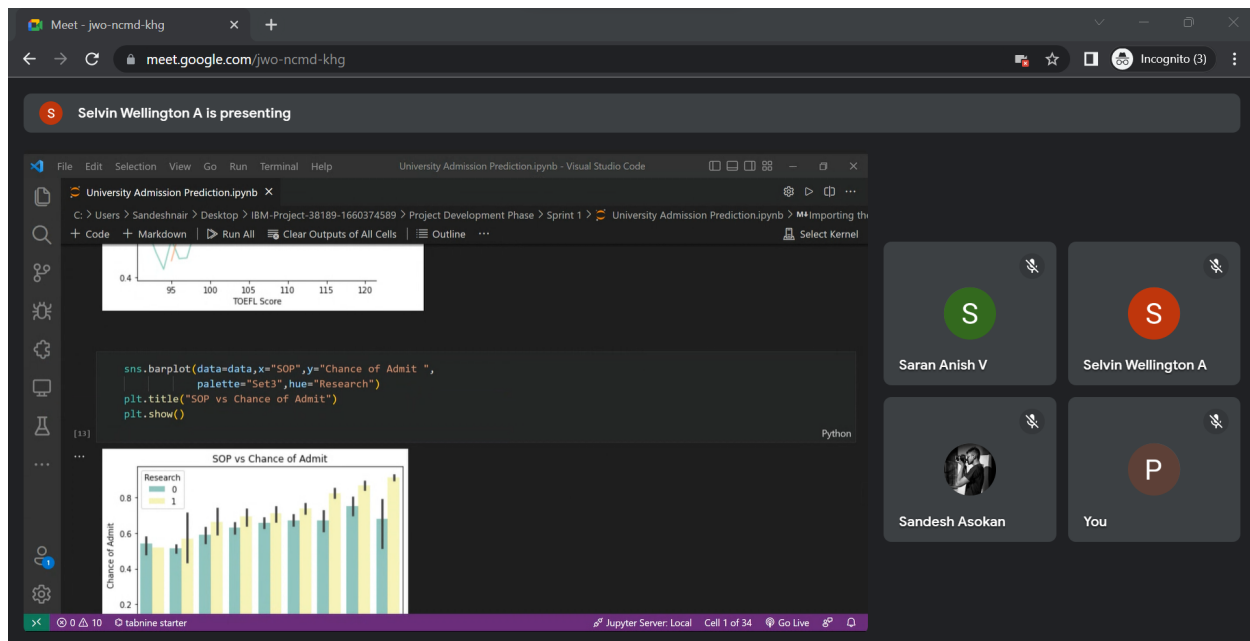


TEAM ID	PNT2022TMID27598
PROJECT NAME	University Admit Eligibility Predictor
DATE	19-11-2022

MINUTES OF THE MEETING: SPRINT-1

1. To complete Ideation
2. To identify Multivariate Analysis and predict accuracy
3. To split data into train and test



TEAM ID	PNT2022TMID27598
PROJECT NAME	University Admit Eligibility Predictor
DATE	19-11-2022

MINUTES OF THE MEETING: SPRINT 2

1. To design the UI for the Website
2. To develop the html Pages
3. To design the app.

The screenshot shows a Google Meet window with a dark theme. At the top, a browser tab is open to 'meet.google.com/jwo-ncmd-khg'. Below the browser, a status bar indicates 'Praveen Rijal is presenting'. The main area is split into two parts. On the left, a Visual Studio Code editor displays the code for 'noChance.html'. The code includes a DOCTYPE declaration, HTML and head tags, meta tags for charset, viewport, and title, and a link to a Bootstrap CSS file. The body contains a 'predict' section with a card layout. On the right, a grid of participant avatars is visible. The participants are Sandesh Asokan, Praveen Rijal, Selvin Wellington A, and You. The bottom of the screen shows a timeline at 5:56 PM and a set of meeting controls including mute, video, chat, and end call buttons.

TEAM ID	PNT2022TMID27598
PROJECT NAME	University Admit Eligibility Predictor
DATE	19-11-2022

MINUTES OF THE MEETING: SPRINT 3

1. To Deploy in IBM Cloud
2. To perform Prediction Test

The screenshot shows a Google Meet window with a presentation titled "University Admission Prediction.ipynb" by Praveen Rijal. The presentation is displayed in a dark-themed Jupyter Notebook interface. The code cell shows the command `!pip install ibm_watson_machine_learning` and its output, which lists various dependencies and their versions. The output includes:

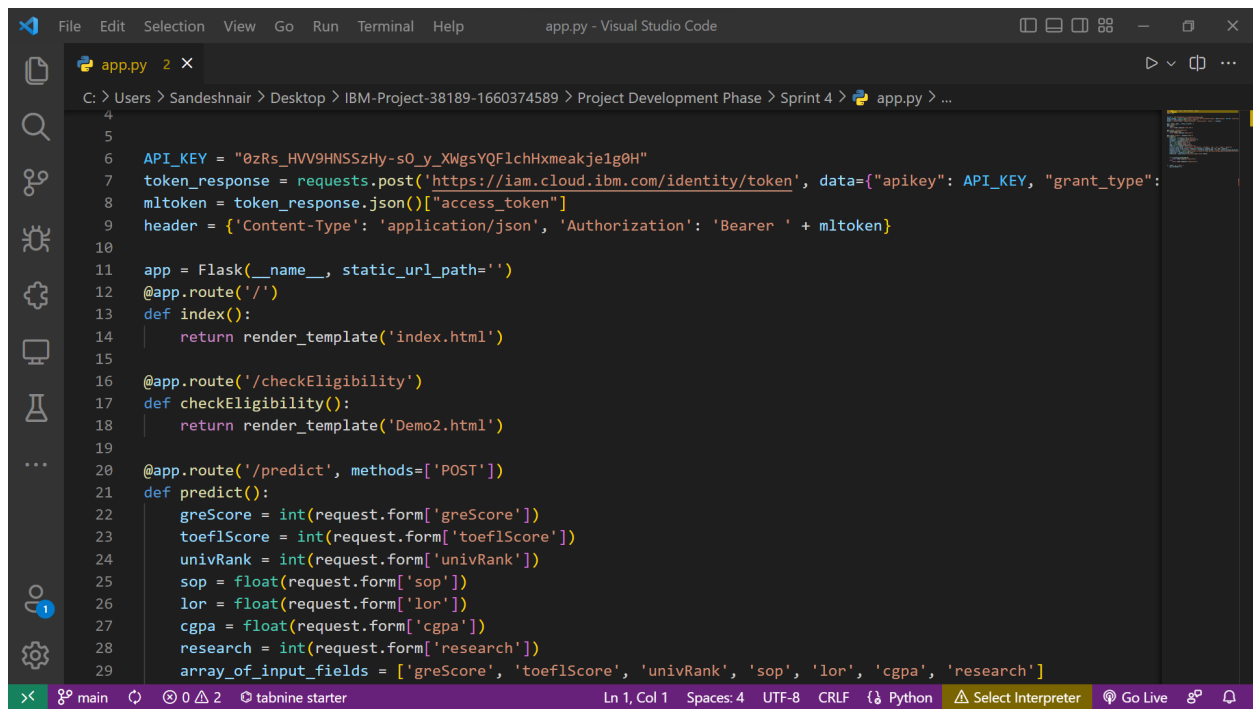
```
Requirement already satisfied: ibm_watson_machine_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.257)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (2.26.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.3.4)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.3.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (1.26.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (0.8.9)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm_watson_machine_learning) (21.3)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
```

On the right side of the Jupyter Notebook, there is a sidebar showing the participants: Sandesh Asokan, Praveen Rijal, Selvin Wellington A, and You. The bottom of the screen shows the Google Meet interface with a timer at 5:59 PM and a status bar indicating the user is in a meeting.

TEAM ID	PNT2022TMID27598
PROJECT NAME	University Admit Eligibility Predictor
DATE	19-11-2022

MINUTES OF THE MEETING: SPRINT 4

1. To integrate using Flask
2. To use the API Key and endpoint



```

4
5
6 API_KEY = "0zRs_HVV9HNSSzHy-sO_y_XWgsYQFlchHxmeakje1g0H"
7 token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type":
8 mltoken = token_response.json()["access_token"]
9 header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
10
11 app = Flask(__name__, static_url_path='')
12 @app.route('/')
13 def index():
14     return render_template('index.html')
15
16 @app.route('/checkEligibility')
17 def checkEligibility():
18     return render_template('Demo2.html')
19
20 @app.route('/predict', methods=['POST'])
21 def predict():
22     greScore = int(request.form['greScore'])
23     toeflScore = int(request.form['toeflScore'])
24     univRank = int(request.form['univRank'])
25     sop = float(request.form['sop'])
26     lor = float(request.form['lor'])
27     cgpa = float(request.form['cgpa'])
28     research = int(request.form['research'])
29     array_of_input_fields = ['greScore', 'toeflScore', 'univRank', 'sop', 'lor', 'cgpa', 'research']

```