Detail project report

CAMPUS PLACEMENT PREDICTION

Table of Contents

[1) Objective 3](#_Toc115160323)

[2) Benefits 3](#_Toc115160324)

[3) Data Sharing Agreement 3](#_Toc115160325)

[4) Architecture 4](#_Toc115160326)

[5) Model Training 5](#_Toc115160327)

[5.1) Data Preprocessing 5](#_Toc115160328)

[5.2) Model Training Part 5](#_Toc115160329)

[5.3) Prediction 6](#_Toc115160330)

[6) Q & A 6](#_Toc115160331)

# 

# 1) Objective

Development of a predictive model which can predict whether the Student will get placed or not placed.

# 2) Benefits

1) It will help Colleges to keep a track of the students and utilize resources well in training them for placements

2)They can have the detailed overview of each candidate and can help in a better way to prepare for placements.

3) It will save the time as compared to be done in manual way.

# 3) Data Sharing Agreement

* File name can be anything
* Columns inside the file will be 10
* Columns are [sl\_no,gender,ssc\_p,ssc\_b,hsc\_p,hsc\_b,degree\_p,degree\_t,workex,etest\_p,specialization, mba\_p, status, salary]
* Columns datatypes are [int,int,float,object,float,object,float,object,object,float,object,float,object,float]

# 4) Architecture

# 5) Model Training

## 5.1) Data Preprocessing

1. First I checked whether is there any null values present inside the data and I found that there is a single row only that’s why I directly removed it.
2. After removing the null values Now, My main task was to handle categorical data and the data was mostly categorical. I use both One hot encoding and Label Encoding for different columns as per requirements.
3. I have used Label Encoding for columns workex, specialization and status
4. I have used One Hot Encoding for hsc\_s and degree\_t.
5. Once done the encoding I removed these columns.
6. I have also removed salary and sl\_no column as they are providing not much value to the data.
7. I found outliers in hsc\_p so I handled them using IQR.

## 5.2) Model Training Part

* In model training I tried a few algorithms like XGBoost, and Random Forest, Decision Tree, Logistic Regression because I know that they give the best score than others.
* I have written my code in such a manner that if the training starts then automatically the best performing model among those will be chosen and on the basis of it hyperparameter tuning will happen using RandomizedSearchCV and finally the best model will be created.
* I saved my model as best\_model.pickle.
* Now I created an API for my model using Flask.
* Finally My project is created and I am going to deploy it to the Heroku



## 5.3) Prediction

1) Now its time for prediction part, First when I will get the file location for testing, I am validating each file with the testing schema mentioned .

2) Data preprocessing is performed same as while training, I am handling the categorical column with a pickle file which I had stored “handle\_categorical\_col\_for\_testing.pickle”.

3) Now everything is just simple I am loading my pickle model and applying model.predict

4) The output file is then copied to the file location provided by the client and with that the log file is also copied.

# 6) Q & A

1) What is the Source of Data?

Ans: Kaggle is the source of the data link <https://www.kaggle.com/c/ml-with-python-course-project/data>

2) What was the type of Data?

Ans: Data type of columns are float, string and integers

3) How logs are managed?

Ans: I am managing the logs using logging module. After the prediction is completed the log file is shared with the client, if in case exception occurs inside the program then also the log file is shared to the file location provided.

4) What techniques are you using for data preprocessing?

Ans: See Data Preprocessing above I have mentioned there in detail.

5) How training was done?

Ans: RandomForest, XGBoost and Decision Tree are applied and out of this whichever performed best is used for model building.

6) What are stages of Deployment?

Ans: Deployment has been done to Heroku Web Services