National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Karen Pinzon
Student ID: x22144137

School of Computing
National College of Ireland

Supervisor: Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Karen Pinzon
Student ID:	x22144137
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	12/08/2024
Project Title:	Configuration Manual
Word Count:	711
Page Count:	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th September 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Karen Pinzon
x22144137

12/08/2024

1 Introduction

This document provides the directions required to replicate the results obtained for the project "Bias analysis in Machine Learning prediction for the secondary school exit exam in Colombia".

2 System Configuration

2.1 Hardware

For this project, the specifications of the local machine used are:

- System OS: Windows 11 Home
- Processor: AMD Ryzen 5 5625U with Radeon Graphics 2.30 GHz
- RAM: 16 GB
- System type: 64-bit operating system

2.2 Software

The project was implemented using Jupiter notebook version 7.0.8 through Anaconda Navigator 2.6.0. The python version used is Python 3.11.5.

2.3 Packages

The packages installed for this project are:

- numpy v1.23.2
- pandas v2.1.3
- matplotlib v3.8.2
- seaborn v0.14.0.dev0
- statsmodels v0.14.0

- datetime
- boruta v0.3
- sklearn v1.2.2
- lazypredict v0.2.12
- ltree v1.0.4
- scipy v1.11.3
- warnings

The exact libraries used are in Figure 3 along with the label explaining the purpose of each library used as part of the code.

3 Dataset

The name of the dataset is "Resultados únicos Saber 11" which means individual results for the Saber 11 exam. The dataset is 2900 Mb in size, as it contains 7.1M rows and 51 columns.

This dataset is available for public use at: `https://www.datos.gov.co/en/Educaci-n/Resultados-nicos-Saber-11/kgxf-xxbe/about_data`

To export the dataset, go to the link above, and click on export, then select the download tab, and last click on download. See Figure 1 as a reference.

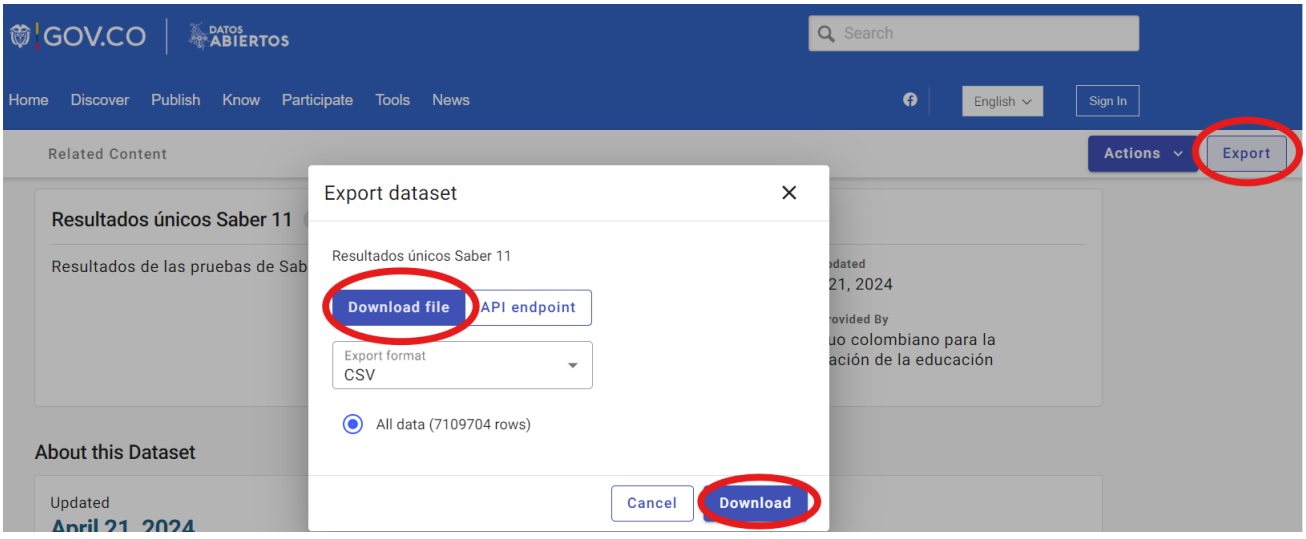

Figure 1: Export dataset steps

The final features are shown in Table 1

Table 1: Data dictionary - Boyaca Dataset

Column	Dtype	Description
AGE	int64	Student Age
HOME_PEOPLE	int64	No people student house
HOME_STRATUM	int64	Home class level
HOME_ROOMS	int64	No rooms home
SCORE_M-H	int64	Binary medium high score
COLE_AREA_UBICACION_URBANO	int32	Binary Urban school
COLE_CARACTER_TÉCNICO	int32	Binary school type technical
COLE_CARACTER_TÉCNICO/ACADÉMICO	int32	Binary school type tech/academic
COLE_GENERO_MASCULINO	int32	Binary all male school
COLE_GENERO_MIXTO	int32	Binary mixed gender school
COLE_JORNADA_MAÑANA	int32	Binary Morning school
COLE_JORNADA_NOCHE	int32	Binary Night school
COLE_JORNADA_SABATINA	int32	Binary Weekend School
COLE_JORNADA_TARDE	int32	Binary Afternoon school
COLE_JORNADA_UNICA	int32	Binary all day school
COLE_NATURALEZA_OFICIAL	int32	Binary public school
COLE_SEDE_PRINCIPAL_S	int32	Binary school main site
ESTU_GENERO_M	int32	Binary student gender male
ESTU_MCPIO_PRESENTACION_CHIQ	int32	Binary munic. Chiquinquira
ESTU_MCPIO_PRESENTACION_CHITA	int32	Binary munic. Chita
ESTU_MCPIO_PRESENTACION_DUITAMA	int32	Binary munic Duitama
ESTU_MCPIO_PRESENTACION_PAIPA	int32	Binary munic Paipa
ESTU_MCPIO_PRESENTACION_PUE BOYACÁ	int32	Binary munic Puerto Boyaca
ESTU_MCPIO_PRESENTACION_SOGAMOSO	int32	Binary munic Sogamoso
ESTU_MCPIO_PRESENTACION_TUNJA	int32	Binary munic Tunja
FAMI_EDUCACIONMADRE_Edu prof incompl	int32	Mother edu prof. complete
FAMI_EDUCACIONMADRE_Postgrado	int32	Mother edu postgrad
FAMI_EDUCACIONMADRE_Primaria completa	int32	Mother basic edu comp
FAMI_EDUCACIONMADRE_Primaria incompl	int32	Mother basic edu incomp
FAMI_EDUCACIONMADRE_Secundaria compl	int32	Mother edu high school comp
FAMI_EDUCACIONMADRE_Secundaria incompl	int32	Mother edu high school incomp
FAMI_EDUCACIONMADRE_Técnica compl	int32	Mother edu tecnichal
FAMI_EDUCACIONPADRE_Edu profes incompl	int32	Mother edu prof. incom
FAMI_EDUCACIONPADRE_Ninguno	int32	Father edu none
FAMI_EDUCACIONPADRE_Postgrado	int32	Father edu postgrad
FAMI_EDUCACIONPADRE_Primaria completa	int32	Father basic edu comp
FAMI_EDUCACIONPADRE_Primaria incompleta	int32	Father edu basic edu incomp
FAMI_EDUCACIONPADRE_Secundaria compl	int32	Father edu high school comp
FAMI_EDUCACIONPADRE_Secundaria incompl	int32	Father edu high school incomp
FAMI_EDUCACIONPADRE_Técnica compl	int32	Father education tecnichal
FAMI_TIENEAUTOMOVIL_Si	int32	Family has a car
FAMI_TIENECOMPUTADOR_Si	int32	Family has a laptop
FAMI_TIENEINTERNET_Si	int32	Family has internet access
FAMI_TIENELAVADORA_Si	int32	Family has washing machine

4 System preparation

The code to be used is "Gender Bias code_final.ipynb" and it runs in Jupyter notebook using Anaconda Navigator.

After downloading the csv file of the dataset, make sure to move the file to the same folder within Anaconda where the Jupyter file is in. This will ensure that the file is readable by the program without any modifications in the code as shown in Figure 2.

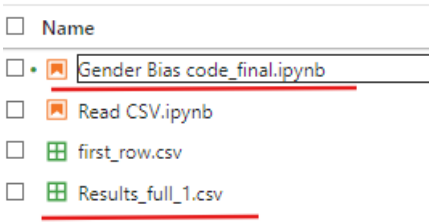

Figure 2: Import module

Once in Jupyter, install the libraries indicated in the section 2.3. The import module should look like as shown in Figure 3 below.

5 Code Sections

It is important to note that each section on the code allows to expand or compress the sections for easier access to each of them. An example of this can be seen in Figure 4. The code divides into six main sections:

- Data Gathering : Importing the dataset and create the Boyaca dataframe. Figure 5 shows how the loading cell of the code. Figure 6 shows how the Boyaca dataframe is created.
- Data Exploration: Tables and graphs to get to know the data. Figure 7 shows how the begining of the data exploration section, which shows missing values count. Then, Figure 8 illustrates how some of the graphs are obtained in the exploration part of the code.
- Data Transformation: Dealing with missing values, outliers, categorical variables, and others. An example of this can be observed at Figure 9 where the age column was fixed as there were ages that were not possible to get in this exam.
- Feature selection: Select the columns to be kept for the final stage of the process. In Figure 10 the partitions created and the Boruta code can be seen. There are different methods to use Boruta, this was the selection of the researcher.
- Model application: This is where the predictive models are applied to the dataset, and where some initial results are obtained. As this study contained different tests based on data, Figure 11 shows the different subsections in the model prediction code.
- Results: Graphical results in terms of images and tables that summarizes the output of the experiments.

```
# Supress Warnings

import warnings
warnings.filterwarnings('ignore')

# Import the numpy and pandas package
import numpy as np
#importing pandas to work with the dataframe
import pandas as pd

# Data Visualisation
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns
sns.set()

#Data plot
import statsmodels.api as sm

#datetime to fix the age column
from datetime import date

#feature selection
#!pip install Boruta
from boruta import BorutaPy
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier

#Lazy predict
#!pip install lazypredict
import lazypredict
from lazypredict.Supervised import LazyClassifier

#Logistic Regression Tree
#!pip install lrtree
from lrtree import Lrtree

#for the confusion matrix
from sklearn import metrics

#statistic analysis
import scipy.stats as stats
```

Figure 3: File and code confirmation

All the necessary libraries for each section are imported at the beginning in the import module. It is also important to note that the sections are dependant and sequential to each other, which means that the data exploration code will require for the data gathering code to be run and so on.

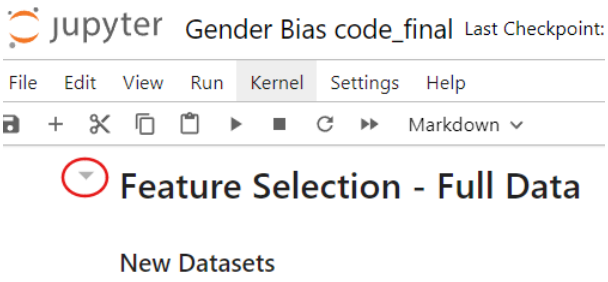

Figure 4: Expand sections of the code

Load data

Full Dataset

Opening the CSV file and loading it as a dataframe. Defining the type of data in two of the columns as it wasn't loading due to these two, as they're not being recognized as numbers.

```
Sab11_full = pd.DataFrame(pd.read_csv("Results_full_1.csv", dtype={"PUNT_INGLES": str, "PUNT_MATEMATICAS": str}))

#out Sab11_full df
```

Figure 5: Loading data module

Create Boyaca dataframe

One region of Colombia will be used for this analysis: Boyaca. Slicing the dataset to obtain only the rows from students who presented the exam in the region of Boyaca.

```
#in Sab11_full df
Boyaca = Sab11_full[Sab11_full['ESTU_COD_DEPTO_PRESENTACION'] == 15]
#output Boyaca df
```

Figure 6: Creating Boyaca dataframe

Explore data

This section is to get to know the data.

Starting with checking the amount of null values per column:

```
]: (boy_per.isnull().sum()).sort_values(ascending=False)
   #there are a number of missing values. I'll deal with them on the next section.
```

```
]: COLE_BILINGUE                  3269
   FAMI_ESTRATOVIVIENDA           1818
   FAMI_TIENEINTERNET             1660
   FAMI_EDUCACIONPADRE            1622
   FAMI_EDUCACIONMADRE            1618
   FAMI_CUARTOSHOGAR              1219
   FAMI_TIENEAUTOMOVIL            1197
   FAMI_TIENELAVADORA             1184
   FAMI_TIENECOMPUTADOR           1184
   FAMI_PERSONASHOGAR             1166
   COLE_CARACTER                   540
   PUNT_INGLES                       3
   DESEMP_INGLES                     3
   ESTU_NACIONALIDAD                 0
   ESTU_PRIVADO_LIBERTAD             0
   ESTU_PAIS_RESIDE                  0
   ESTU_MCPIO_RESIDE                 0
   ESTU_MCPIO_PRESENTACION           0
```

Figure 7: Missing value count

Plotting the counts of some of the columns with the higher missing values and the global score

```
Bilischool_hist= px.histogram(boy_per, x='COLE_BILINGUE', title="Bilingual school",barmode='group', width=600, height=350)
Bilischool_hist.update_layout(bargap=0.2)
Bilischool_hist.show()

Schoolarea_hist= px.histogram(boy_per, x='COLE_AREA_UBICACION', title="School location", barmode='group', width=600, height=350)
Schoolarea_hist.update_layout(bargap=0.2)
Schoolarea_hist.show()

Homeclass_hist= px.histogram(boy_per, x='FAMI_ESTRATOVIVIENDA', title="Home class level", barmode='group', width=600, height=350)
Homeclass_hist.update_layout(bargap=0.2)
Homeclass_hist.show()

Father_ed_hist= px.histogram(boy_per, x='FAMI_EDUCACIONPADRE', title="Father's level of education", barmode='group', width=600, height=350)
Father_ed_hist.update_layout(bargap=0.2)
Father_ed_hist.show()

Mother_ed_hist= px.histogram(boy_per, x='FAMI_EDUCACIONMADRE', title="Mother's level of education", barmode='group', width=600, height=350)
Mother_ed_hist.update_layout(bargap=0.2)
Mother_ed_hist.show()
```

Figure 8: Data Exploration graphs

As the most repeated values are between 15 and 21, all values below 15 will be approximated to the average value 15-21 which is 18.

```
: #function to transform age into average age:
def age_18(x):
    if  x < 15:
        return 18
    elif x >= 15:
        return x
    else:
        return "NA"
```

Figure 9: Age column transformation

```
# seperate input and output variables
X_mh = boy_med_high.drop("SCORE_M-H", axis = 1)
y_mh = boy_med_high["SCORE_M-H"]
```

```
#taken from https://www.kaggle.com/code/yasinnsariyildiz/feature-selection-with-borutapy

rfc_mh = RandomForestClassifier(random_state=37, n_estimators=1000, max_depth=5)
boruta_selector_mh = BorutaPy(rfc_mh, n_estimators='auto', verbose=2, random_state=37)
boruta_selector_mh.fit(np.array(X_mh), np.array(y_mh))

print("Ranking: ",boruta_selector_mh.ranking_)
print("No. of significant features: ", boruta_selector_mh.n_features_)
```

Figure 10: Boruta code

```
# Create a df with each gender's data
boy_male = boy_med_high[boy_med_high['ESTU_GENERO_M'] == 1]
boy_female = boy_med_high[boy_med_high['ESTU_GENERO_M'] == 0]
```

Mixed Gender

+ 14 cells hidden

Male Gender

+ 13 cells hidden

Female Gender

+ 13 cells hidden

Mixed testing

Testing with male training data and female testing data to see the accuracy.

+ 31 cells hidden

Figure 11: Gender sections for ML application