Campus Recruitment Prediction Report

Name: Eduardo Williams

Student Code: C0896405

Date: June 26, 2024

Contents

Tables

Figures

Dataset Selection

The placement of students is one of the most important objectives of an educational institution. The model's main goal is to predict whether the student will be recruited in campus placements or not, based on the available factors in the dataset.

Dataset was gotten from Kaggle:

https://www.kaggle.com/c/ml-with-python-course-project/data

Dataset schema

Dataset had originally the following schema:

Column name	Type	Observation
sl_no	int64	anonymous unique id
Gender	int64	categorical
ssc_p	float64	numerical
ssc_b	object	categorical
hsc_p	float64	numerical
hsc_b	object	categorical
hsc_s	object	categorical
degree_p	float64	numerical
degree_t	object	categorical
workex	object	categorical
etest_p	float64	numerical
specialisation	object	categorical
mba_p	float64	numerical
status	object	TARGET
salary	float64	numerical

Table 1 Dataset Schema

Dataset factsheet

Description	Value
Rows	**215**
Columns	**15**
Target column	**status**
Target categories	**Placed / Not Placed**

Table 2 Dataset factsheet

Data Preprocessing

This stage performs the following steps:

- Dealing with null values

  Null vales were found just in Salary column. They were replaced by zeroes.

- Removing duplicate rows

- Dropping useless columns

  According to Kaggle **sl_no** column is an anonymous unique id, therefore it is useless for analysis.

  Additionally, Salary column was deleted.

Encoding

Target column (status) was encoded according to Table 3:

Label	Value
Placed	**1**
Not Placed	**0**

Table 3 Target encoding

One hot encoding was applied to **hsc_s** and **degree_t** categorical columns, since they had more than two possible values.

On the other hand, Label encoding was applied to **workex**, **ssc_b**, **hsc_b** and **specialization** categorical columns, since they had just two possible values.

StandardScaler

StandardScaler was applied to **ssc_p**, **hsc_p**, **degree_p**, **etest_p**, **mba_p**, as they were numerical continuous columns.

Splitting dataset

The dataset was split into **train** (70%) and **test** (30%) using **random_state** parameter set to 42.

Selecting features

RFE technique was used to determine the ten most relevant columns. According to this, those columns were: **gender**, **ssc_p**, **hsc_p**, **hsc_b**, **degree_p**, **workex**, **specialisation**, **mba_p**, **hsc_s_Arts** and **degree_t_Comm&Mgmt**. Therefore, **ssc_b**, **etest_p**, **hsc_s_Commerce**, **hsc_s_Science**, **degree_t_Others**, **degree_t_Sci&Tech** columns were dropped.

Model Selection, Train and Evaluation

Since the target column was binary categorical, I considered the following models to approach this problem:

Logistic Regression

This model was selected since it is highly suitable for binary classification problems.

Support Vector Machine – SVC

Support Vector Machine has two models SVR and SVC, for regression and classification problems accordingly, so for this problem SVC was used. In addition, this model is suitable for small datasets.

K-Nearest Neighbors

The KNN was chosen because it is one of the simplest models and has good performance when working with small datasets.

Model Training

Logistic regression and **Support Vector Machine** models were fitted with their default parameters. However, **K-Nearest Neighbors** was fitted with **n_neighbors** parameter set to 5.

Following metrics were generated upon perform predictions: **r2_score**, **mean_absolute_error**, **mean_squared_error**, **accuracy_score**, **confusion_matrix** and **classification_report**.

Grid Search Cross-Validation

After evaluating, Grid search cross-validation was performed with the following hyper-parameters:

Logistic regression

Hyper-parameter	Values
solver	newton-cg, lbfgs, liblinear, sag, saga
penalty	l1, l2, elasticnet, None
C	0.001, 0.01, 0.1, 1, 10, 100

Table 4 Logistic regression hyper-parameters

Support Vector Machine

Hyper-parameter	Values
gamma	1, 0.1, 0.01, 0.001, 0.0001
kernel	rbf
C	0.001, 0.01, 0.1, 1, 10, 100

Table 5 Support Vector Machine hyper-parameters

K-Nearest Neighbors

Hyper-parameter	Values
n_neighbors	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29

Table 6 K-Nearest Neighbors hyper-parameters

As a result, Logistic regression was determined as the best model. However, it must be fitted with the following hyper-parameters:

Hyper-parameter	Value
solver	liblinear
penalty	l2
C	1

Table 7 Logistic regression best hyper-parameters

With these hyper-parameters the following confusion matrix was gotten:

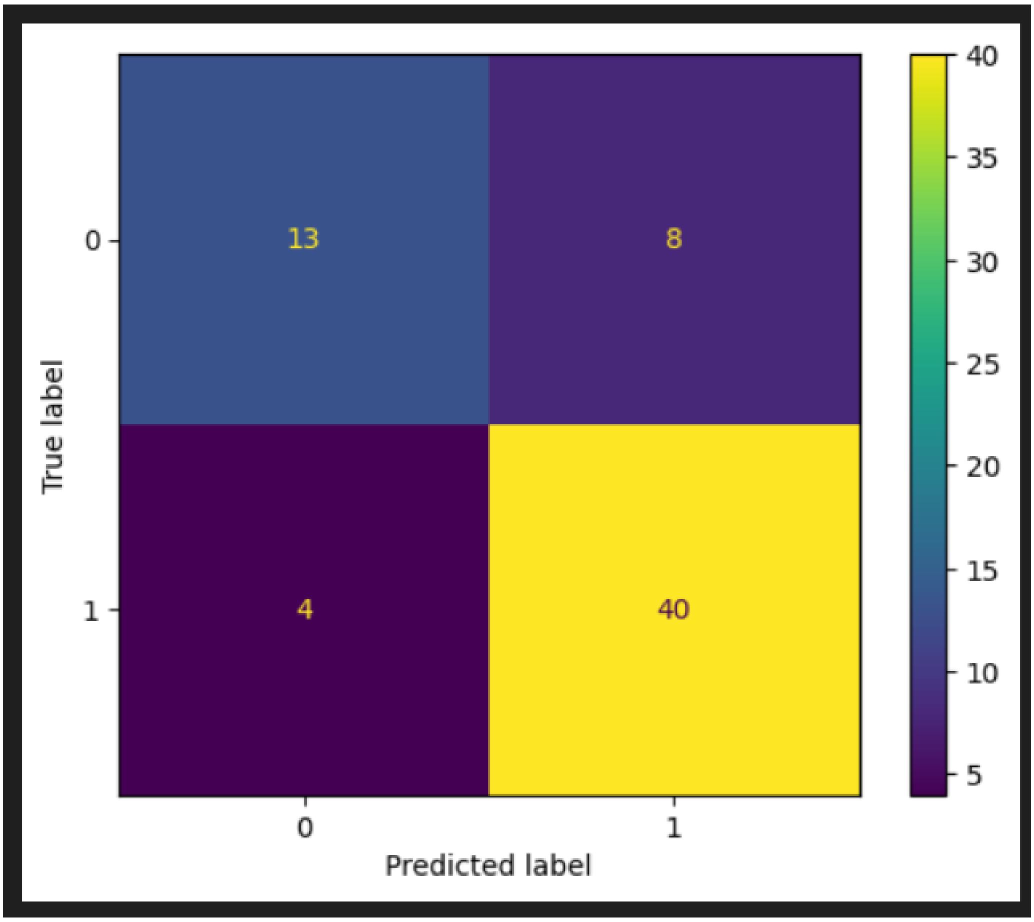

Figure 1 Logistic regression confusion matrix

Voting Classifier

Finally, Voting Classifier was performed, however similar accuracy was gotten.