***SOURCE CODE:***

# 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:

df[col] = labelEncoder.fit\_transform(df[col])

le\_name\_mapping = dict(zip(labelEncoder.classes\_, labelEncoder.transform(labelEncoder.classes\_)))

mapping\_dict[col] = le\_name\_mapping

logging.debug('Database Pre-processing is Finished') # model featuring

X = df[['gender', 'ssc\_p',

'ssc\_b',

'hsc\_p',

'hsc\_b',

'hsc\_s', 'degree\_p', 'degree\_t', 'workex', 'etest\_p',

'specialisation', 'mba\_p']]

y = df['status']

# Data Spliting For model training

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

# summarize class distribution

print("Before oversampling: ",Counter(y\_train))

# define oversampling strategy SMOTE = SMOTE()

# fit and apply the transform

X\_train\_SMOTE, y\_train\_SMOTE = SMOTE.fit\_resample(X\_train, y\_train)

# summarize class distribution

print("After oversampling: ",Counter(y\_train\_SMOTE))

# model fitting using LGBMClassifier clf = lgb.LGBMClassifier()

clf.fit(X\_train\_SMOTE, y\_train\_SMOTE)

# Printing Accuracy

predictions\_e = clf.predict(X\_test)

print('Accuracy: ', accuracy\_score(y\_test, predictions\_e))

# pkl export & finish log

pickle.dump(clf, open("model.pkl", "wb")) logging.debug(' Execution of Model.py is finished ')

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:

prediction ='Not Placed'

return render\_template("result.html", prediction\_text = prediction)

if \_\_name == " main ": app.run(debug=True)

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Campus Placement Prediction</title>

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to- fit=no">

<meta name="title" content="Campus Placement Prediction">

<meta name="author" content="Devansh">

<link rel="stylesheet" href="{{ url\_for('static', filename='css/neumorphism.css')

}}">

<link rel="javascript" href="{{ url\_for('static', filename='js/neumorphism.js') }}">

<link rel="shortcut icon" href="{{ url\_for('static', filename='favicon.ico') }}">

</head>

<body>

<main>

<section class="container-fluid py-3">

<div class="row justify-content-md-around">

<div class="col-11">

<div class="card bg-primary shadow-soft border-light px-4 py-4">

<div class="card-header pb-0 text-center">

<h2 class="h1 mb-3">Campus Placement Prediction</h2>

<p class="mb-5 lead"> Enter Details for forecast.

</p>

</div>

<form name="frmregi" action="/result" method="POST">

<!-- Form -->

<div class="row mb-4 mb-lg-5">

<div class="col-lg-4 col-sm-6">

<div class="form-group">

<label for="ssc\_p">SSC PR%:[45-100]</label>

<div class="input-group mb-4">

<input class="form-control" type="number" id="ssc\_p" name="ssc\_p" required min="45" max="100">

</div>

</div>

<div class="form-group">

<label for="hsc\_p">HSC PR%:[45-100]</label>

<div class="input-group mb-4">

<input class="form-control" type="number" id="hsc\_p" name="hsc\_p" required min="45" max="100">

</div>

</div>

<div class="form-group">

<label for="degree\_p">Degree PR%:[35-100]</label>

<div class="input-group mb-4">

<input class="form-control" type="number" id="degree\_p" name="degree\_p" required min="35" max="100">

</div>

</div>

<div class="form-group">

<label for="etest\_p">etest\_p:[1-99]</label>

<div class="input-group mb-4">

<input class="form-control" type="number" id="etest\_p" name="etest\_p" required min="0" max="100">

</div>

</div>

</div>

<div class="col-lg-4 col-sm-6">

<div class="form-group">

<label for="mba\_p">mba\_p:[1-99]</label>

<div class="input-group mb-4">

<input class="form-control" type="number" id="mba\_p" name="mba\_p" required min="0" max="100">

</div>

</div>

<div class="form-group">

<label class="my-1 mr-2" for="gender">gender</label>

<select class="custom-select my-1 mr-sm-2" id="gender" name="gender" required>

<option value="0" selected>Male</option>

<option value="1">Female</option>

</select>

</div>

<div class="form-group">

<label class="my-1 mr-2" for="ssc\_b">ssc\_b</label>

<select class="custom-select my-1 mr-sm-2" id="ssc\_b"

name="ssc\_b" required>

<option value="0" selected>Central</option>

<option value="1">Others</option>

</select>

</div>

<div class="form-group">

<label class="my-1 mr-2" for="hsc\_b">hsc\_b</label>

<select class="custom-select my-1 mr-sm-2" id="hsc\_b"

name="hsc\_b" required>

<option value="0" selected>Central</option>

<option value="1">Others</option>

</select>

</div>

</div>

<div class="col-lg-4 col-sm-6">

<div class="form-group">

<label class="my-1 mr-2" for="hsc\_s">hsc\_s</label>

<select class="custom-select my-1 mr-sm-2" id="hsc\_s"

name="hsc\_s" required>

<option value="0" selected>Arts</option>

<option value="1">Commerce</option>

<option value="2">Science</option>

</select>

</div>

<div class="form-group">

<label class="my-1 mr-2" for="degree\_t">degree\_t</label>

<select class="custom-select my-1 mr-sm-2" id="degree\_t" name="degree\_t" required>

<option value="0" selected>Comm&Mgmt</option>

<option value="1">Others</option>

<option value="2">Sci&Tech</option>

</select>

</div>

<div class="form-group">

<label class="my-1 mr-2" for="workex">workex</label>

<select class="custom-select my-1 mr-sm-2" id="workex" name="workex" required>

<option value="0" selected>No</option>

<option value="1">Yes</option>

</select>

</div>

<div class="form-group">

<label class="my-1 mr-2" for="specialisation">specialisation</label>

<select class="custom-select my-1 mr-sm-2" id="specialisation" name="specialisation" required>

<option value="0" selected>Mkt&Fin</option>

<option value="1">Mkt&HR</option>

</select>

</div>

</div>

</div>

<button type="submit" class="btn btn-block btn-primary" onclick="last()">Predict</button>

</form>

</div>

</div>

</div>

</section>

</main>

</body>

</html>