CAMPUS PLACEMENT PREDICTION WEBSITE USING ML

ROSPL MINI PROJECT REPORT

Submitted in partial fulfilment of the

requirements

For the degree of

Bachelor of Engineering

(Information Technology Engineering)

By

Suprit Giri 20IT5002

Prajwal Babar 19IT1038

Amruta Ahinave 19IT1015

Sanraj bhosale 19IT1067

Supervisor

Mrs.Sujata Oak

Department of Electronics Engineering
Ramrao Adik Institute of Technology,
Sector 7, Nerul , Navi Mumbai (Affiliated to
University of Mumbai December 2021)

Ramrao Adik Education Society's

Ramrao Adik Institute of Technology

(Affiliated to the University of Mumbai)
Dr. D. Y. Patil Vidyanagar,Sector 7, Nerul, Navi Mumbai
400706.

Certificate

This is to certify that, the project report-A titled

" CAMPUS PLACEMENT PREDICTION WEBSITE "

Suprit Giri 20IT5002

Prajwal Babar 19IT1038

Amruta Ahinave 19IT1015

Sanraj bhosale 19IT1067

and is submitted in the partial fulfillment of the requirement for the

degree of

Bachelor of Engineering

(Information Technology Engineering)
to the

University of Mumbai.

Examiner 1 Supervisor

Head of Department Principal

CERTIFICATE

This is to certify that the “**CAMPUS PLACEMENT PREDICTION WEBSITE”** being submitted by Suprit Giri 20IT5002, Prajwal Babar 19IT1038, Amruta Ahinave 19IT1015, Sanraj Bhosale 19IT1067 to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of ‘B.E.I.T’ in ‘ROSPL PROJECT’.

Project Guide	**External Examiner**	**Head of Department**
(Mrs.Sujata Oak)	( )	(Dr. Ashish Jadhav)

Abstract

Placement of students is one of the most important objective of an educational institution. Reputation and yearly admissions of an institution invariably depend on the placements it provides it students with. That is why all the institutions, arduously, strive to strengthen their placement department so as to improve their institution on a whole. Any assistance in this particular area will have a positive impact on an institution's ability to place its students. This will always be helpful to both the students, as well as the institution. In this study, the objective is to analyse previous year's student's data and use it to predict the placement chance of the current students.

This model is proposed with an algorithm to predict the same. Data pertaining to the study were collected form the same institution for which the placement prediction is done, and also suitable data pre-processing methods were applied. This proposed model is also compared with other traditional algorithms such as logistic regression and Random forest with respect to accuracy, precision and recall. From the results obtained it is found that the proposed algorithm performs significantly better in comparison with the other algorithms mentioned.

LIST OF FIGURES

LIST OF TABLES

1. Introduction

1.1 Fundamentals of OSS

Definition:

Open-source software (OSS) is software that is distributed with its source code, making it available for use, modification, and distribution with its original rights. Source code is the part of software that most computer users don't ever see; it's the code computer programmers manipulate to control how a program or application behaves. Programmers who have access to source code can change a program by adding to it, changing it, or fixing parts of it that aren't working properly. OSS typically includes a license that allows programmers to modify the software to best fit their needs and control how the software can be distributed.

How does OSS work?

Open-source code is usually stored in a public repository and shared publicly. Anyone can access the repository to use the code independently or contribute improvements to the design and functionality of the overall project.

OSS usually comes with a distribution license. This license includes terms that define how developers can use, study, modify, and most importantly, distribute the software. According to the Synopsys Black Duck Knowledge Base, five of the most popular licenses are:

- MIT License
- GNU General Public License (GPL) 2.0—this is more restrictive and requires that copies of modified code are made available for public use
- Apache License 2.0
- GNU General Public License (GPL) 3.0
- BSD License 2.0 (3-clause, New or Revised)—this is less restrictive

When source code is changed, OSS must include what was altered as well as the methods involved. Depending on the license terms, the software resulting from these modifications may or may not be required to be made available for free.

1.2 Gpl

Short for **GNU General Public License**, the **GPL** is a general license published by GNU project. Any software author may use the GPL to legally control the way their software may be used by others. It is a copyleft license, meaning that any code derived from GPL-licensed code must also be licensed under the GPL.

Historically, the GPL license family has been one of the most popular software licenses in the free and open-source software domain. Prominent free software programs licensed under the GPL include the Linux kernel and the GNU Compiler Collection (GCC). David A. Wheeler argues that the copyleft provided by the GPL was crucial to the success of Linux-based systems, giving the programmers who contributed to the kernel the assurance that their work would benefit the whole world and remain free, rather than being exploited by software companies that would not have to give anything back to the community.

In 2007, the third version of the license (GPLv3) was released to address some perceived problems with the second version (GPLv2) which were discovered during the latter's long-time usage. To keep the license up to date, the GPL license includes an optional "any later version" clause, allowing users to choose between the original terms or the terms in new versions as updated by the FSF. Developers can omit it when licensing their software; the Linux kernel, for instance, is licensed under GPLv2 without the "any later version" clause. The "or any later version" clause also known as a lifeboat clause allows combinations between different versions of GPL licensed software to maintain compatibility.

1.3 Different ways to contribute to OSP

- Discovering relevant projects
- Finding good first issues
- Opening an issue
- Validating an issue or pull request
- Reproducing a reported bug
- Testing a pull request
- Updating issues
- Create open source alternatives to commercial software
- Create your own open source project.

2. Contribution to Open source

2.1 Guidelines/steps involved in contribution

When we say contributing to open-source, it does not necesarilly mean that you need to know how to code. There are different ways in which you can contribute even if you are a non-coder – but having some coding skills will help you (and the projects) out a lot.

Some common contributions can be through:

- Adding a description to a project's documentation to elaborate on a certain point, mostly referred to as a README file (check this guide on how to write a Good README file).
- Giving guidance on a specific project and how to use it.
- Adding sample output to show how the code works.
- Writing in-depth tutorials for the project.
- Adding translation for a project - A good place to start with this might be with the translation program.
- Answering questions about a project (like on Stack Overflow or Reddit)
- You can fix typos and arrange the project's work folder correctly.

2.2 Why to contribute in OSP

- It will help sharpen your skills of coding and improvement into writing clean code.
- It helps the community and your peers get to know you. This recognition can bring you a lot of opportunities in your career.
- It helps you learn more about project management, and it could leave you inspired to start your own project.

2.3 Identifying the new/existing open-source projects to contribute- To Create open source alternatives to commercial software,Create your own open source project. The project is made with proper research of the different placement process and we have also focused on the problems faced by the different colleges and the needs that must be provided to them. The theme of project is basically it is an Online platform for campus placement prediction so that both the students and the Placement team could directly connect to each other. The site is compatible on most of the all modern software’s and devices.

3.Contribution to Open source in machine learning / python

3.1 Problem Definition

The placement prediction model considers only academic performances of the students so that the prediction of the student getting placed or not can be done. We cannot consider the placement of students just by their academic performances because some students may be good at aptitude, technical and communication skills due to their low score in their academic that may tend to be their drawback. For predicting the placement of a Student needs parameters like cgpa, logical and technical skills Academic performances may be important but the model is design to predict the placements based on the parameters of the students.

dataset

The Campus Recruitment Prediction uses Placement_Data_Full_Class.csv

Dataset has been used for this purpose, taken from the Kaggle.

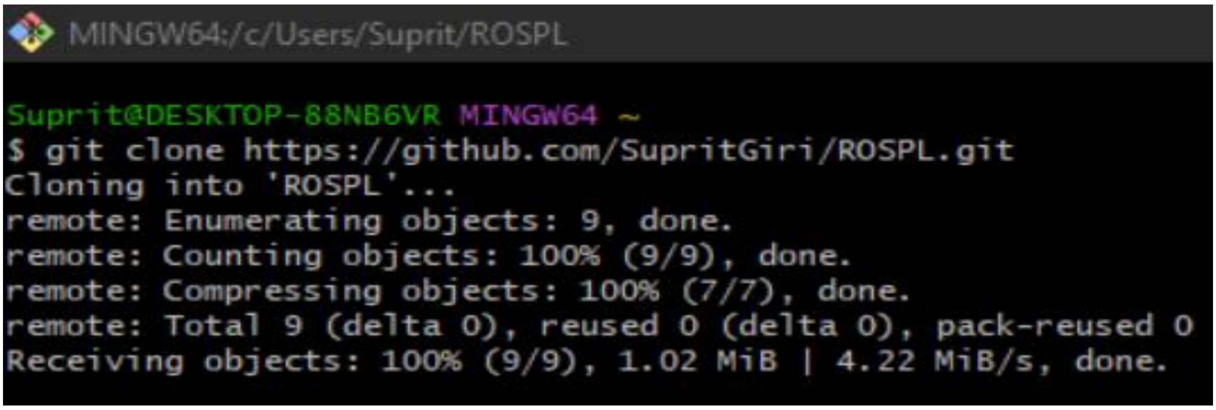
Life Cycle of implementing machine learning application.

- Gathering the Data
- Data Preparation
- Data Preprocessing
- Create Model
- Evaluate Model
- Deploy the model

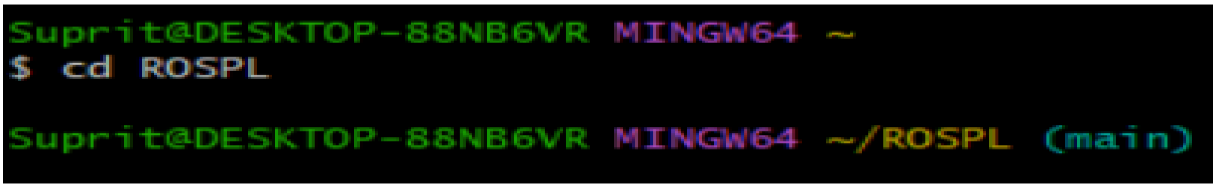
3.2 Objectives

The aim of project is to predict whether the student will be recruited in campus placements or not based on the available factors in the dataset. To analyse data and predict most important factors in placement process.

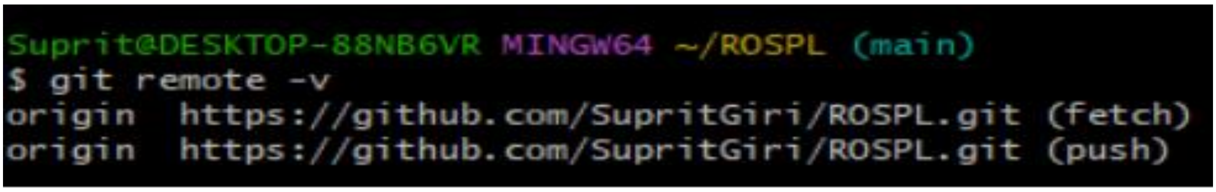
3.3 Submission of contribution snapshots



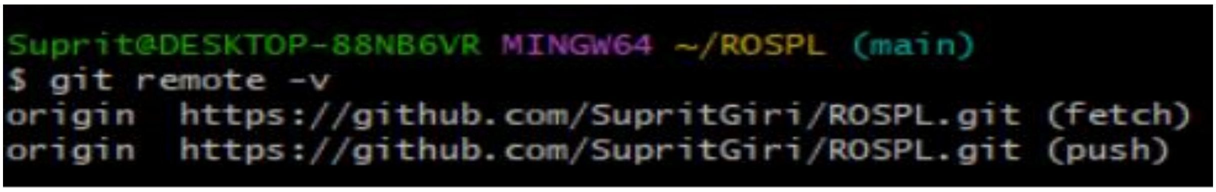
```
Suprit@DESKTOP-88NB6VR MINGW64 ~
$ git clone https://github.com/SupritGiri/ROSPL.git
Cloning into 'ROSPL'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (9/9), 1.02 MiB | 4.22 MiB/s, done.
```



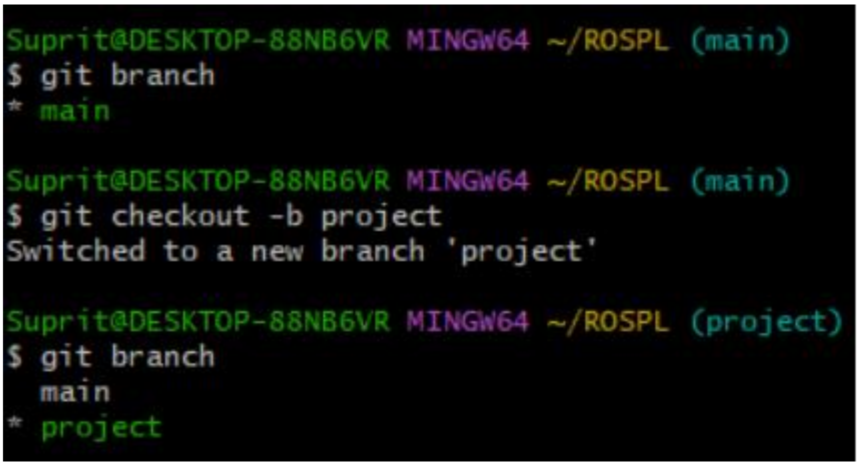
```
Suprit@DESKTOP-88NB6VR MINGW64 ~
$ cd ROSPL

Suprit@DESKTOP-88NB6VR MINGW64 ~/ROSPL (main)
```



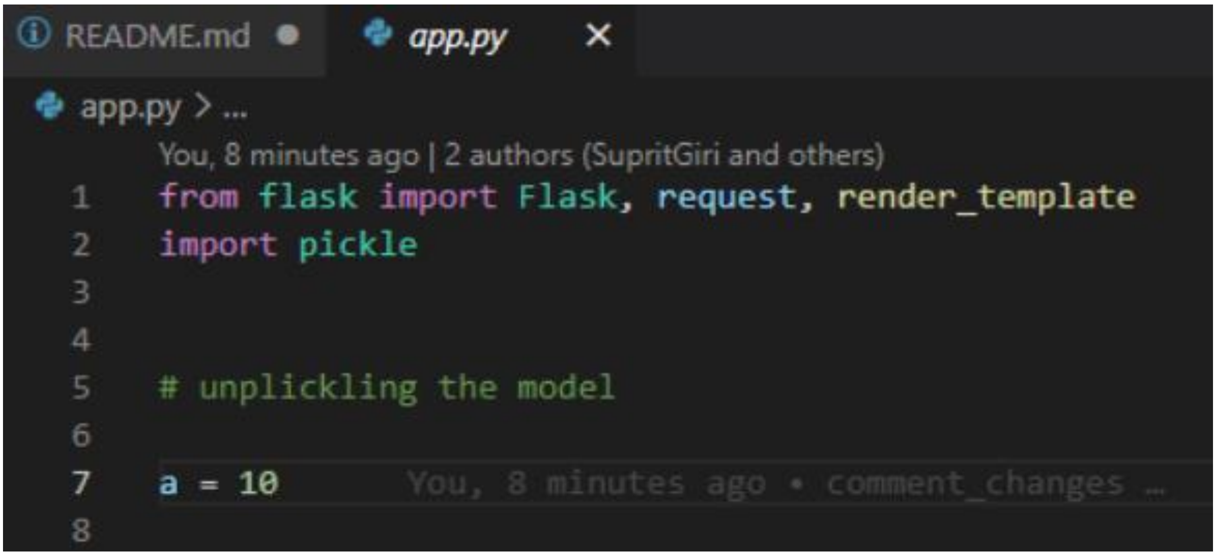
```
Suprit@DESKTOP-88NB6VR MINGW64 ~/ROSPL (main)
$ git remote -v
origin  https://github.com/SupritGiri/ROSPL.git (fetch)
origin  https://github.com/SupritGiri/ROSPL.git (push)
```



```
Suprit@DESKTOP-88NB6VR MINGW64 ~/ROSPL (main)
$ git remote -v
origin  https://github.com/SupritGiri/ROSPL.git (fetch)
origin  https://github.com/SupritGiri/ROSPL.git (push)
```

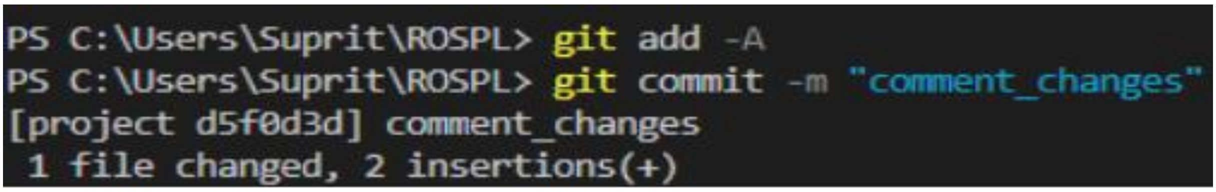


```
Suprit@DESKTOP-88NB6VR MINGW64 ~/ROSPL (main)
$ git branch
* main

Suprit@DESKTOP-88NB6VR MINGW64 ~/ROSPL (main)
$ git checkout -b project
Switched to a new branch 'project'

Suprit@DESKTOP-88NB6VR MINGW64 ~/ROSPL (project)
$ git branch
  main
* project
```

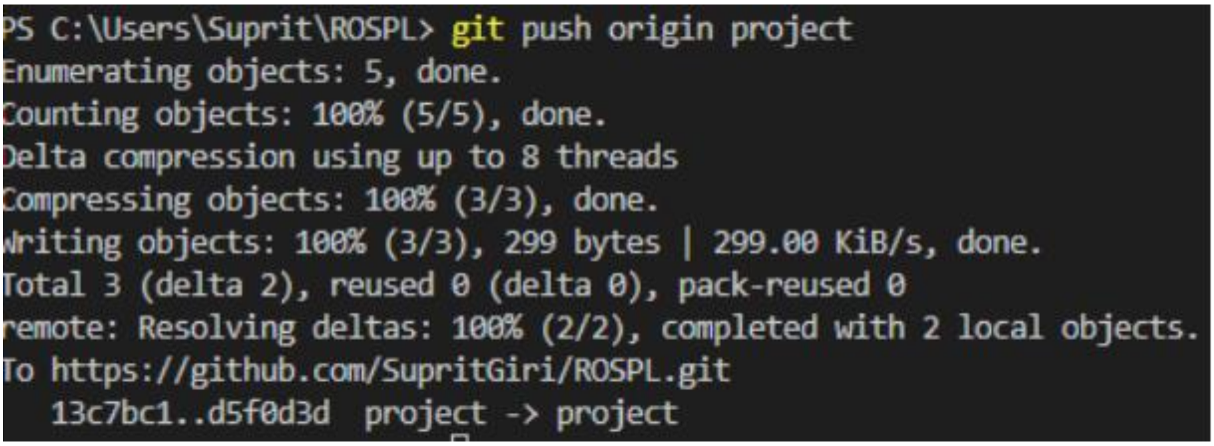


You, 8 minutes ago | 2 authors (SupritGiri and others)

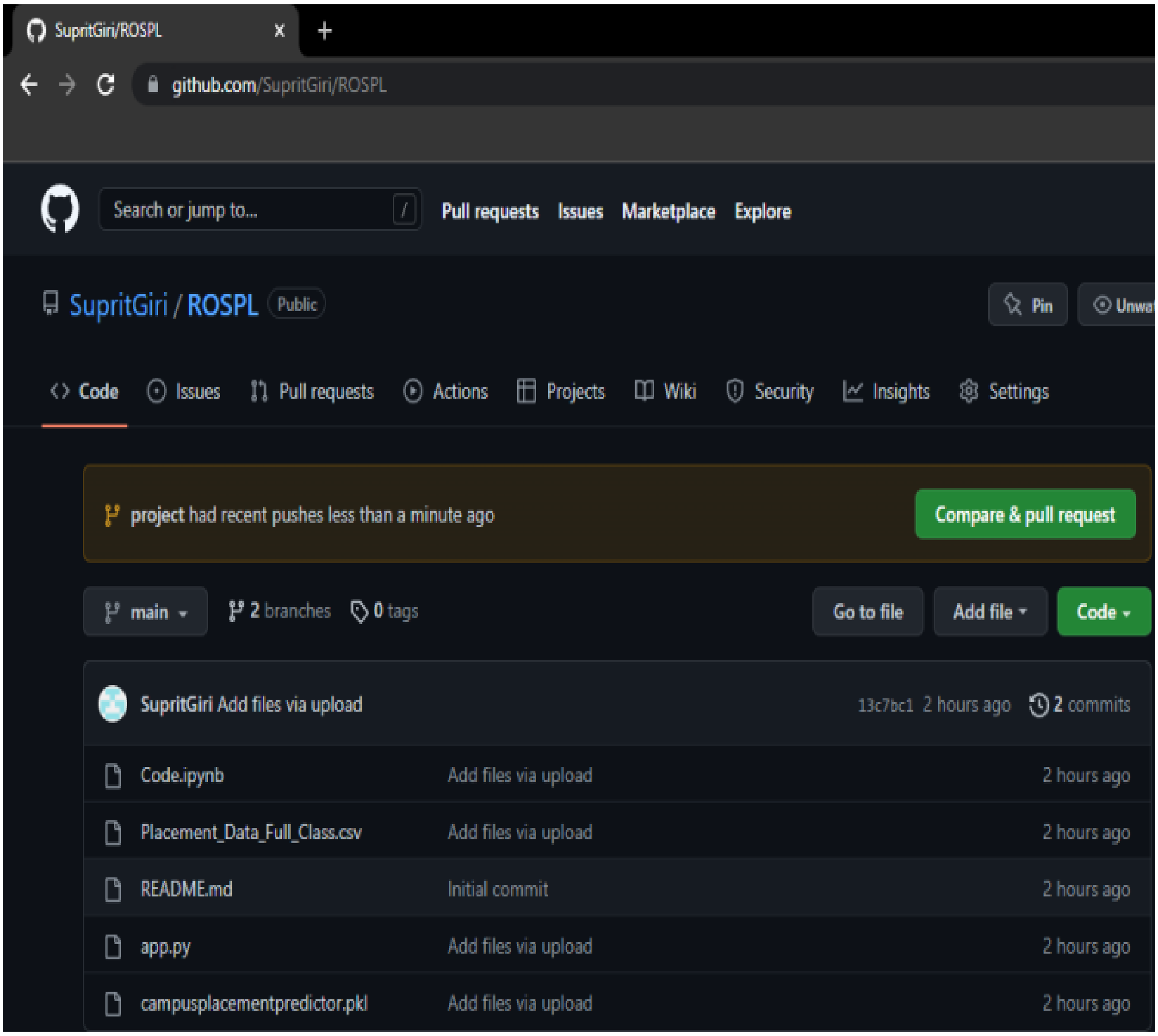
```python
from flask import Flask, request, render_template
import pickle


# unplickling the model

a = 10          You, 8 minutes ago • comment_changes ...

```



```
PS C:\Users\Suprit\ROSPL> git add -A
PS C:\Users\Suprit\ROSPL> git commit -m "comment_changes"
[project d5f0d3d] comment_changes
 1 file changed, 2 insertions(+)
```



```
PS C:\Users\Suprit\ROSPL> git push origin project
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/SupritGiri/ROSPL.git
   13c7bc1..d5f0d3d  project -> project
```



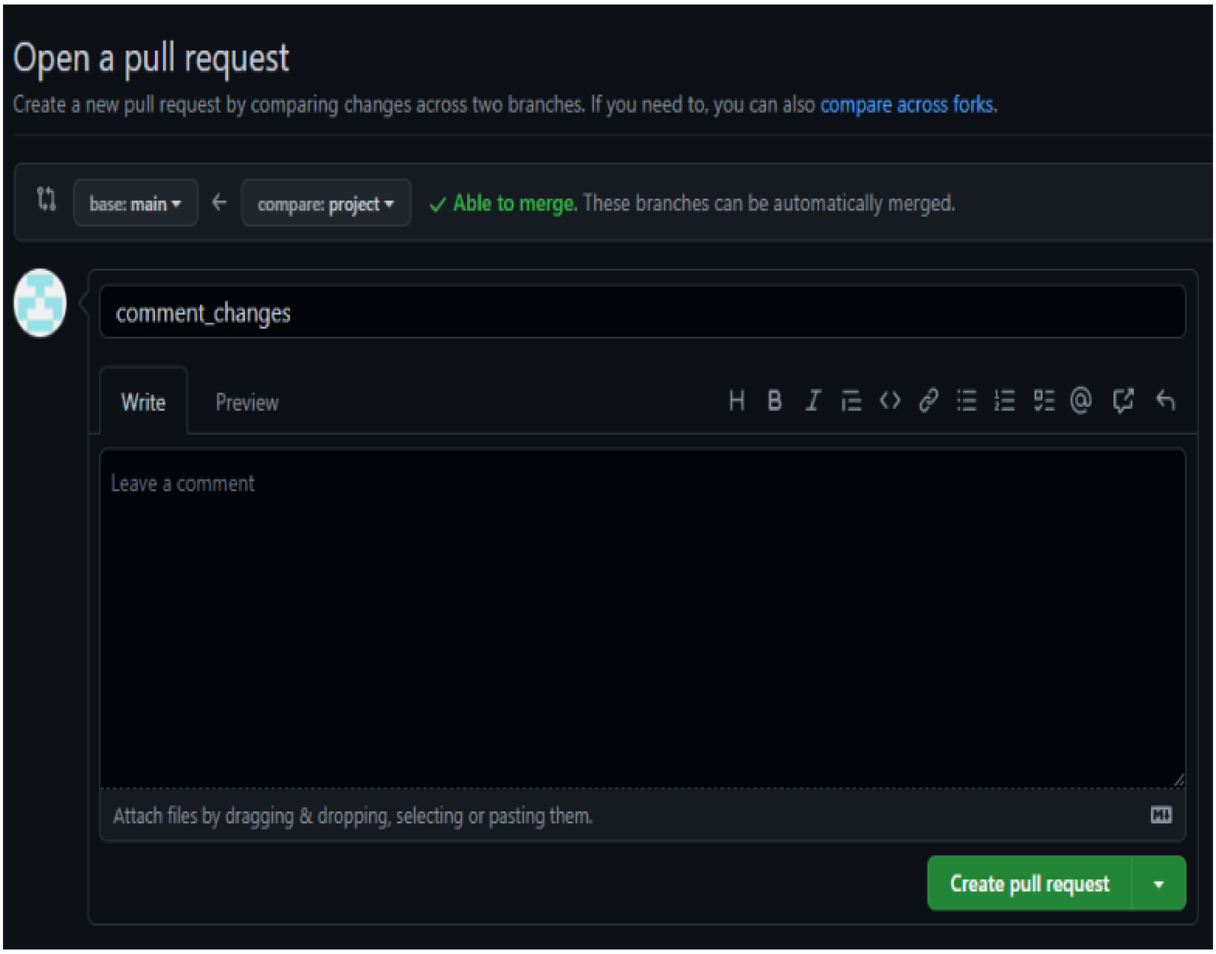
SupritGiri / ROSPL Public

project had recent pushes less than a minute ago

2 branches 0 tags

SupritGiri Add files via upload 13c7bc1 2 hours ago 2 commits

Code.ipynb	Add files via upload	2 hours ago
Placement_Data_Full_Class.csv	Add files via upload	2 hours ago
README.md	Initial commit	2 hours ago
app.py	Add files via upload	2 hours ago
campusplacementpredictor.pkl	Add files via upload	2 hours ago



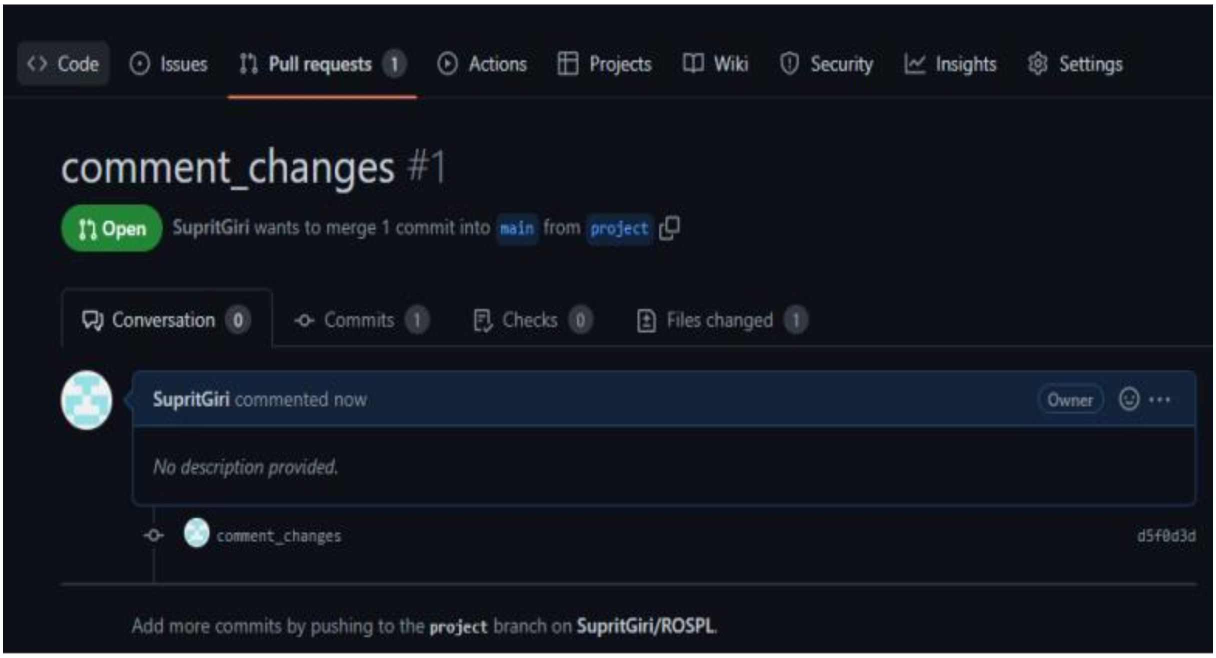
Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: main ← compare: project ✓ Able to merge. These branches can be automatically merged.

comment_changes

Write
Preview
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Create pull request



comment_changes #1

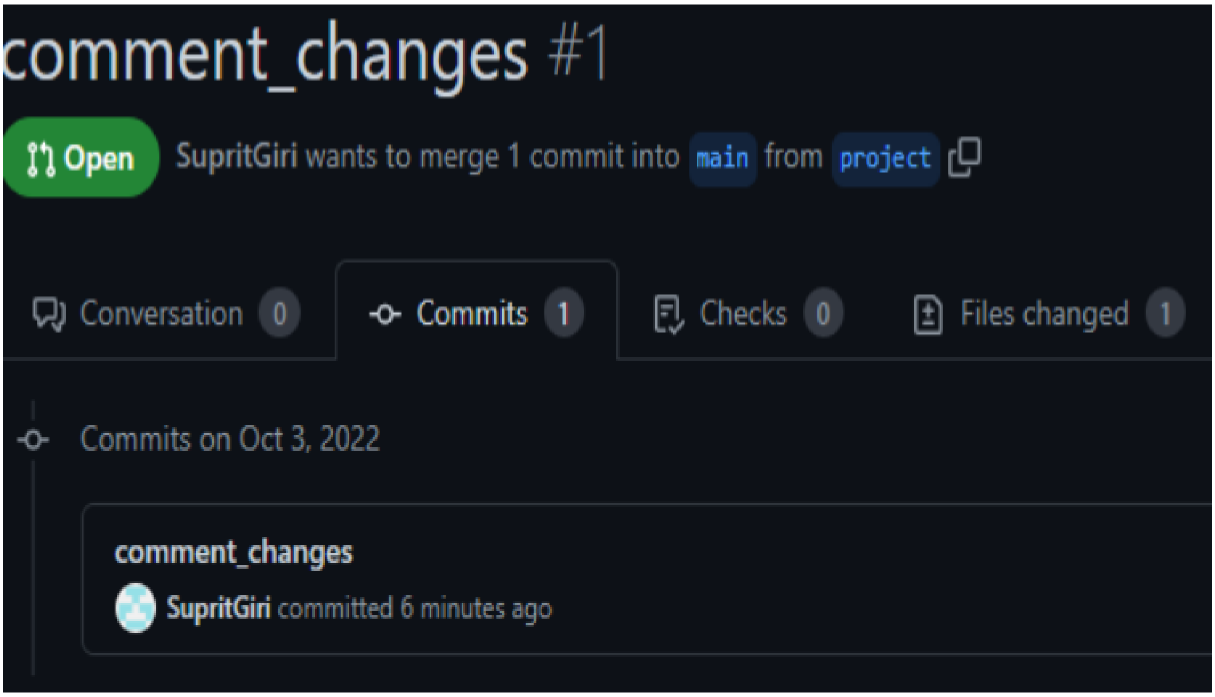
SupritGiri wants to merge 1 commit into main from project

SupritGiri commented now

No description provided.

comment_changes d5f0d3d

Add more commits by pushing to the **project** branch on **SupritGiri/ROSPL**.



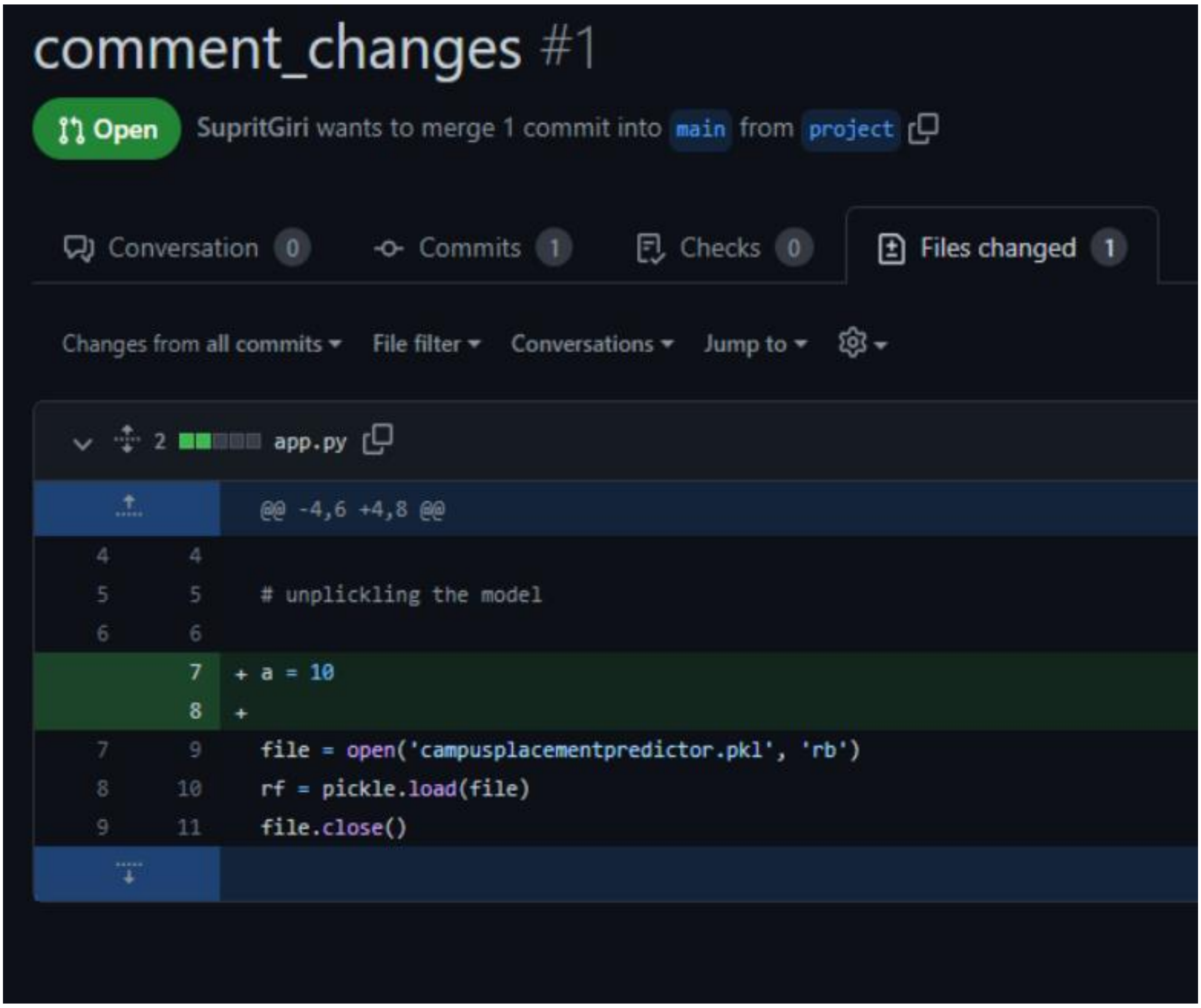
comment_changes #1

SupritGiri wants to merge 1 commit into main from project

Commits on Oct 3, 2022

comment_changes

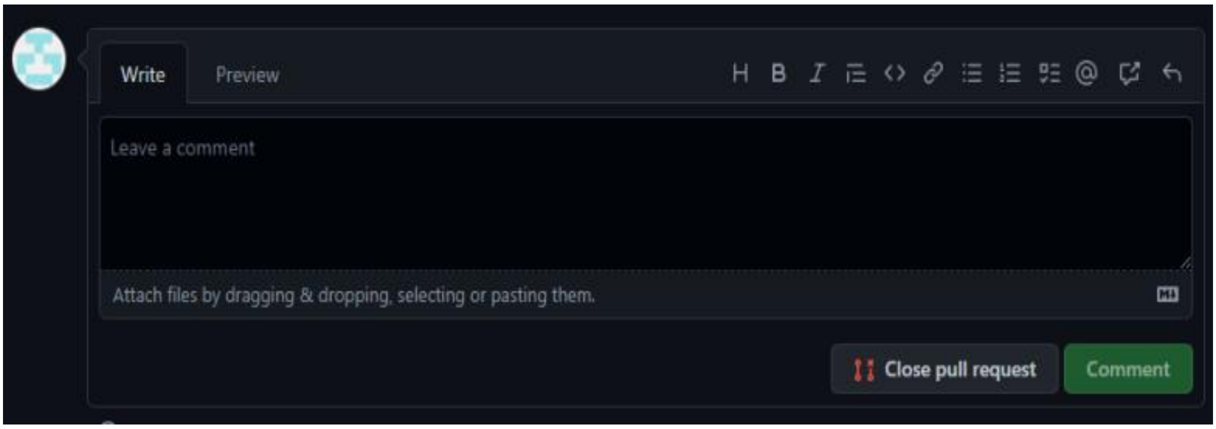
SupritGiri committed 6 minutes ago



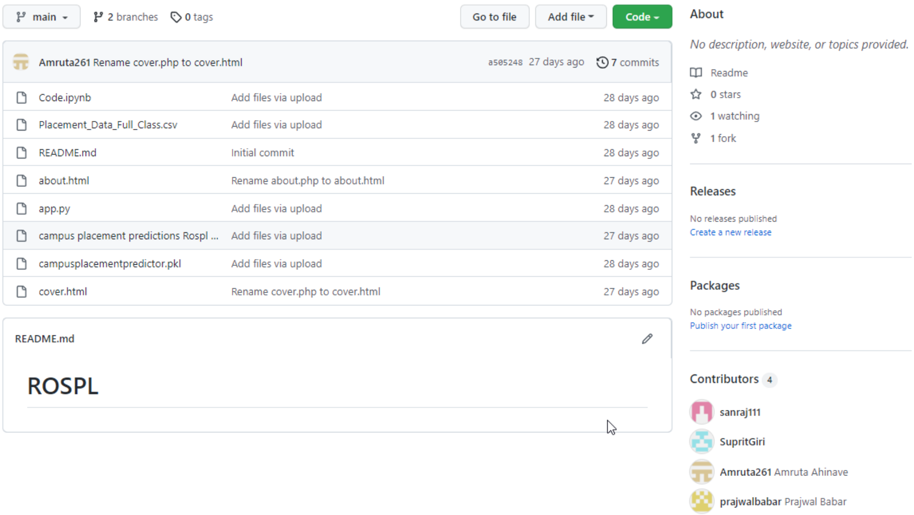
comment_changes #1

SupritGiri wants to merge 1 commit into main from project

```
@@ -4,6 +4,8 @@

  # unplickling the model

+ a = 10
+
  file = open('campusplacementpredictor.pkl', 'rb')
  rf = pickle.load(file)
  file.close()
```



Write
Preview
H
B
I
Leave a comment
Attach files by dragging & dropping, selecting or pasting them.
Close pull request
Comment



Amruta261 Rename cover.php to cover.html a505248 27 days ago 7 commits

Code.ipynb	Add files via upload	28 days ago
Placement_Data_Full_Class.csv	Add files via upload	28 days ago
README.md	Initial commit	28 days ago
about.html	Rename about.php to about.html	27 days ago
app.py	Add files via upload	28 days ago
campus placement predictions Rospl ...	Add files via upload	27 days ago
campusplacementpredictor.pkl	Add files via upload	28 days ago
cover.html	Rename cover.php to cover.html	27 days ago

README.md

ROSPL

About

No description, website, or topics provided.

Readme
0 stars
1 watching
1 fork

Releases

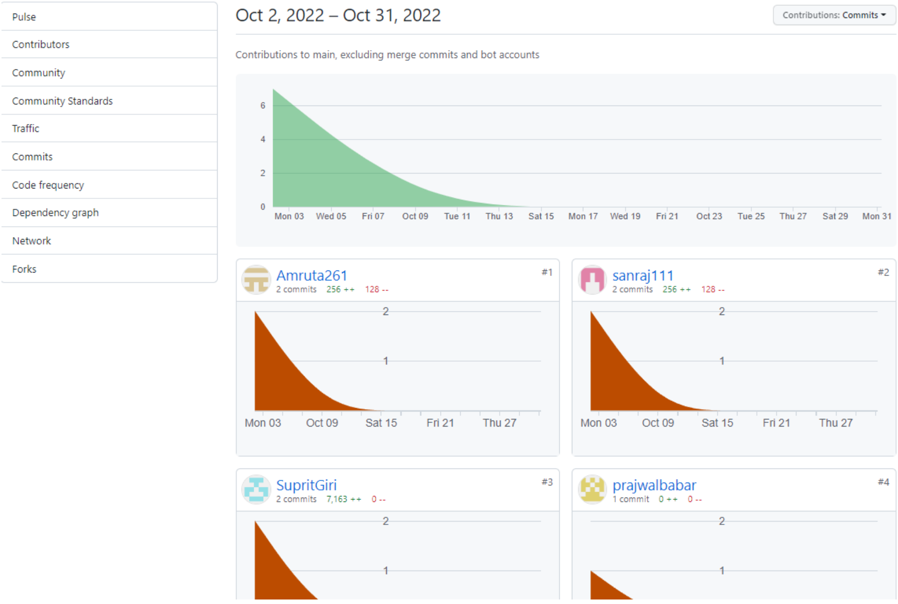
No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 4

sanraj111
SupritGiri
Amruta261 Amruta Ahinave
prajwalbabar Prajwal Babar



Oct 2, 2022 – Oct 31, 2022

Contributions to main, excluding merge commits and bot accounts

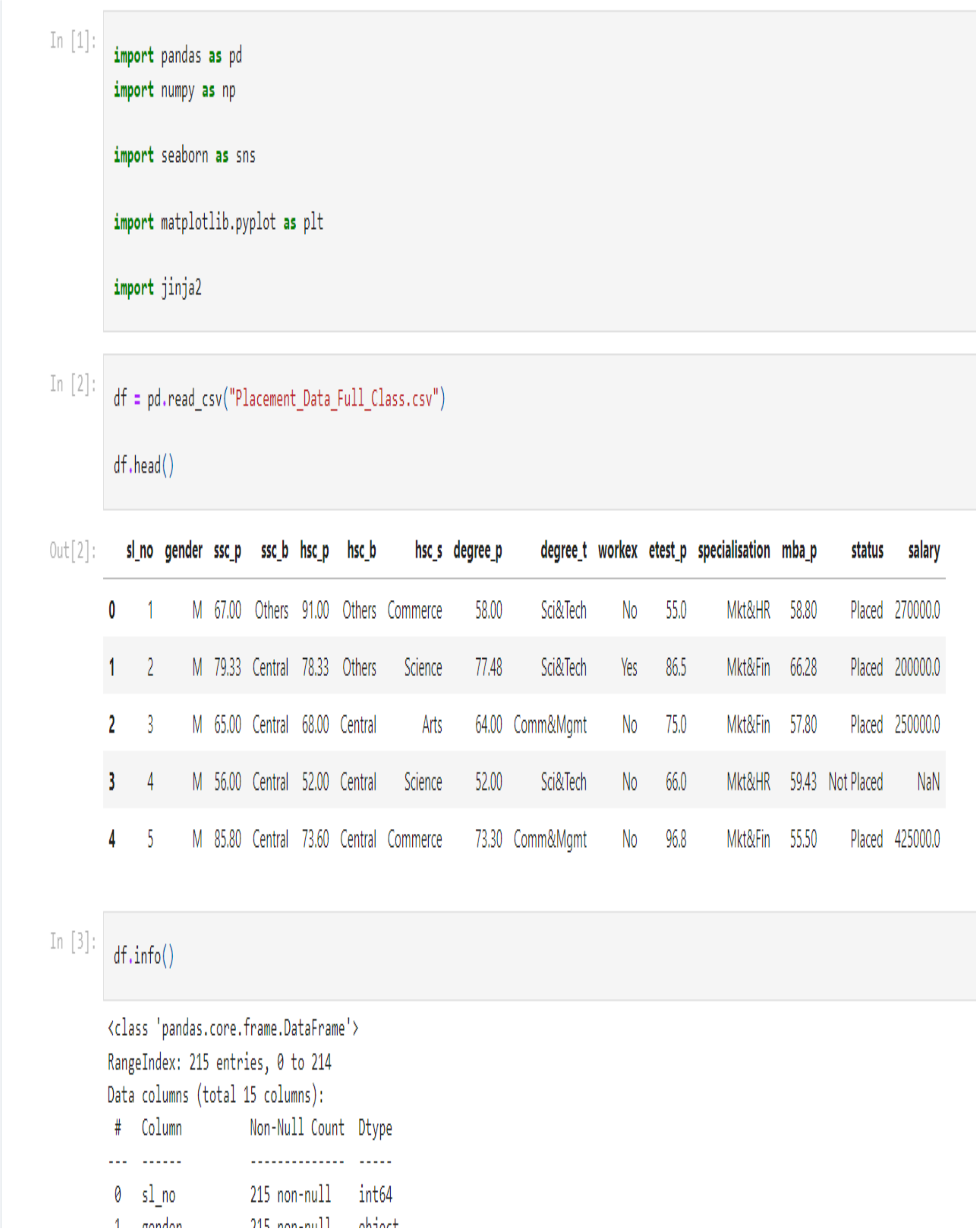
Amruta261 #1
2 commits 256 ++ 128 --

sanraj111 #2
2 commits 256 ++ 128 --

SupritGiri #3
2 commits 7,163 ++ 0 --

prajwalbabar #4
1 commit 0 ++ 0 --

3.4 Source code



In [1]:

```python
import pandas as pd
import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

import jinja2
```

In [2]:

```python
df = pd.read_csv("Placement_Data_Full_Class.csv")

df.head()
```

Out[2]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0
1	2	M	79.33	Central	78.33	Others	Science	77.48	Sci&Tech	Yes	86.5	Mkt&Fin	66.28	Placed	200000.0
2	3	M	65.00	Central	68.00	Central	Arts	64.00	Comm&Mgmt	No	75.0	Mkt&Fin	57.80	Placed	250000.0
3	4	M	56.00	Central	52.00	Central	Science	52.00	Sci&Tech	No	66.0	Mkt&HR	59.43	Not Placed	NaN
4	5	M	85.80	Central	73.60	Central	Commerce	73.30	Comm&Mgmt	No	96.8	Mkt&Fin	55.50	Placed	425000.0

In [3]:

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 215 entries, 0 to 214
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   sl_no           215 non-null    int64
 1   gender          215 non-null    object
```

Handling Null Values

In [6]:

```
def plotdistplot(col):

    plt.figure(figsize = (15, 7))
    sns.distplot(df["salary"], kde = True, hist = False, label = "Actual Salary", color = "red")
    sns.distplot(df[col], kde = True, hist = False, label = col, color = "black")
```

In [7]:

```
df["salary"].mode()[0]
```

Out[7]:

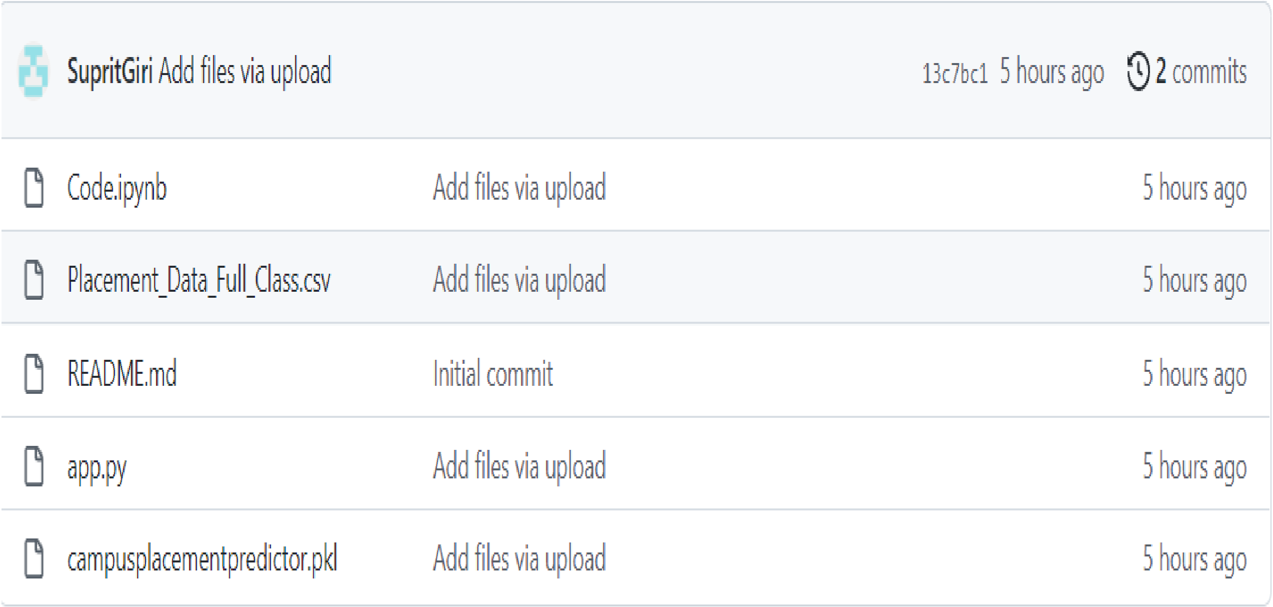
```
300000.0
```

In [8]:

```
df["salary_mean"] = df["salary"].fillna(df["salary"].mean())

df["salary_median"] = df["salary"].fillna(df["salary"].median())

df["salary_mode"] = df["salary"].fillna(df["salary"].mode()[0])

df.head(3)
```

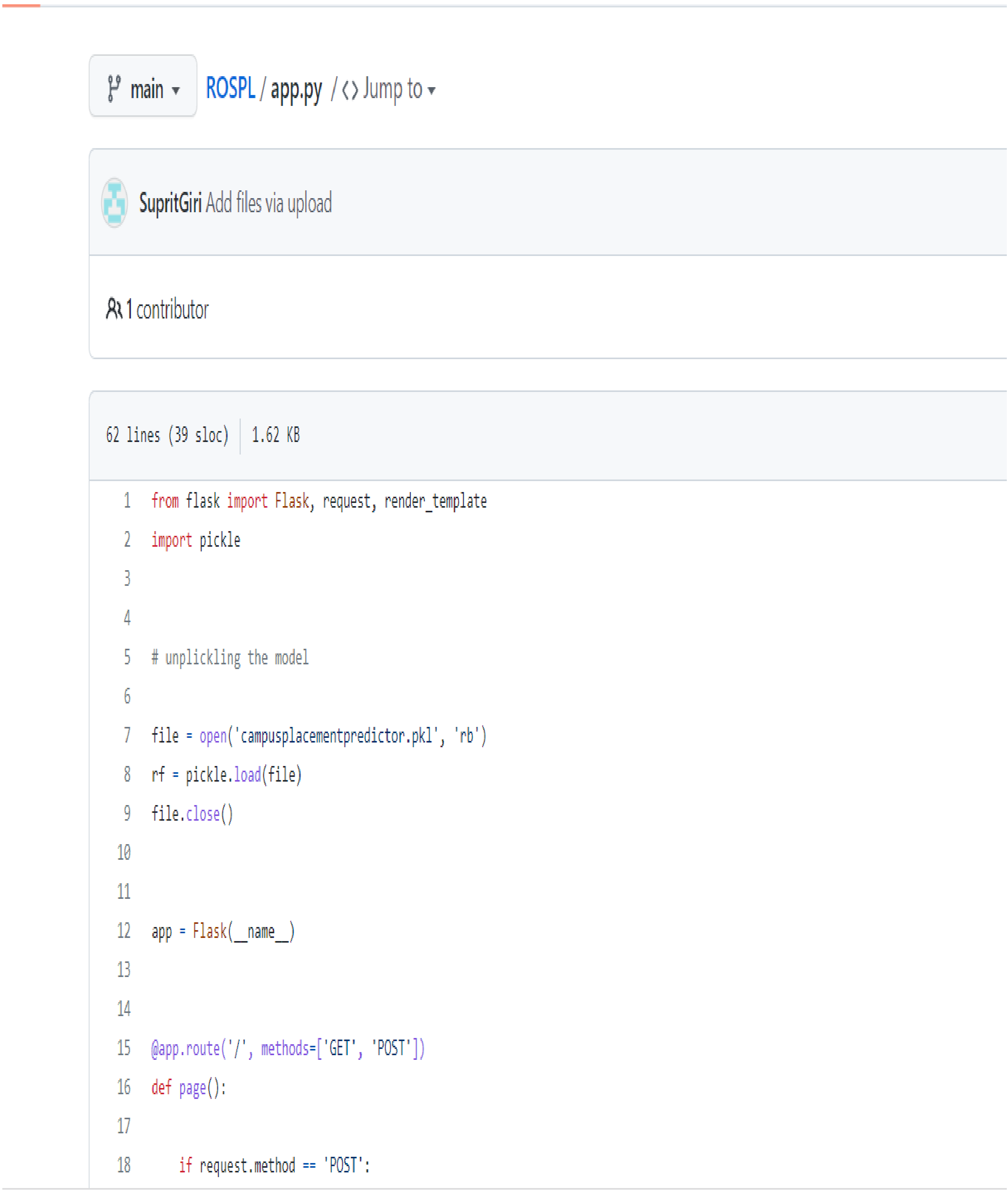
Out[8]:

	sl_no	gender	ssc_p	ssc_b	hsc_p	hsc_b	hsc_s	degree_p	degree_t	workex	etest_p	specialisation	mba_p	status	salary	salary_mean	salary_median	salary_
0	1	M	67.00	Others	91.00	Others	Commerce	58.00	Sci&Tech	No	55.0	Mkt&HR	58.80	Placed	270000.0	270000.0	270000.0	270



SupritGiri Add files via upload 13c7bc1 5 hours ago **2** commits

Code.ipynb	Add files via upload	5 hours ago
Placement_Data_Full_Class.csv	Add files via upload	5 hours ago
README.md	Initial commit	5 hours ago
app.py	Add files via upload	5 hours ago
campusplacementpredictor.pkl	Add files via upload	5 hours ago



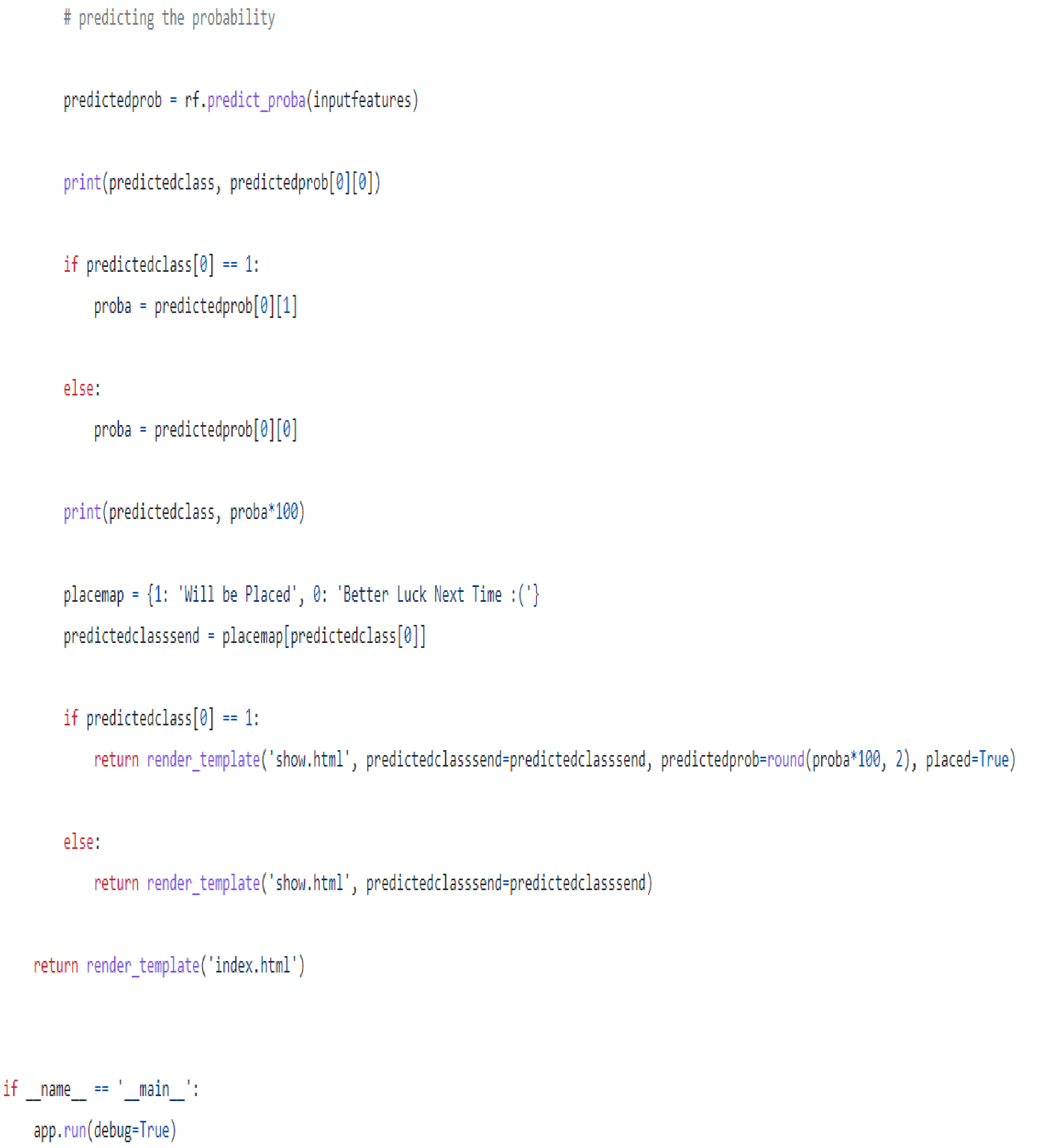
ROSPL / app.py / <> Jump to ▾

SupritGiri Add files via upload

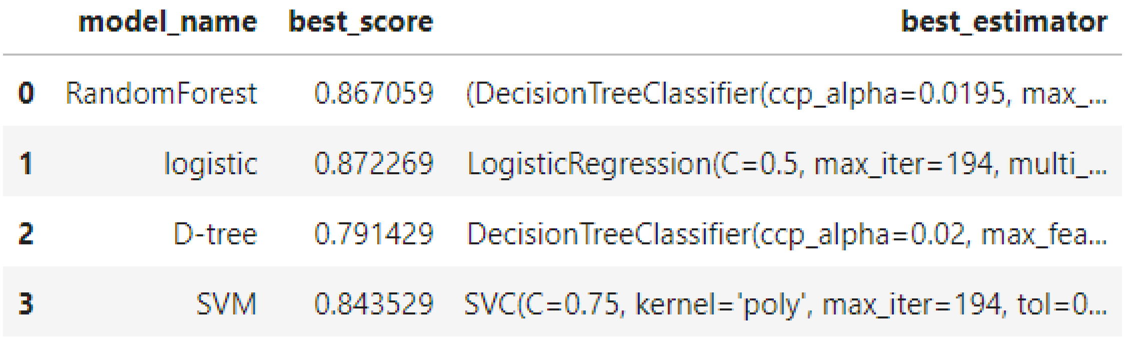
1 contributor

62 lines (39 sloc) | 1.62 KB

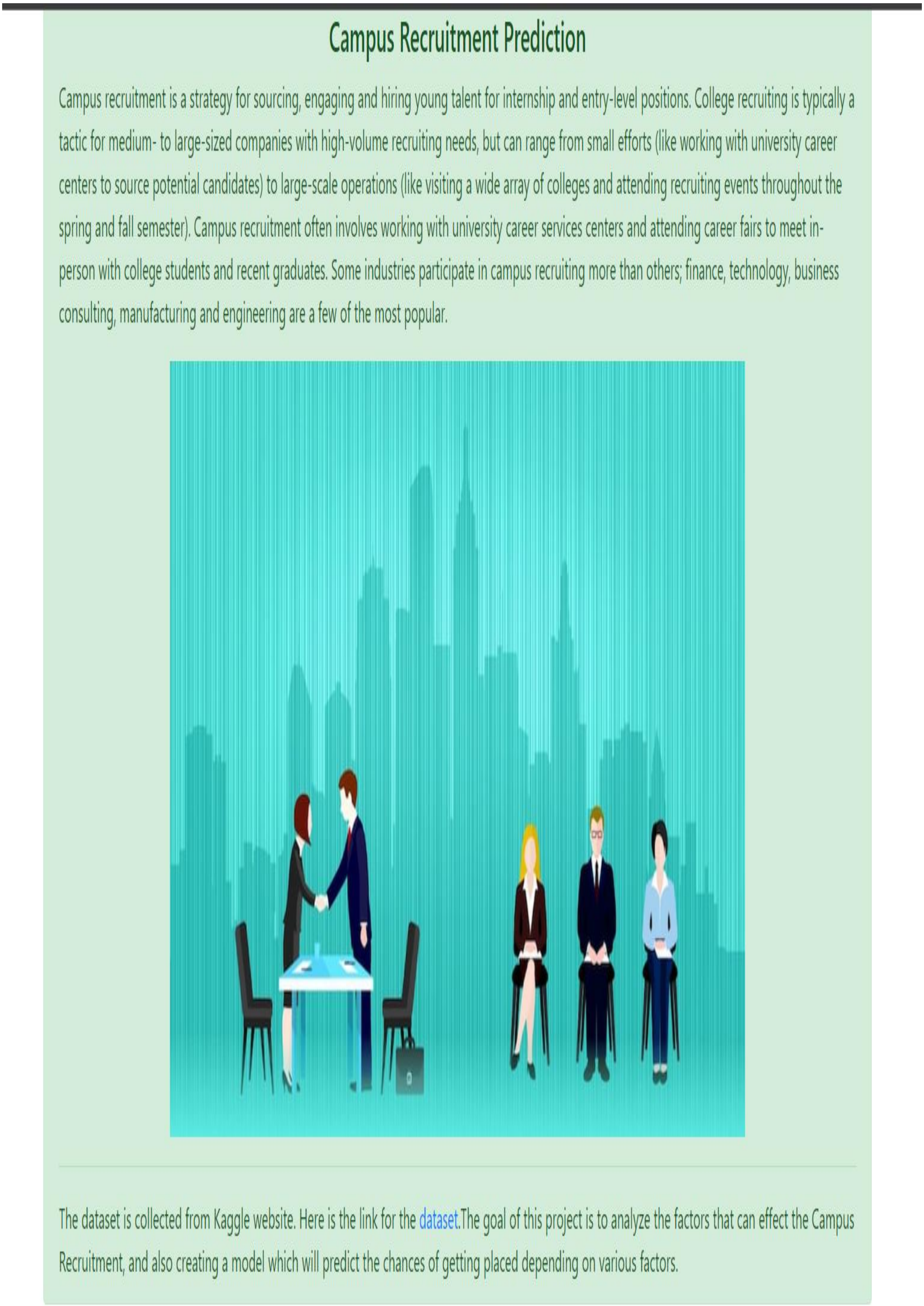
```python
from flask import Flask, request, render_template
import pickle


# unplickling the model

file = open('campusplacementpredictor.pkl', 'rb')
rf = pickle.load(file)
file.close()


app = Flask(__name__)


@app.route('/', methods=['GET', 'POST'])
def page():

    if request.method == 'POST':
```



```
        # predicting the probability

        predictedprob = rf.predict_proba(inputfeatures)

        print(predictedclass, predictedprob[0][0])

        if predictedclass[0] == 1:
            proba = predictedprob[0][1]

        else:
            proba = predictedprob[0][0]

        print(predictedclass, proba*100)

        placemap = {1: 'Will be Placed', 0: 'Better Luck Next Time :('}
        predictedclasssend = placemap[predictedclass[0]]

        if predictedclass[0] == 1:
            return render_template('show.html', predictedclasssend=predictedclasssend, predictedprob=round(proba*100, 2), placed=True)

        else:
            return render_template('show.html', predictedclasssend=predictedclasssend)

    return render_template('index.html')


if __name__ == '__main__':
    app.run(debug=True)
```



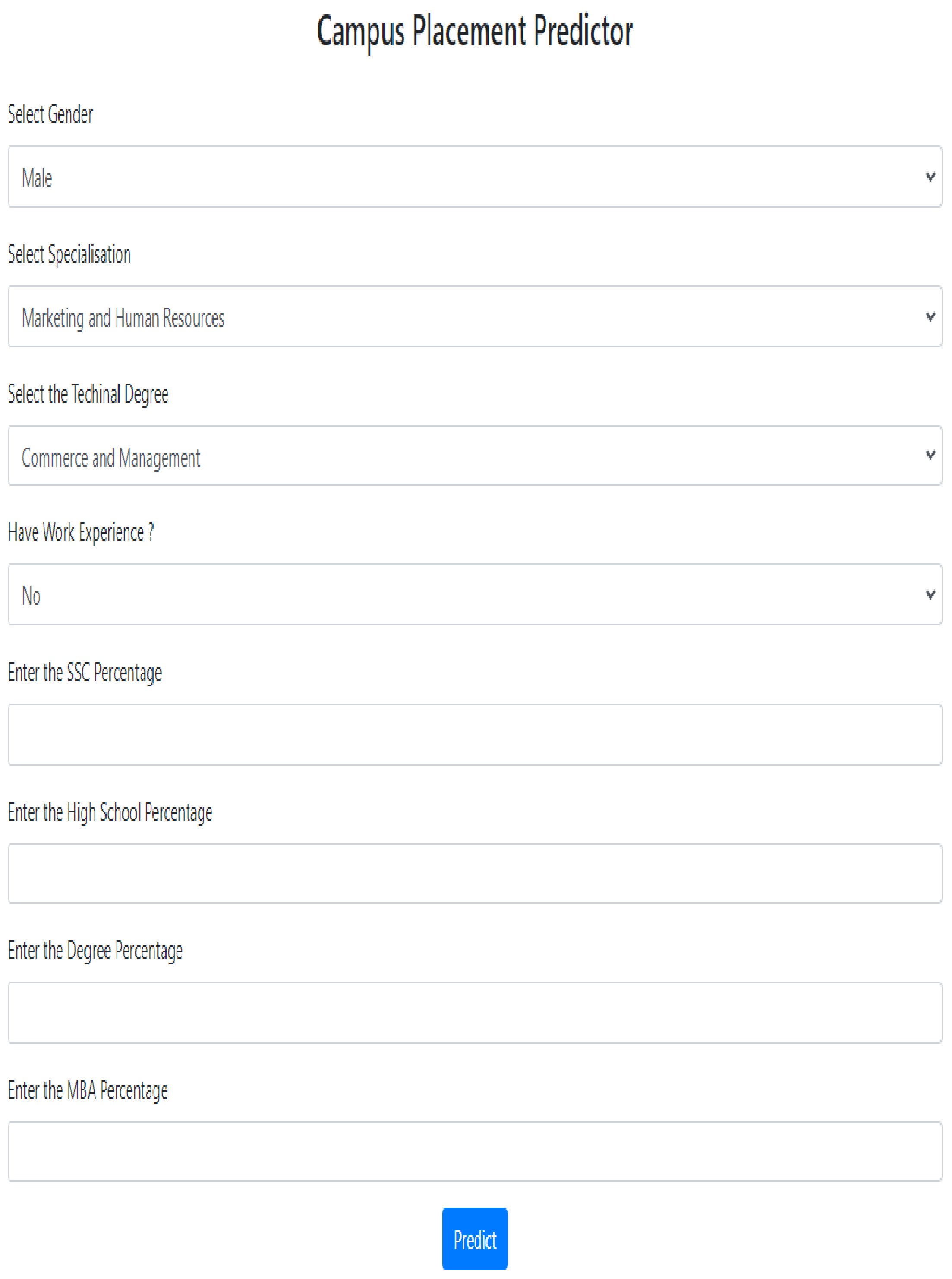
	model_name	best_score	best_estimator
0	RandomForest	0.867059	(DecisionTreeClassifier(ccp_alpha=0.0195, max_...
1	logistic	0.872269	LogisticRegression(C=0.5, max_iter=194, multi_...
2	D-tree	0.791429	DecisionTreeClassifier(ccp_alpha=0.02, max_fea...
3	SVM	0.843529	SVC(C=0.75, kernel='poly', max_iter=194, tol=0...



Campus Recruitment Prediction

Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry-level positions. College recruiting is typically a tactic for medium- to large-sized companies with high-volume recruiting needs, but can range from small efforts (like working with university career centers to source potential candidates) to large-scale operations (like visiting a wide array of colleges and attending recruiting events throughout the spring and fall semester). Campus recruitment often involves working with university career services centers and attending career fairs to meet in-person with college students and recent graduates. Some industries participate in campus recruiting more than others; finance, technology, business consulting, manufacturing and engineering are a few of the most popular.

The dataset is collected from Kaggle website. Here is the link for the dataset.The goal of this project is to analyze the factors that can effect the Campus Recruitment, and also creating a model which will predict the chances of getting placed depending on various factors.



Campus Placement Predictor

Select Gender

Male

Select Specialisation

Marketing and Human Resources

Select the Techinal Degree

Commerce and Management

Have Work Experience ?

No

Enter the SSC Percentage

Enter the High School Percentage

Enter the Degree Percentage

Enter the MBA Percentage

4.Conclusion and Future Scope

4.1 Conclusion

The placement department plays an important role in student placements, which raises the institute's worth. The current system follows the standard procedure of a firm visiting institutions and doing campus selection. Following the interview procedure, we shall learn the names of the students who have been chosen. However, we require 100% placements in order to improve the college's reputation. As a result, we require a system that can anticipate student placements in advance. As a result, “Student Placement Prediction” application has been created. The system assists colleges in predicting student placement status and boosting placement opportunities.

4.2 Future Scope

We can employ a larger number of algorithms and apply them to training datasets, allowing us to find the best algorithm. For placement prediction, a greater number of parameters and training datasets can be used.

5. References

- https://github.com/SupritGiri/ROSPL
- https://opensource.guide/how-to-contribute/
- https://www.freecodecamp.org/news/how-to-contribute-to-open-source-projects-beginners-guide/
- https://www.jetir.org/papers/JETIR2107359.pdf

6. Acknowledgement

We take this opportunity to express our profound gratitude and deep regards to my project guide Mrs. Sujata Oak for her exemplary guidance, monitoring and constant encouragement throughout the completion of this report. We are truly grateful to her efforts to improve our technical writing skills. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark. We are also thankful to Dr. Ashish Jadhav, Head of Department of Information Technology, for his valuable support and guidance.

Suprit Giri (20IT5002)

Prajwal Babar (19IT1038)

Amruta Ahinave (19IT1015)

Sanraj bhosale (19IT1067)