEXTO	Almacena el numero de la cedula de ciudadanía perteneciente a un alumno estudió o se encuentra estudiando.
apellidos	TEXTO	Este campo contiene los apellidos del estudiante
nombres	TEXTO	Este campo contiene los nombres del estudiante
fecha_nacimiento	TEXTO	Corresponde a la fecha en que nació el estudiante
genero	TEXTO	Contiene el género masculino o femenino del alumno.
estado_civil	TEXTO	Almacena el estado civil que corresponde a cada estudiante.
etnia	TEXTO	Corresponde a la etnia con la que se identifica el estudiante.
sector_procedencia	TEXTO	Sector urbano o rural del que proviene el estudiante
nacionalidad	TEXTO	Nacionalidad a la que pertenece el estudiante
pais_nacimiento	TEXTO	País en donde nació el estudiante
provincia_nacimiento	TEXTO	Provincia o departamento en donde nació el alumno
canton_nacimiento	TEXTO	Cantón en donde nació el estudiante
ciudad_nacimiento	TEXTO	Ciudad en donde nació el estudiante
direccion_nacimiento	TEXTO	Dirección o ubicación en donde nació el estudiante
pais_actual	TEXTO	País en donde vive actualmente el estudiante
provincia_actual	TEXTO	Provincia en donde vive actualmente el estudiante
canton_actual	TEXTO	Cantón en donde vive actualmente el estudiante
parroquia_actual	TEXTO	Parroquia en donde vive actualmente el estudiante
ciudad_actual	TEXTO	Ciudad en donde vive actualmente el estudiante
direccion_actual	TEXTO	Dirección del domicilio en donde vive actualmente el estudiante
trabaja	TEXTO	Condición de empleo actual del alumno, es decir si trabaja o no trabaja.
ingreso_estudiante	NUMÉRICO	Corresponde a la cantidad de dinero en dólares que el alumno tiene como ingresos mensuales
numero_hijos	NUMÉRICO	Representa la cantidad de hijos o personas bajo la responsabilidad del estudiante
colegio	TEXTO	Es el nombre del colegio del que proviene el alumno

tipo_colegio	TEXTO	Corresponde al tipo (público, privado, etc.) de colegio del que proviene el estudiante.
pais_colegio	TEXTO	Nombre del país en que se encuentra el colegio del que proviene el estudiante.
provincia_colegio	TEXTO	Nombre de la provincia en que se encuentra el colegio del que proviene el estudiante.
canton_colegio	TEXTO	Nombre del cantón en que se encuentra el colegio del que proviene el estudiante.
oferta_academica	TEXTO	Corresponde al nombre la malla curricular que cumple el alumno
ciclo	NUMÉRICO	Es el número que representa al ciclo académico que el estudiante cursó.
numero_matricula	NUMÉRICO	Es el número que identifica la matrícula en cada ciclo académico
malla_curricular	TEXTO	Es el nombre del pensum académico al que pertenece el alumno
estado_matricula	TEXTO	Es el resultado (aprobó o reprobó) final que obtiene un estudiante en un ciclo académico.
paralelo	TEXTO	Es el nombre del paralelo al que perteneció el estudiante en cada ciclo académico
jornada	TEXTO	Representa la jornada en que se desarrolla la Carrera profesional del estudiante
asignatura	TEXTO	Es el nombre de la asignatura que cursó el estudiante
estado_asignatura	TEXTO	Es el resultado (aprobó o reprobó) final que obtiene un estudiante en cada asignatura desarrollada durante cada periodo académico.
obligatoria	TEXTO	Cualidad de las asignaturas, es decir, representa si una asignatura es obligatoria o no.
arrastrable	TEXTO	Cualidad de las asignaturas, es decir, representa si una asignatura es arrastrable o no.
asistencia_obligatoria	TEXTO	Cualidad de las asignaturas, es decir, representa si una asignatura es de asistencia obligatoria o no.
nota_ingresada	NUMÉRICO	Es el valor numérico de la calificación final que obtuvo el estudiante en cada asignatura.
nota_ponderada	NUMÉRICO	Corresponde al valor ponderado de las calificaciones ingresadas
porcentaje_asistencias	NUMÉRICO	Corresponde al porcentaje de horas asistidas, en relación al total de horas planificadas
promedio_matricula	NUMÉRICO	Corresponde al promedio final obtenido por el estudiante sobre las asignaturas desarrolladas en un ciclo específico
homologada	TEXTO	Representa si un estudiante homologó alguna(s) materia(s)
observacion_homologacion	TEXTO	En este campo se encuentran las observaciones registradas cuando existe una homologación

2.3. *Explorar los datos*

Para cumplir con esta tarea se procedió a visualizar el contenido del documento (csv) que contiene la información de los estudiantes. Este primer acercamiento se la realizó mediante la herramienta OpenRefine, como se muestra en la **Figura 5**, en la cual también se puede observar que los registros de cada estudiante se encuentran organizados en forma vertical; es decir, existe una fila por cada asignatura a la que ha pertenecido el estudiante.



36362 rows

Extensions: Wikidata

Show as: rows records Show: 5 10 25 50 rows « first ‹ previous 1 - 10 next › last »

	All	facultad	carrera	periodo_lectivo	nivel	modalidad	cedula	apellidos	nombres	fecha_nacimient	genero	estado_civil	etnia	sector_proceder	na
1.		AEIRNNR	COMPUTACION	2017-2018	DE GRADO REGIMEN 2013	presencial	0706592268	ABALO PALADINES	JEAN PIERRE	2000-08-14	masculino	Soltero(a)	Mestizo	Urbano	Ec
2.		AEIRNNR	COMPUTACION	2017-2018	DE GRADO REGIMEN 2013	presencial	0706592268	ABALO PALADINES	JEAN PIERRE	2000-08-14	masculino	Soltero(a)	Mestizo	Urbano	Ec
3.		AEIRNNR	COMPUTACION	2017-2018	DE GRADO REGIMEN 2013	presencial	0706592268	ABALO PALADINES	JEAN PIERRE	2000-08-14	masculino	Soltero(a)	Mestizo	Urbano	Ec
4.		AEIRNNR	COMPUTACION	2017-2018	DE GRADO REGIMEN 2013	presencial	0706592268	ABALO PALADINES	JEAN PIERRE	2000-08-14	masculino	Soltero(a)	Mestizo	Urbano	Ec

Figura 5 Visualización de los datos a través de OpenRefine

Para completar las tareas de la presente fase, se trabajó desde Google Colab y Google Drive. La **Figura 6** enseña la interfaz principal de la herramienta mencionada. Google Drive permite trabajar con archivos almacenados en la nube, tal como el cuaderno de Jupyter en formato ***ipynb*** y el dataset del histórico de matrículas en formato ***csv.*** Se importó las librerías necesarias para realizar la conexión con Google Drive y las librerías para manipular el conjunto de datos. Para cargar el dataset y mantenerlo en memoria, se almacenó el contenido en una variable de tipo *dataframe*.

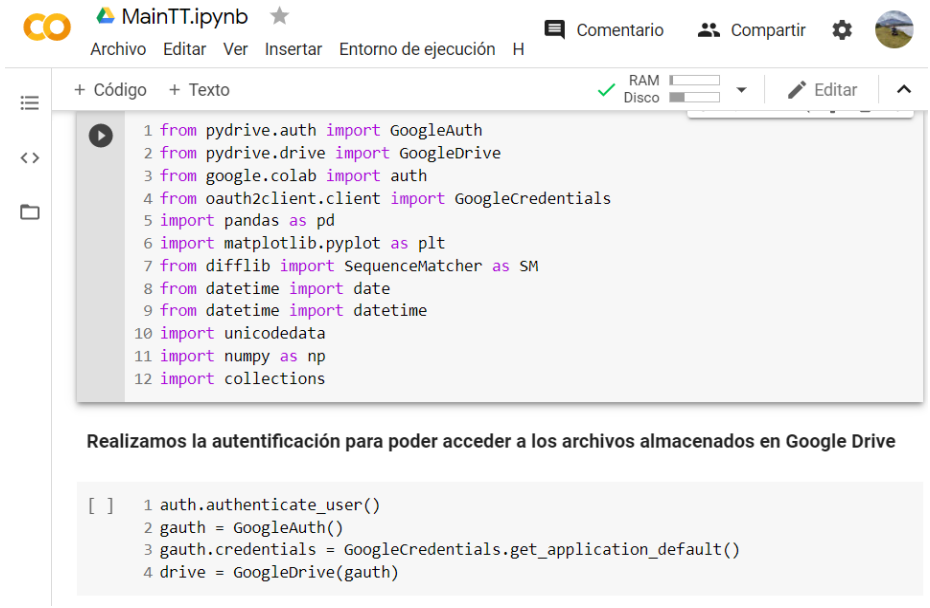
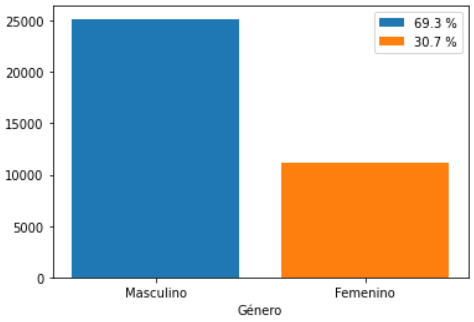
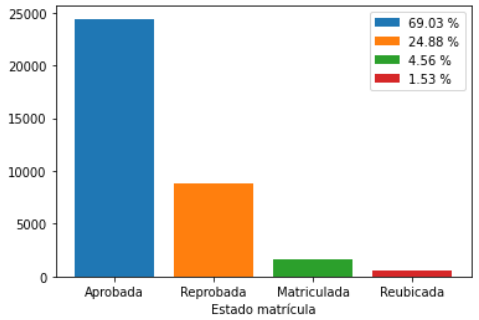
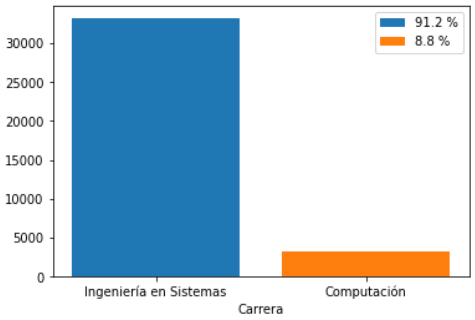


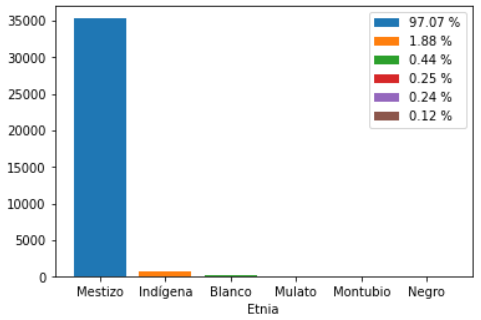
Figura 6 Interfaz principal de Google Colab

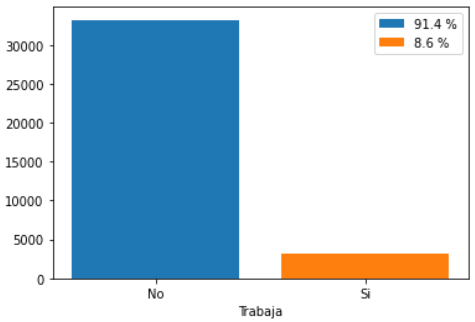
A partir de los datos originales y mediante diagramas de barras, se procede a mostrar la frecuencia de los siguientes atributos: género, estado de la matrícula, Carrera, etnia, trabaja, sector de procedencia, estado civil y malla curricular, y se obtuvo los siguientes resultados.

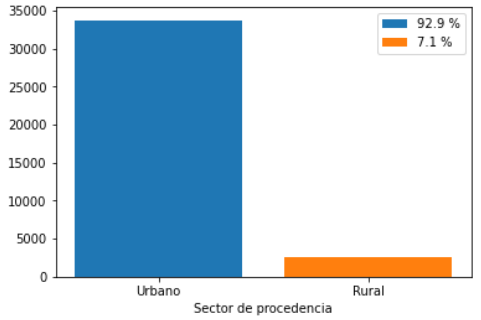


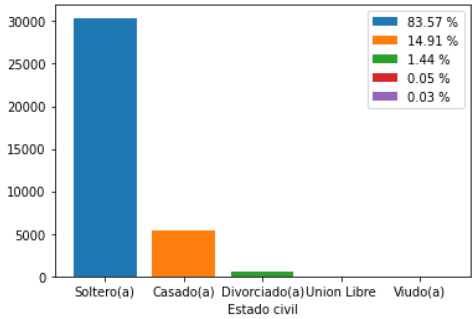


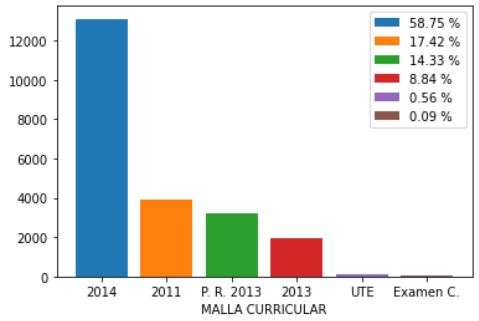












Para más detalles, se recomienda revisar el **Informe del análisis exploratorio de datos**.

2.4. Verificar la calidad de los datos

De acuerdo con el análisis exploratorio y mediante el comando *df.isnull().any().any()* se conoció de que sí existen datos faltantes en el dataset. La ***Tabla 2*** presenta cada una de las columnas con su respectivo porcentaje que refleja la completitud de los datos. La correción (agregar, eliminar, completar) de los datos se aplicarán en la siguiente fase.

Tabla 2 Porcentaje de datos completos

CAMPO	PORCENTAJE DE DATOS COMPLETOS
facultad	100.00 %
Carrera	100.00 %
periodo_lectivo	97.30 %
nivel	100.00 %
modalidad	100.00 %
cedula	100.00 %
apellidos	100.00 %
nombres	100.00 %
fecha_nacimiento	100.00 %
genero	100.00 %
estado_civil	100.00 %
etnia	100.00 %
sector_procedencia	100.00 %
nacionalidad	100.00 %
pais_nacimiento	100.00 %

provincia_nacimiento	99.98 %
canton_nacimiento	100.00 %
ciudad_nacimiento	98.44 %
direccion_nacimiento	97.57 %
pais_actual	99.93 %
provincia_actual	99.99 %
canton_actual	99.99 %
parroquia_actual	89.95 %
ciudad_actual	99.18 %
direccion_actual	95.89 %
trabaja	100.00 %
ingreso_estudiante	100.00 %
numero_hijos	100.00 %
colegio	85.83 %
tipo_colegio	81.79 %
pais_colegio	81.79 %
provincia_colegio	81.79 %
canton_colegio	81.79 %
oferta_academica	97.30 %
ciclo	100.00 %
numero_matricula	100.00 %
malla_curricular	61.36 %
estado_matricula	97.30 %
paralelo	96.99 %
jornada	96.94 %
asignatura	92.05 %
estado_asignatura	90.80 %
obligatoria	100.00 %
arrastrable	97.30 %
asistencia_obligatoria	100.00 %
nota_ingresada	84.77 %
nota_ponderada	84.77 %
porcentaje_asistencias	91.15 %
promedio_matricula	96.11 %
homologada	100.00 %
observacion_homologacion	2.70 %

El atributo ***cedula*** se utilizará para identificar los registros que se encuentran relacionados o que pertenecen a un solo estudiante. Para lo cual, en esta fase también se procedió a validar los números de cédula, considerando la cantidad de caracteres y aplicando el algoritmo de validación de cédulas ecuatorianas.

```
allCedulas = df['cedula']
cedulas = allCedulas.drop_duplicates().to_list()
print("Total de cédulas únicas: ", len(cedulas))
lista = list()
for cedula in cedulas:
  cedula = str(cedula)
  if(len(cedula) == 10):
    if(validarCedulaEcuador(cedula)==False):
      lista.append(cedula)
  else:
    lista.append(cedula)

print("Las cédulas no validadas son:", len(lista))
print(lista)
```

```
Total de cédulas únicas:  1814
Las cédulas no validadas son: 11
```

Figura 7 Script para identificar cedulas no validadas

Los hallazgos obtenidos mediante la ejecución del script que se muestra en la **Figura 7**, son: 1814 números de cédula únicos, de los cuales, 11 no cumplen con el algoritmo de validación de cédulas ecuatorianas. Mediante un resumen de valores nulos aplicado al dataset con el comando ***df.isna().sum().sum()***, se obtuvo que existen 116096 valores desconocidos, y para contar los registros únicamente de la columna ***malla_curricular,*** con el comando ***values_counts(dropna=false)***, se conoció que, de 36362, 14052 son nulos (NAN), la **Figura 8** exterioriza lo mencionado.

```
df['malla_curricular'].value_counts(dropna=False)
```

```
NaN                   14052
2014                  13108
2011                   3886
PENSUM_REGIMEN_2013    3198
2013                   1973
UTE                     125
Exámen complexivo        20
Name: malla_curricular, dtype: int64
```

Figura 8 Valores nulos de la columna malla curricular

Se agrupó la información, considerando las diversas mallas curriculares y se obtuvo que los datos pertenecientes a periodos anteriores al año 2013 presentan inconsistencias y un elevado número de datos nulos (desconocidos). Por tal motivo, se puede concluir que las mallas diferentes a las que se encuentran vigentes actualmente en la Carrera de Ingeniería en Sistemas y la Carrera de Computación, no brindan información completa y precisa. También es necesario mencionar que el sistema de Gestión Académica se ha venido actualizando frecuentemente, lo que ha permitido automatizar más los procesos, dando así una mejor calidad de la información que se guarda en el sistema; además de lo mencionado anteriormente, es importante conocer que la Carrera de Ingeniería en Sistemas inició con el Sistema Académico Modular por Objetos de Transformación (SAMOT) y que, con la aparición de la Ley Orgánica de Educación Superior, en 2012 paso a la aprobación por créditos.

3. Preparación de datos

Para cumplir con esta fase se utilizó la información de la ***Tabla 2*** como base para identificar los atributos más relevantes, y se relacionó con los objetivos planteados inicialmente.

3.1. Seleccione los datos.

En base a los objetivos planteados para el desarrollo de la presenta guía, se procedió a eliminar las siguientes columnas (features) del dataset: facultad, modalidad, pais_nacimiento, provincia_nacimiento, tipo_colegio, ciudad_nacimiento, direccion_nacimiento, pais_actual, provincia_actual, canton_actual, parroquia_actual, ciudad_actual, direccion_actual, colegio, pais_colegio, provincia_colegio, canton_colegio, numero_matricula, jornada, obligatoria, arrastrable, asistencia_obligatoria, nota_ingresada, nota_ponderada, nombres, apellidos, promedio_matricula, porcentaje_asistencias, homologada y observacion_homologacion. Este primer filtro de información se realizó en base al contexto de la investigación,

luego del primer análisis exploratorio y la comprensión de los datos. Las columnas que se eliminaron presentan en su mayoría valores incompletos y poco significativos en el presente proyecto.

El resultado luego de eliminar las columnas anteriormente mencionadas, es un nuevo dataset que contiene las siguientes características: Carrera, periodo_lectivo, nivel, cedula, genero, estado_civil, etnia, sector_procedencia, nacionalidad, canton_nacimiento, trabaja, ingreso_estudiante, numero_hijos, oferta_academica, ciclo, malla_curricular, estado_matricula, paralelo, asignatura, estado_asignatura y edad.

La columna ***edad*** es un nuevo atributo que se originó utilizando la variable ***fecha_nacimiento***, y se calculó la edad del estudiante utilizando la fecha actual. Posiblemente, más adelante se elimine otras columnas, según el uso y discernimiento del autor.

Luego de conocer el conjunto de datos, se procedió a realizar la discretización de los datos para que los algoritmos de machine learning funcionen de una mejor manera. Las clases que se discretizarán tiene datos binarios, es decir existen dos clases en cada columna, por tal motivo se reemplazó directamente los valores conocidos como: ***estado_asignatura, estado_matricula, genero, estado_civil, etnia, sector_procedencia, cantón_nacimiento, trabaj, numero_hijos*** e ***ingreso_estudiante***. Se remplazó los datos de texto por valores binarios, 1 y 0 para las diferentes clases. Para comprender los nuevos valores, se recomienda revisar el diccionario de datos, en donde se describe su significado. El script correspondiente a esta acción se presenta en la **Figura 9**.

```
dataset.estado_asignatura[dataset.estado_asignatura=='APROBADA']='1'
dataset.estado_asignatura[dataset.estado_asignatura=='REPROBADA']='0'

dataset.estado_matricula[dataset.estado_matricula=='APROBADA']='1'
dataset.estado_matricula[dataset.estado_matricula=='REPROBADA']='0'

dataset.genero[dataset.genero=='MASCULINO']='1'
dataset.genero[dataset.genero=='FEMENINO']='0'

dataset.estado_civil[dataset.estado_civil=='SOLTERO(A)']='1'
dataset.estado_civil[dataset.estado_civil!='SOLTERO(A)']='0'

dataset.etnia[dataset.etnia=='MESTIZO'] = '1'
dataset.etnia[dataset.etnia!='MESTIZO'] = '0'

dataset.sector_procedencia[dataset.sector_procedencia=='URBANO']='1'
dataset.sector_procedencia[dataset.sector_procedencia=='RURAL']='0'

dataset.canton_nacimiento[dataset.canton_nacimiento=='LOJA']='1'
dataset.canton_nacimiento[dataset.canton_nacimiento!='LOJA']='0'

dataset.trabaja[dataset.trabaja=='SI'] = '1'
dataset.trabaja[dataset.trabaja=='NO'] = '0'

dataset.numero_hijos[dataset.numero_hijos != 0] = '1'
dataset.numero_hijos[dataset.numero_hijos == 0] = '0'

dataset.ingreso_estudiante[dataset.ingreso_estudiante!= 0] = '1'
dataset.ingreso_estudiante[dataset.ingreso_estudiante== 0] = '0'
```

Figura 9 Discretización de los datos

Luego de discretizar lo datos anteriormente mencionados, se continuó con la generación de datasets a partir del ya preprocesado, lo que se realizó fue lo siguiente:

a. Se generó dos nuevos datasets, uno correspondiente a los estudiantes de la Carrera de Ingeniería en Sistemas y otro de la Carrera de Computación. La **Figura 10** muestra el código utilizado.

```
dataset_sistemas = dataset[dataset['carrera'] == "INGENIERIA EN SISTEMAS"]
dataset_computacion = dataset[dataset['carrera']  == "COMPUTACION"]
```

Figura 10 División del dataset por Carreras.

b. Luego de haber generado los datasets ***dataset_sistemas*** y ***dataset_computacion***, se procedió a eliminar de cada conjunto de datos las siguientes columnas: Carrera, nivel, malla_curricular, oferta_academica, nacionalidad, periodo_lectivo y fecha_nacimiento; ya que, en estas alturas carecen de importancia. La **Figura 11** muestra la codificación aplicada.

```
columns = ['carrera', 'nivel', 'malla_curricular', 'oferta_academica', 'nacionalidad', 'periodo_lectivo', 'fecha_nacimiento']
dataset_sistemas = dataset_sistemas.drop(columns, axis=1)
dataset_computacion = dataset_computacion.drop(columns, axis=1)
```

Figura 11 Eliminar columnas

c. Con los nuevos datasets ya un poco depurados, de continuó generando nuevos conjuntos de datos, pero en este paso se lo realizó por ciclos académicos.

En la Carrera de Computación hasta el momento de la extracción de datos había 4 ciclos, por tal motivo se dividió el dataset de sistemas en 4 nuevos conjuntos, uno por cada ciclo; según se lo puede visualizar en la **Figura 12**.

```
dc4 = dataset_computacion[dataset_computacion['ciclo'] == 4]
dc3 = dataset_computacion[dataset_computacion['ciclo'] == 3]
dc2 = dataset_computacion[dataset_computacion['ciclo'] == 2]
dc1 = dataset_computacion[dataset_computacion['ciclo'] == 1]
```

Figura 12 Conjuntos de datos de Computación

Del dataset de la Carrera de Ingeniería en Sistemas, se obtuvo 10 nuevos conjuntos de datos, conforme se lo puede visualizar en la **Figura 13**.

```
ds10 = dataset_sistemas[dataset_sistemas['ciclo'] == 10]
ds9 = dataset_sistemas[dataset_sistemas['ciclo'] == 9]
ds8 = dataset_sistemas[dataset_sistemas['ciclo'] == 8]
ds7 = dataset_sistemas[dataset_sistemas['ciclo'] == 7]
ds6 = dataset_sistemas[dataset_sistemas['ciclo'] == 6]
ds5 = dataset_sistemas[dataset_sistemas['ciclo'] == 5]
ds4 = dataset_sistemas[dataset_sistemas['ciclo'] == 4]
ds3 = dataset_sistemas[dataset_sistemas['ciclo'] == 3]
ds2 = dataset_sistemas[dataset_sistemas['ciclo'] == 2]
ds1 = dataset_sistemas[dataset_sistemas['ciclo'] == 1]
```

Figura 13 Conjuntos de datos de Sistemas

d. Luego de obtener los conjuntos de datos separados por Carrera y por cada uno de los ciclos, se procedió a transformar los datasets. Estos nuevos datasets tienen las asignaturas de primer y segundo ciclo en forma de columnas del dataset, ya que, la predicción se espera hacer en base a estas asignaturas aprobadas o reprobadas. La finalidad de estos últimos datasets es que se los utilice para generar los modelos predictivos.

Las nuevas columnas para los datasets de la Carrera de Ingeniería en Sistemas son: ***PROGRAMACION_I, CALCULO_INTEGRAL, PROBABILIDAD_E_INFERENCIA_ESTADISTICA, ECOLOGIA_Y_MEDIO_AMBIENTE_TECNOLOGICO, ALGEBRA_LINEAL, ESTRUCTURA_DE_DATOS, FISICA_II, CALCULO_DIFERENCIAL, FISICA, QUIMICA, EXPRESION_ORAL_Y_ESCRITA, FUNDAMENTOS_INFORMATICOS, cedula, genero, estado_civil, etnia, sector_procedencia, canton_nacimiento, trabaja, ingreso_estudiante, numero_hijos, ciclo, estado_matricula, paralelo*** y ***edad***. Para determinar los dataset que se utilizaran en las siguientes tareas de la presente fase, se procedió a eliminar las siguientes columnas: ciclo y cedula de los datasets. En esta parte se procedió a descartar los datos de la carrera de Ciencias de la Computación, debido a que la literatura dice que para crear los modelos deberíamos tener como mínimo 50 muestras, y como la carrera mencionada estaba empezando, al momento de realizar el presente trabajo, la cantidad de datos es insuficiente.

3.2. Limpieza de los datos

Hasta este momento, los conjuntos de datos han sido tratados, pero no depurados ni preprocesados completamente. Por lo tanto, en esta tarea se buscar completar los datos vacíos. Para el tratamiento de los datos faltantes se utilizó el método ***bfill***, de la librería pandas, este método consiste en completar los datos faltantes con los valores vecinos anteriores, es decir, es este caso, para completar los datos faltantes se utiliza los dos valores anteriores al dato faltante. El comando aplicado es el siguiente: ***dt_tmp=dt_tmp.fillna(method='bfill', limit=2)***.

3.3. Estructuración de los datos

Luego de haber finalizado la limpieza de los datos, se obtuvo presumiblemente unos datasets definitivos para aplicar las técnicas de clasificación previamente seleccionadas. En esta tarea también se ejecutó un proceso que permitió seleccionar las características. Luego de este proceso, se comprendió que se debía utilizar más data, y no solo los seleccionados correspondientes a las mallas vigentes.

3.4. Integrar los datos

Esta tarea perteneciente a la fase de preparación de los datos no se ejecutó, debido a que el dataset (en formato csv) fue generado directamente desde la UTI-UNL, por tal motivo, el autor de este proyecto no tuvo que unir diversas tablas o base de datos.

3.5. Formateo de los datos

En esta tarea se alteró el orden de cómo se encontraban las columnas de los registros en el dataset, ubicando la columna estado_matrícula al final.

4. Modelado

4.1. Seleccione la técnica de modelado

Las técnicas de machine learning que mejor se adaptan para dar cumplimiento a los objetivos del presente proyecto son: árboles de decisión, regresión logística y red neuronal, estos algoritmos se encuentran disponibles en la librería skilearn para Python.

4.2. Generar el diseño de la prueba

En esta tarea se dividió aleatoriamente los datos en tres grupos: primeramente, al conjunto inicial se dividió en el 80 % y el 20 %, donde el 20 % se estableció para la validación; se dividió nuevamente al 80 %, en 80 % y 20 %, y se obtuvo que el nuevo 80% es para el entrenamiento y el 20 % para test de los modelos. Para evaluar el rendimiento de los modelos entrenados, se utilizará la acurracy, recall y precisión, como métricas de evaluación. Estos indicadores se utilizan en los modelos de clasificación y se calculan directamente con los métodos implementados en la librería scikit-learn.

4.3. Construcción del Modelo

En esta fase se procede a ejecutar los algoritmos seleccionados, sobre sobre los datos de entrenamiento. En relación a los hiperparámetros de los algoritmos, se trabajó con los valores por defecto, ya que por medio de la técnica de ensayo y error el performance se mantuvo.

a) Modelo correspondiente al tercer ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 388 muestras, donde: 289 corresponden a la clase APRUEBA y 99 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 3** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 3 Resultados del entrenamiento del modelo del tercer ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[51 4] [16 7]]	0.744	0.636	0.304
RandomForestClassifier	[[52 3] [16 7]]	0.756	0.700	0.304
Neural network	[[51 4] [14 9]]	0.769	0.692	0.391

Los resultados que se exponen en la **Tabla 4** se obtuvieron aplicando el oversampling, ajustando al 45% de la clase mayoritaria. De este modo, se obtuvo un total de 419 muestras, donde: 289 corresponden a la clase APRUEBA y 130 a la clase REPRUEBA.

Tabla 4 Resultados del entrenamiento del modelo del tercer ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[50 4] [11 19]]	0.821	0.826	0.633
RandomForestClassifier	[[53 1] [12 18]]	0.845	0.947	0.600
Neural network	[[50 4] [11 19]]	0.821	0.826	0.633

b) Modelos correspondientes al cuarto ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 361 muestras, donde: 283 corresponden a la clase APRUEBA y 78 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 5** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 5 Resultados del entrenamiento del modelo del cuarto ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[42 0] [18 6]]	0.727	1.000	0.250
RandomForestClassifier	[[38 4] [12 12]]	0.758	0.750	0.500
Neural network	[[39 3] [16 8]]	0.712	0.727	0.333

Los resultados que se exponen en la **Tabla 6** se obtuvieron aplicando el oversampling, ajustando al 45% de la clase mayoritaria. De este modo, se obtuvo un total de 410 muestras, donde: 283 corresponden a la clase APRUEBA y 127 a la clase REPRUEBA.

Tabla 6 Resultados del entrenamiento del modelo del cuarto ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[51 1] [19 11]]	0.756	0.917	0.367
RandomForestClassifier	[[48 4] [12 18]]	0.805	0.818	0.600
Neural network	[[51 1] [18 12]]	0.768	0.923	0.400

c) Modelos correspondientes al quinto ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 326 muestras, donde: 260 corresponden a la clase APRUEBA y 66 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 7** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 7 Resultados del entrenamiento del modelo del quinto ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[47 1] [18 0]]	0.712	0.000	0.000
RandomForestClassifier	[[47 1] [18 0]]	0.712	0.000	0.000
Neural network	[[48 0] [18 0]]	0.727	0.000	0.000

Los resultados que se exponen en la **Tabla 7** se obtuvieron aplicando el oversampling, ajustando al 50% de la clase mayoritaria. De este modo, se obtuvo un total de 390 muestras, donde: 260 corresponden a la clase APRUEBA y 130 a la clase REPRUEBA.

Tabla 8 Resultados del entrenamiento del modelo del quinto ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[53 2] [19 4]]	0.731	0.667	0.174
RandomForestClassifier	[[54 1] [15 8]]	0.795	0.889	0.348
Neural network	[[54 1] [16 7]]	0.782	0.875	0.304

d) Modelos correspondientes al sexto ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 259 muestras, donde: 225 corresponden a la clase APRUEBA y 34 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 9**, se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 9 Resultados del entrenamiento del modelo del sexto ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[45 0] [7 0]]	0.865	0.000	0.000
RandomForestClassifier	[[43 2] [6 1]]	0.846	0.333	0.143
Neural network	[[44 1] [7 0]]	0.846	0.000	0.000

Los resultados que se exponen en la **Tabla 10** se obtuvieron aplicando el oversampling, ajustando al 50% de la clase mayoritaria. De este modo, se obtuvo un total de 337 muestras, donde: 225 corresponden a la clase APRUEBA y 112 a la clase REPRUEBA.

Tabla 10 Resultados del entrenamiento del modelo del sexto ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[47 1] [15 5]]	0.765	0.833	0.250
RandomForestClassifier	[[47 1] [12 8]]	0.809	0.889	0.400
Neural network	[[47 1] [13 7]]	0.794	0.875	0.350

e) Modelos correspondientes al séptimo ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 222 muestras, donde: 190 corresponden a la clase APRUEBA y 32 a la clase REPRUEBA. Las métricas que se exponen en la **Tabla 11** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 11 Resultados del entrenamiento del modelo del séptimo ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[41 0] [4 0]]	0.911	0.000	0.000
RandomForestClassifier	[[40 1] [4 0]]	0.889	0.000	0.000
Neural network	[[40 1] [4 0]]	0.889	0.000	0.000

Los resultados que se exponen en la **Tabla 12** se obtuvieron aplicando el oversampling, ajustando al 80% de la clase mayoritaria. De este modo, se obtuvo un total de 342 muestras, donde: 190 corresponden a la clase APRUEBA y 152 a la clase REPRUEBA.

Tabla 12 Resultados del entrenamiento del modelo del séptimo ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[38 1] [22 8]]	0.667	0.889	0.267
RandomForestClassifier	[[37 2] [18 12]]	0.710	0.857	0.400
Neural network	[[38 1] [22 8]]	0.667	0.889	0.267

f) Modelos correspondientes al octavo ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 198 muestras, donde: 168 corresponden a la clase APRUEBA y 30 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 13** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 13 Resultados del entrenamiento del modelo del octavo ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[31 1] [8 0]]	0.775	0.000	0.000
RandomForestClassifier	[[29 3] [7 1]]	0.750	0.250	0.125
Neural network	[[29 3] [7 1]]	0.750	0.250	0.125

Los resultados que se exponen en la **Tabla 14** se obtuvieron aplicando el oversampling, ajustando al 50% de la clase mayoritaria. De este modo, se obtuvo un total de 252 muestras, donde: 168 corresponden a la clase APRUEBA y 84 a la clase REPRUEBA.

Tabla 14 Resultados del entrenamiento del modelo del octavo ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[32 3] [9 7]]	0.765	0.700	0.438
RandomForestClassifier	[[32 3] [7 9]	0.804	0.750	0.562
Neural network	[[33 2] [7 9]]	0.824	0.818	0.562

g) Modelos correspondientes al noveno ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 184 muestras, donde: 157 corresponden a la clase APRUEBA y 27 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 15** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 15 Resultados del entrenamiento del modelo del noveno ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[33 0] [4 0]]	0.892	0.000	0.000
RandomForestClassifier	[[33 0] [4 0]]	0.892	0.000	0.000
Neural network	[[33 0] [4 0]]	0.892	0.000	0.000

Los resultados que se exponen en la **Tabla 16** se obtuvieron aplicando el oversampling, ajustando al 60% de la clase mayoritaria. De este modo, se obtuvo un total de 251 muestras, donde: 157 corresponden a la clase APRUEBA y 94 a la clase REPRUEBA.

Tabla 16 Resultados del entrenamiento del modelo del noveno ciclo de la CIS, aplicando el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[29 1] [12 9]]	0.745	0.900	0.429
RandomForestClassifier	[[29 1] [11 10]]	0.765	0.909	0.476
Neural network	[[29 1] [12 9]]	0.745	0.900	0.429

h) Modelos correspondientes al décimo ciclo.

Para entrenar el modelo correspondiente a este ciclo se utilizó un total de 165 muestras, donde: 125 corresponden a la clase APRUEBA y 45 a la clase REPRUEBA. Los resultados que se exponen en la **Tabla 17** se obtuvieron utilizando las muestras sin aplicar el oversampling.

Tabla 17 Resultados del entrenamiento del modelo del décimo ciclo de la CIS, sin aplicar el oversampling.

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[25 0] [8 0]]	0.758	0.000	0.000
RandomForestClassifier	[[25 0] [7 1]]	0.788	1.000	0.125
Neural network	[[25 0] [8 0]]	0.758	0.000	0.000

Los resultados que se exponen en la **Tabla 18** se obtuvieron aplicando el oversampling, ajustando al 60% de la clase mayoritaria. De este modo, se obtuvo un total de 200 muestras, donde: 125 corresponden a la clase APRUEBA y 75 a la clase REPRUEBA.

Tabla 18 Resultados del entrenamiento del modelo del décimo ciclo de la CIS, aplicando el oversampling

Model	Confusion matrix	Acurracy	Presicion	Recall
LogisticRegression	[[24 3] [6 7]]	0.775	0.700	0.538
RandomForestClassifier	[[23 4] [5 8]]	0.775	0.667	0.615
Neural network	[[23 4] [5 8]]	0.775	0.667	0.615

5. Evaluación

El proceso de validación que se realizó en la fase de construcción de los modelos permitió conocer el rendimiento de los modelos luego del entrenamiento de los mismos. En esta ocasión, el test también hace referencia al rendimiento de los modelos, pero con conjuntos de datos que no han sido utilizados durante el entrenamiento, ni la validación.

A continuación, se presenta los resultados obtenidos por cada uno de los modelos:

a) Test del modelo del tercer ciclo de la CIS

Tabla 19 Resultados del test del modelo del tercer ciclo de la CIS.

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.746	0.583	0.368

b) Test del modelo del cuarto ciclo de la CIS

Tabla 20 Resultados del test del modelo del cuarto ciclo de la CIS.

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.757	0.75	0.5

c) Test del modelo del quinto ciclo de la CIS

Tabla 21 Resultados del test del modelo del quinto ciclo de la CIS

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.793	0.8	0.421

d) Test del modelo del sexto ciclo de la CIS

Tabla 22 Resultados del test del modelo del sexto ciclo de la CIS

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.648	0.5	0.157

e) Test del modelo del séptimo ciclo de la CIS

Tabla 23 Resultados del test del modelo del séptimo ciclo de la CIS

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.6	0.705	0.413

f) Test del modelo del octavo ciclo de la CIS

Tabla 24 Resultados del test del modelo del octavo ciclo de la CIS

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.756	0.875	0.437

g) Test del modelo del noveno ciclo de la CIS

Tabla 25 Resultados del test del modelo del noveno ciclo de la CIS

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.525	0.428	0.166

h) Test del modelo del décimo ciclo de la CIS

Tabla 26 Resultados del test del modelo del décimo ciclo de la CIS

Model	Acurracy	Presicion	Recall
RandomForestClassifier	0.75	0.571	0.444

6. Despliegue

Para desplegar los diferentes modelos obtenidos se utilizó la plataforma Heroku, y se lo hizo mediante API'S en las que se realiza peticiones HTTP por medio del método POST e intercambiando datos en formato JSON. Esta fase no se explica a detalle, ya que no se encuentra especificada como objetivo. Pero en el trabajo final de titulación se expone algo similar al despliegue de los modelos.

7. Bibliografía

[1] UNL, “Universidad Nacional De Loja,” Universidad Nacional De Loja, 2017. [Online]. Available: https://www.unl.edu.ec/. [Accessed: 11-Jun-2020].

[2] E. M. Rojas, “Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo,” Rev. Ibérica Sist. e Tecnol. Informação, pp. 586–599, 2020.

[3] KDnuggets, “Python lidera las 11 principales plataformas de ciencia de datos y aprendizaje automático: tendencias y análisis,” Kdnuggets, 2019. [Online]. Available: https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html. [Accessed: 12-Jun-2020].

[4] A. Martini et al., “PyFitit: The software for quantitative analysis of XANES spectra using machine-learning algorithms,” Comput. Phys. Commun., vol. 250, May 2020, doi: 10.1016/j.cpc.2019.107064.

[5] N. Pilnenskiy and I. Smetannikov, “Feature selection algorithms as one of the python data analytical tools,” Futur. Internet, vol. 12, no. 3, Mar. 2020, doi: 10.3390/fi12030054.

[6] B. H. HemaMalini, L. Suresh, and M. Kushal, “Comprehensive Analysis of Students’ Performance by Applying Machine Learning Techniques,” in Smart Innovation, Systems and Technologies, 2020, vol. 160, pp. 547–556, doi: 10.1007/978-981-32-9690-9_60.

[7] C. Ruvinga, D. Malathi, and J. D. Dorathi Jayaseeli, “Human concentration level recognition based on vgg16 cnn architecture,” Int. J. Adv. Sci. Technol., vol. 29, no. 6 Special Issue, pp. 1364–1373, Apr. 2020.

[8] P. Kanani and M. Padole, “Deep learning to detect skin cancer using google colab,” Int. J. Eng. Adv. Technol., vol. 8, no. 6, pp. 2176–2183, Aug. 2019, doi: 10.35940/ijeat.F8587.088619.

[9] Kaggle, “Kaggle’s State of Data Science and Machine Learning 2019,” 2019.