CHIKKANNA GOVERNMENT ARTS COLLEGE

TIRUPUR-641602

(AFFILIATED TO BHARATHIAR UNIVERSITY)

TEAM MEMBERS NAME : SARANRAJ A(2022K0050)

SANTHOSKUMAR S(2022K0049)

GOWTHAM K(2022K0022)

SURYA R(2022K0056)

NAGARAJ S(2022K0036)

DEPARTMENT OF COMPUTER SCIENCE

CHIKKANNA GOVERNMENT ARTS COLLEGE

NAAN MUDHALVAN PROJECT WORK

(AFFILIATED TO BHARATHIAR UNIVERSITY)

TIRUPUR-641602

TITLE :Identifying patterns and Trends in campus placement data using machine learning

This is to certify that this is a bonafide record of work done by the above students of III B.Sc (CS) Degree **NAAN MUDHALVAN PROJECT** during the year …….

Submitted for the Naan Mudhalvan project work held on _____20

CLASS TUTOR HEAD OF THE DEPARTMENT

INDEX

PROBLEM SELECTION

Collect the data: **You** need to gather campus placement data from different colleges and universities. The data should include various parameters such as the student's academic performance, their skills, the company's requirements, the placement offers, and other relevant information.Preprocess the data: The data collected needs to be preprocessed by removing any missing values, outliers, and other noise in the data. You also need to normalize or scale the data so that it can be easily fed into the machine learning algorithm.Define the problem: Identify the problem you want to solve using machine learning. For example, you may want to predict the likelihood of a student getting placed based on their academic performance and other parameters.Select the appropriate machine learning algorithm: There are several machine learning algorithms available such as Linear Regression, Logistic Regression, Decision Trees, Random Forest, etc. Choose the algorithm that best suits your problem.Train the model: Once you have selected the algorithm, you need to train the model using the preprocessed data. Split the data into training and testing sets and use the training set to train the model.

- Evaluate the model: After training the model, evaluate its performance using the testing set. Use metrics such as

accuracy, precision, recall, and F1-score to evaluate the model's performance.

- Interpret the results: Analyze the results to identify any patterns or trends in the data. You can use technique

IDEATION

Predicting Placement Offers: Use machine learning algorithms to predict which students are likely to receive placement offers based on their academic performance, skills, and other parameters. This can help colleges and universities to identify which students need additional support to increase their chances of getting placed.Identifying Factors Affecting Placement: Use machine learning algorithms to identify which factors such as academic performance, skills, personality traits, and other parameters are affecting students' placement offers. This can help colleges and universities to provide targeted support to improve students' chances of getting placed.Matching Students with Companies: Use machine learning algorithms to match students' profiles with companies' requirements.

This can help companies to find the right candidates for their open positions, and students to find jobs that match their skills and interests.Identifying Campus Recruitment Trends: Use machine learning algorithms to identify trends in campus recruitment, such as which companies are hiring the most, which sectors are popular, and which locations are attracting the most recruitment. This can help colleges and universities to plan their placement activities accordingly.Predicting Future Demand: Use

machine learning algorithms to predict future demand for certain skills and jobs based on industry trends, economic indicators, and other parameters. This can help colleges and universities to adapt their curricula and training programs to meet the needs of the job market.

Predicting Salary Offers: Use machine learning algorithms to predict salary offers for different job roles based on industry trends, location, and other factors. This can help students negotiate better salaries and make informed decisions about job offers.

Identifying Skill Gaps: Use machine learning algorithms to identify skill gaps among students and recommend training programs to bridge those gaps. This can help colleges and universities to better prepare students for the job market and improve their placement rates.

Analyzing Alumni Data: Use machine learning algorithms to analyze alumni data and identify trends in their career paths, salaries, and job satisfaction.

This can help colleges and universities to better understand the impact of their education and training programs and improve them accordingly.

REQUIREMNT ANALYSIS

- Identify the stakeholders: The first step is to identify the stakeholders involved in the campus placement process, such as colleges, universities, students, recruiters, and alumni.
- Define the problem: Identify the problem you want to solve using machine learning. For example, you may want to predict the likelihood of a student getting placed based on their academic performance and other parameters.
- Collect the data: Identify the data sources and types of data needed to solve the problem. For example, you may need data on student academic performance, skills, personality traits, job offers, and other relevant information.
- Preprocess the data: Preprocess the data by removing any missing values, outliers, and other noise in the data. Normalize or scale the data so that it can be easily fed into the machine learning algorithm.
- Select the appropriate machine learning algorithm: Choose the machine learning algorithm that best suits the problem you want to solve. For example, you may choose a decision tree algorithm to identify factors affecting placement.
- Train the model: Train the machine learning model using the preprocessed data. Split the data into training and testing sets and use the training set to train the model.

- Evaluate the model: Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score.
- Interpret the results: Analyze the results to identify any patterns or trends in the data. Use techniques such as feature importance or visualization to interpret the results.
- Implement the solution: Implement the machine learning solution by integrating it with existing campus placement systems or building a new system.
- Monitor and maintain the system: Monitor the performance of the machine learning system and make any necessary adjustments to ensure it continues to provide accurate and reliable results.

PROJECT DESIGN

- Project Overview: Provide an overview of the project, including the problem statement, objectives, and expected outcomes.
- Data Collection: Collect data from relevant sources such as college and university placement offices, alumni associations, and job portals. The data may include academic performance, skills, personality traits, job offers, and other relevant information.

- Data Preprocessing: Preprocess the data by cleaning, filtering, and transforming the data. This may involve removing missing values, handling outliers, and scaling or normalizing the data.
- Feature Engineering: Engineer new features from the existing data that may improve the machine learning algorithm's performance. This may include combining or transforming existing features or creating new features.
- Machine Learning Algorithm Selection: Choose the appropriate machine learning algorithm for the problem at hand. For example, you may choose a decision tree algorithm to identify factors affecting placement.
- Model Training: Split the preprocessed data into training and testing sets and train the machine learning model using the training set. Evaluate the model's performance on the testing set and make any necessary adjustments to improve its accuracy.
- Results Interpretation: Interpret the results of the machine learning model to identify any patterns or trends in the data. This may involve using techniques such as feature importance or visualization.
- System Integration: Integrate the machine learning solution with existing campus placement systems or build a new system. This may involve developing a web application or API that can be used by stakeholders.
- System Testing: Test the machine learning solution thoroughly to ensure it is working as intended and providing accurate and reliable results.

- Deployment and Maintenance: Deploy the machine learning solution in a production environment and monitor its performance. Make any necessary adjustments to ensure it continues to provide accurate and reliable results.

PROJECT PLANNING PHASE

- Gather data: Once you have defined the problem, you need to gather the relevant data that will help you to develop a machine learning model. This data may include information about past campus placements, job openings, candidate profiles, and more
- Preprocess the data: Before you can use the data to train a machine learning model, you need to preprocess it. This involves cleaning the data, removing any outliers, and transforming the data into a format that can be easily fed into a machine learning algorithm.
- Choose a machine learning algorithm: There are many different machine learning algorithms that you can use to analyze campus placement data. Some popular options include decision trees, random forests, and neural networks. Choose an algorithm that is appropriate for your data and your problem.

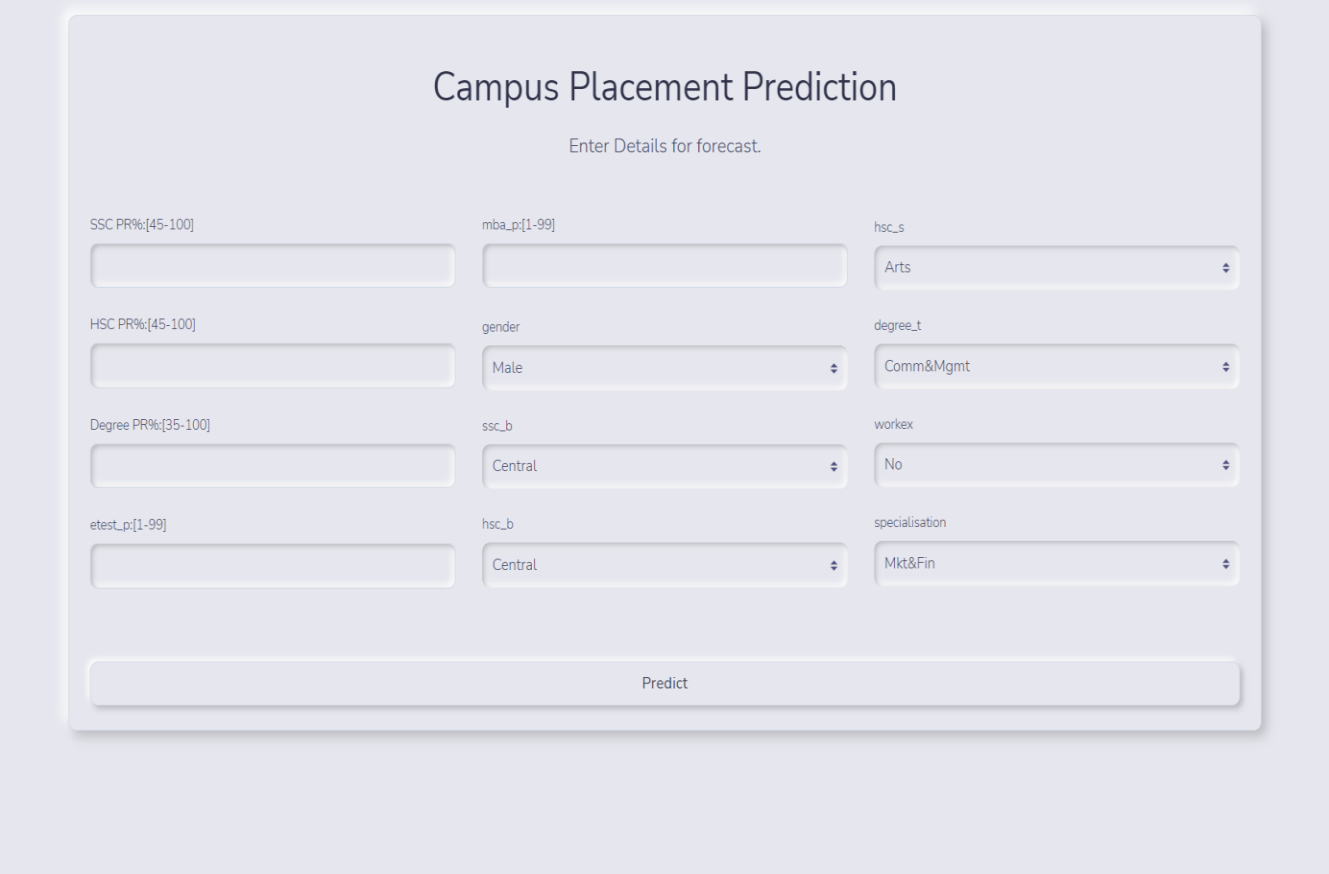
- Train the model: Once you have chosen a machine learning algorithm, you need to train the model using your preprocessed data. This involves splitting the data into training and testing sets, running the algorithm on the training set, and evaluating its performance on the testing set.
- Fine-tune the model: After you have trained the model, you may need to fine-tune it to improve its accuracy. This can involve tweaking the algorithm's hyperparameters, experimenting with different feature engineering techniques, or using a different algorithm altogether.
- Deploy the model: Once you have a machine learning model that accurately identifies pattern trends in campus placement data, you can deploy it to make predictions on new data. This may involve integrating the model into a larger software system or building a web application that allows users to interact with the model directly.
- Evaluate the results: Finally, you need to evaluate the results of your machine learning model to see how well it performs in the real world. This can involve monitoring its accuracy over time, collecting feedback from users, and iterating on the model to improve its performance.

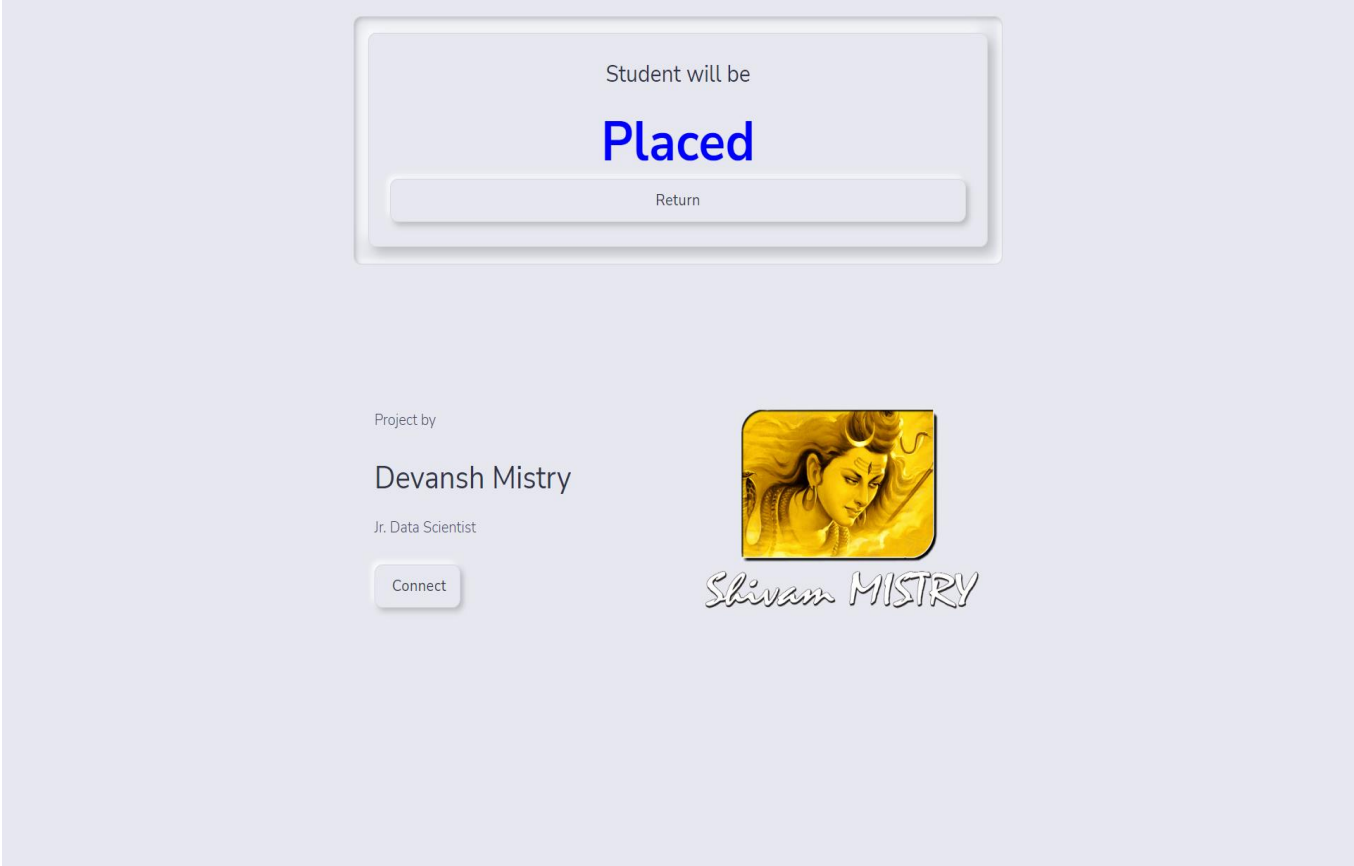
PROJECT

DEVELOPMENT PHASE

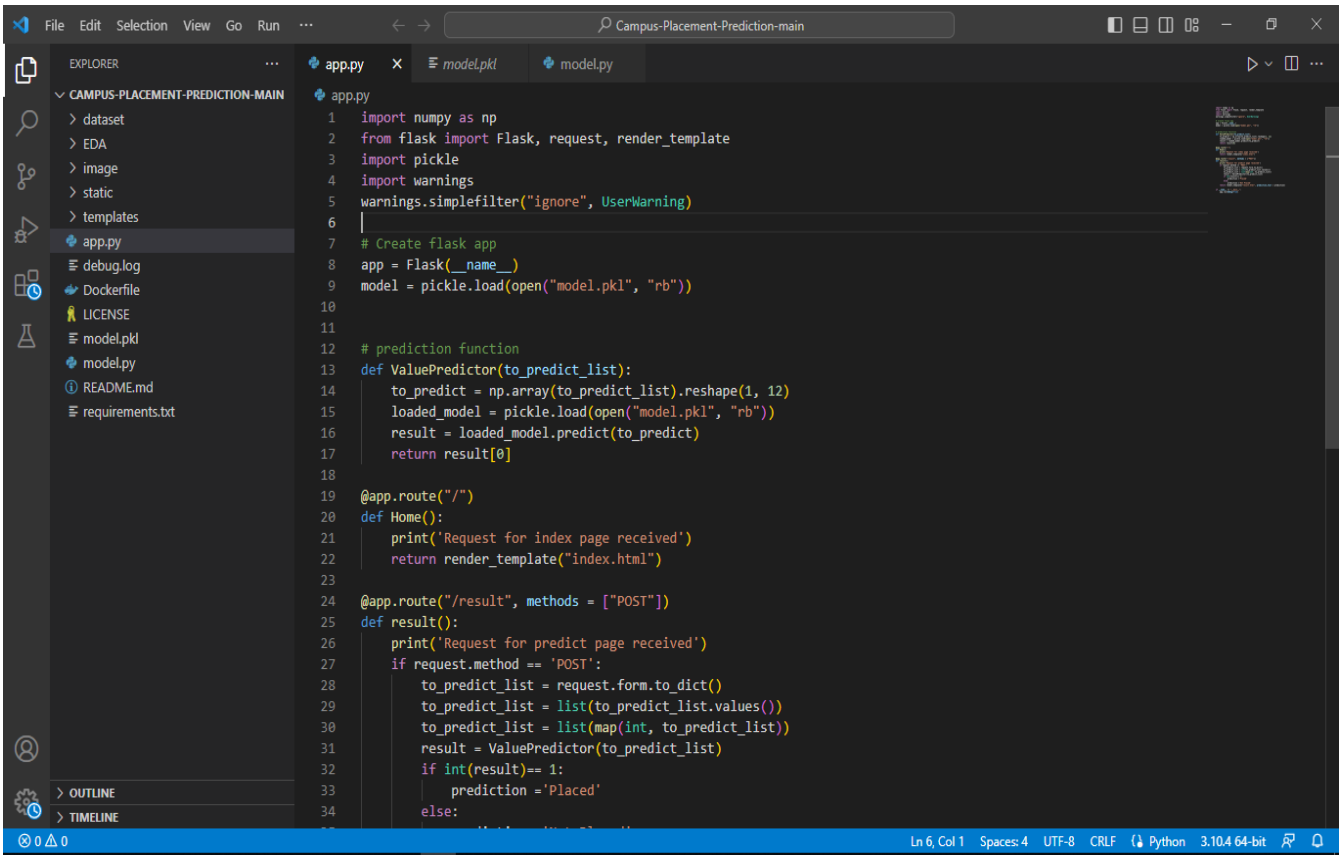
- Data Collection: Collecting relevant data related to campus placements, job openings, candidate profiles, and more.
- Data Preprocessing: Cleaning, transforming, and organizing the data into a format that can be easily fed into a machine learning algorithm.
- Feature Engineering: Selecting and engineering relevant features for the machine learning algorithm, such as the candidate's academic performance, work experience, skill set, and more.
- Model Selection: Choosing an appropriate machine learning algorithm that can be trained and optimized for the given data. Popular options for this project may include decision trees, random forests, or neural networks.
- Model Training: Using the preprocessed data and selected machine learning algorithm to train a model that can identify pattern trends and analyze campus placement data.
- Model Evaluation: Testing and validating the model's accuracy, precision, and recall by comparing the predicted results with actual placement outcomes.
- Model Tuning: Tweaking the algorithm's hyperparameters and feature engineering techniques to improve the model's performance.
- Deployment: Integrating the model into a software system, building a web application, or providing an API service that allows users to interact with the model directly.
- Maintenance: Continuously monitoring the model's performance, collecting feedback from users, and updating the model as needed to improve its accuracy.

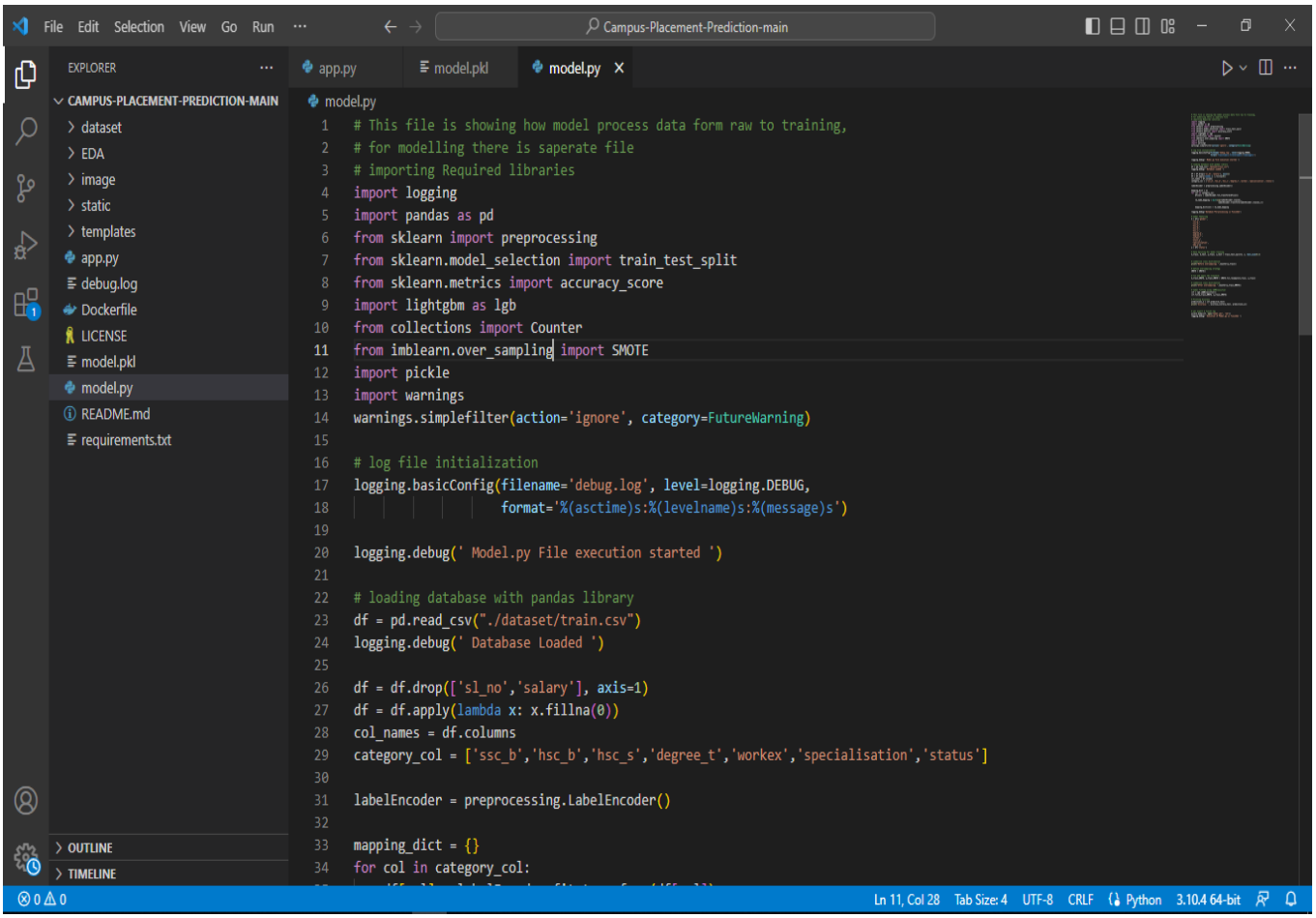
RESULT


Campus Placement Prediction
Enter Details for forecast.
SSC PR%:[45-100]
mba_p:[1-99]
hsc_s
Arts
HSC PR%:[45-100]
gender
Male
degree_t
Comm&Mgmt
Degree PR%:[35-100]
ssc_b
Central
workex
No
etest_p:[1-99]
hsc_b
Central
specialisation
Mkt&Fin
Predict



SAMPLE CODING


File Edit Selection View Go Run ···
Campus-Placement-Prediction-main
EXPLORER
app.py
model.pkl
model.py
CAMPUS-PLACEMENT-PREDICTION-MAIN
dataset
EDA
image
static
templates
app.py
debug.log
Dockerfile
LICENSE
model.pkl
model.py
README.md
requirements.txt
OUTLINE
TIMELINE
import numpy as np
from flask import Flask, request, render_template
import pickle
import warnings
warnings.simplefilter("ignore", UserWarning)
# Create flask app
app = Flask(__name__)
model = pickle.load(open("model.pkl", "rb"))
# prediction function
def ValuePredictor(to_predict_list):
to_predict = np.array(to_predict_list).reshape(1, 12)
loaded_model = pickle.load(open("model.pkl", "rb"))
result = loaded_model.predict(to_predict)
return result[0]
@app.route("/")
def Home():
print('Request for index page received')
return render_template("index.html")
@app.route("/result", methods = ["POST"])
def result():
print('Request for predict page received')
if request.method == 'POST':
to_predict_list = request.form.to_dict()
to_predict_list = list(to_predict_list.values())
to_predict_list = list(map(int, to_predict_list))
result = ValuePredictor(to_predict_list)
if int(result)== 1:
prediction ='Placed'
else:
0 0
Ln 6, Col 1 Spaces: 4 UTF-8 CRLF Python 3.10.4 64-bit


File Edit Selection View Go Run
Campus-Placement-Prediction-main
EXPLORER
CAMPUS-PLACEMENT-PREDICTION-MAIN
dataset
EDA
image
static
templates
app.py
debug.log
Dockerfile
LICENSE
model.pkl
model.py
README.md
requirements.txt
OUTLINE
TIMELINE
app.py
model.pkl
model.py
model.py
# This file is showing how model process data form raw to training,
# for modelling there is saperate file
# importing Required libraries
import logging
import pandas as pd
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import lightgbm as lgb
from collections import Counter
from imblearn.over_sampling import SMOTE
import pickle
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
# log file initialization
logging.basicConfig(filename='debug.log', level=logging.DEBUG,
format='%(asctime)s:%(levelname)s:%(message)s')
logging.debug(' Model.py File execution started ')
# loading database with pandas library
df = pd.read_csv("./dataset/train.csv")
logging.debug(' Database Loaded ')
df = df.drop(['sl_no','salary'], axis=1)
df = df.apply(lambda x: x.fillna(0))
col_names = df.columns
category_col = ['ssc_b','hsc_b','hsc_s','degree_t','workex','specialisation','status']
labelEncoder = preprocessing.LabelEncoder()
mapping_dict = {}
for col in category_col:
0 0
Ln 11, Col 28 Tab Size: 4 UTF-8 CRLF Python 3.10.4 64-bit