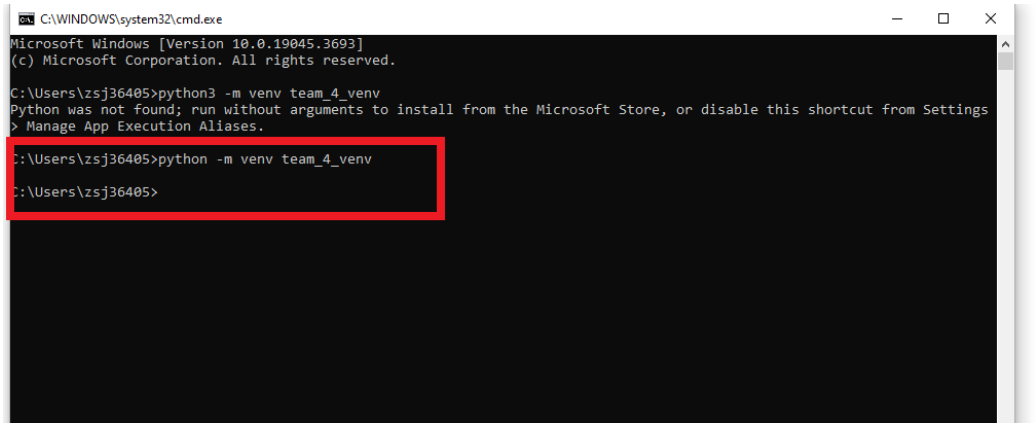
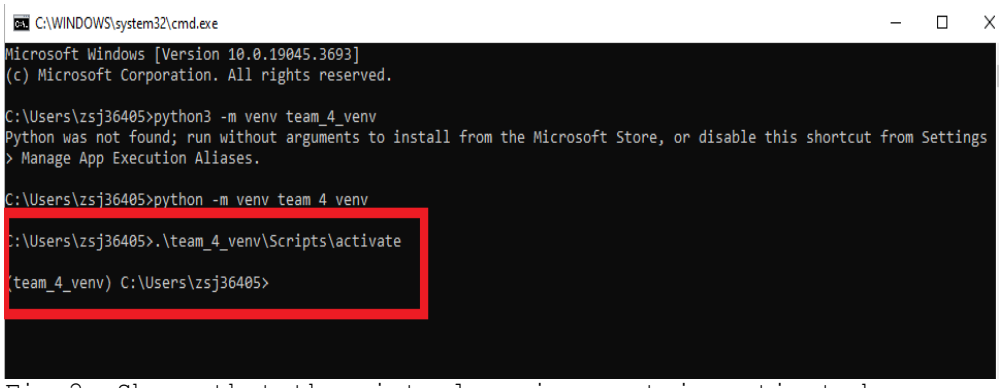
INSTALLATION GUIDE – TEAM 4

This document serves purpose of informing users the method to install and run the project – Taxi Price Prediction Using Multiple Features (Team 4 – Zorawar Jaiswal & Kashif Hashmi).

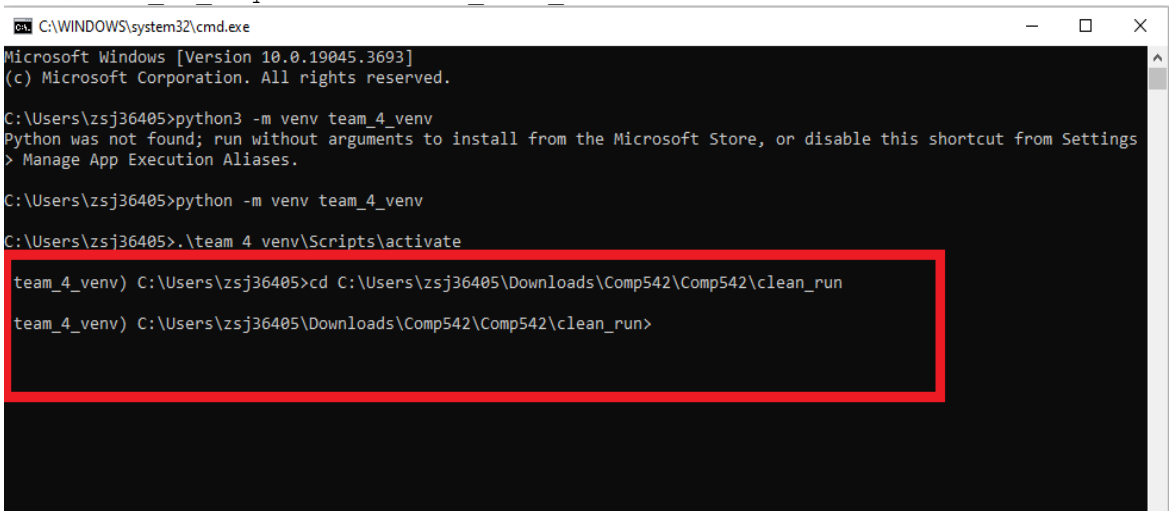
- Download and unzip the folder in some directory and note this path. This path will be the base directory path.
- **Create a Virtual Environment**
  - Go to Start
  - Open Command Prompt
  - Type the command:

    ```
    python -m venv team_4_venv
    ```
  - Press Enter.
  - 

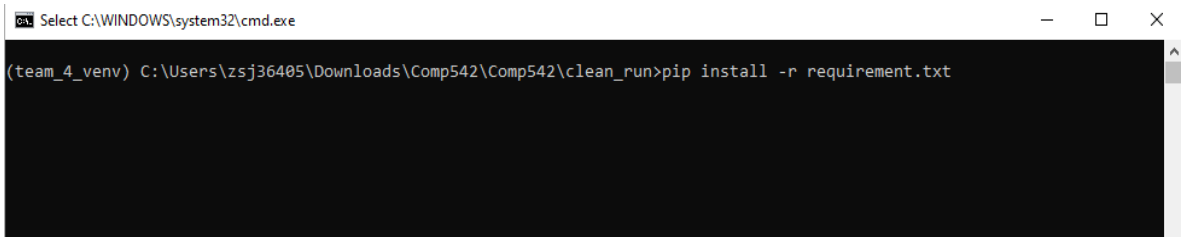
  -
  - Fig 1. Here shows creation of a virtual environment
  - **Note – The recommended version of python is 3.11.4**
- **Now we activate the virtual environment and install the libraries.**
  - In the same cmd, type:
  - `.\team_4_venv\Scripts\activate`
  - Press Enter.
  - 

  -
  - Fig 2. Shows that the virtual environment is activated.

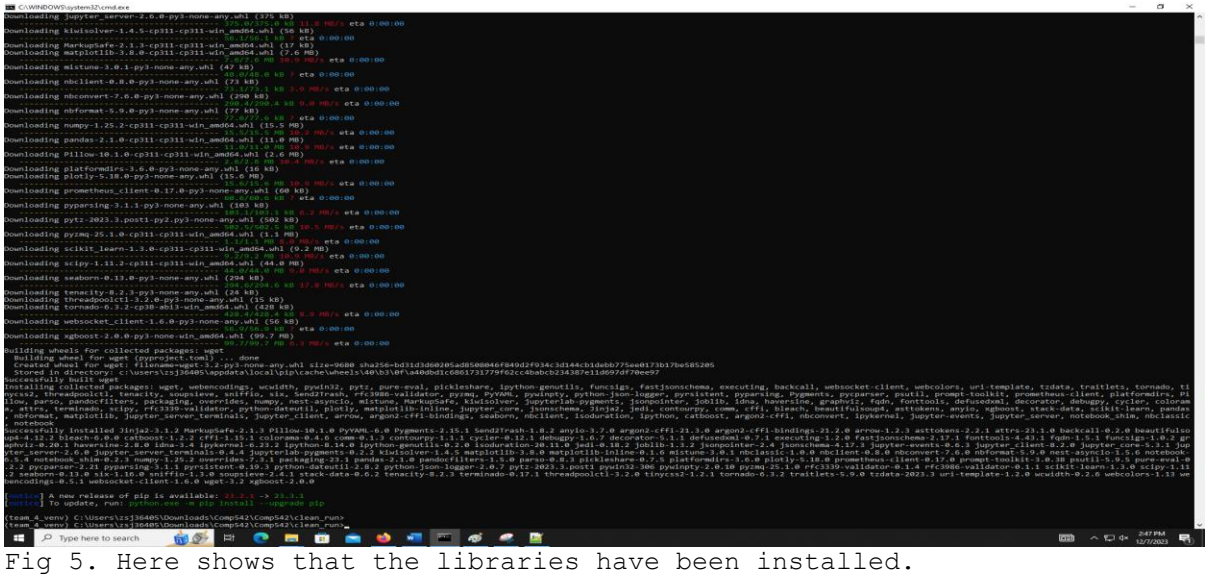
- In the zip folder trace the requirements.txt file and copy the path.
- Go to the CMD, type:
- cd </Path_to_Requirements.txt_file_folder>



- On pressing Enter, Fig 3. Shows that the directory has changed.
- Now, type:
- pip install -r requirement.txt
- Press Enter.

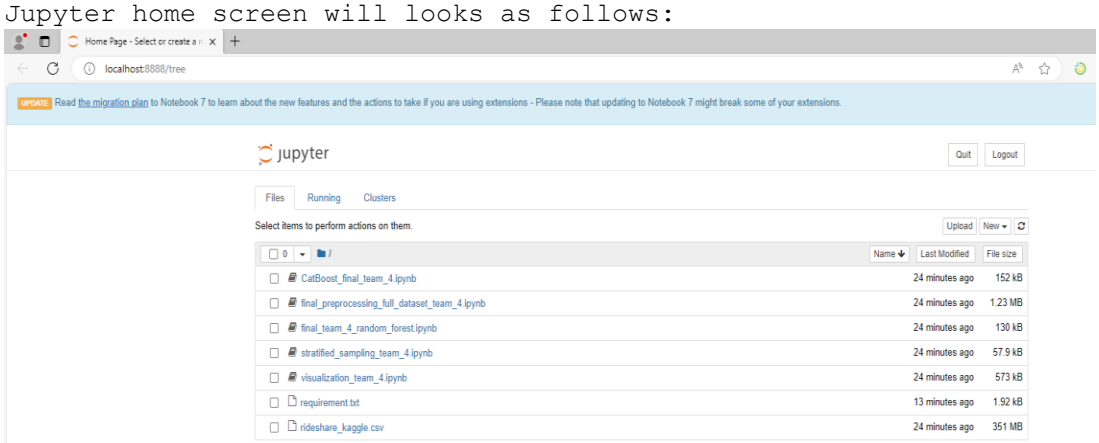


- The libraries will start installing.
- Once installation completes, the CMD will look as below:

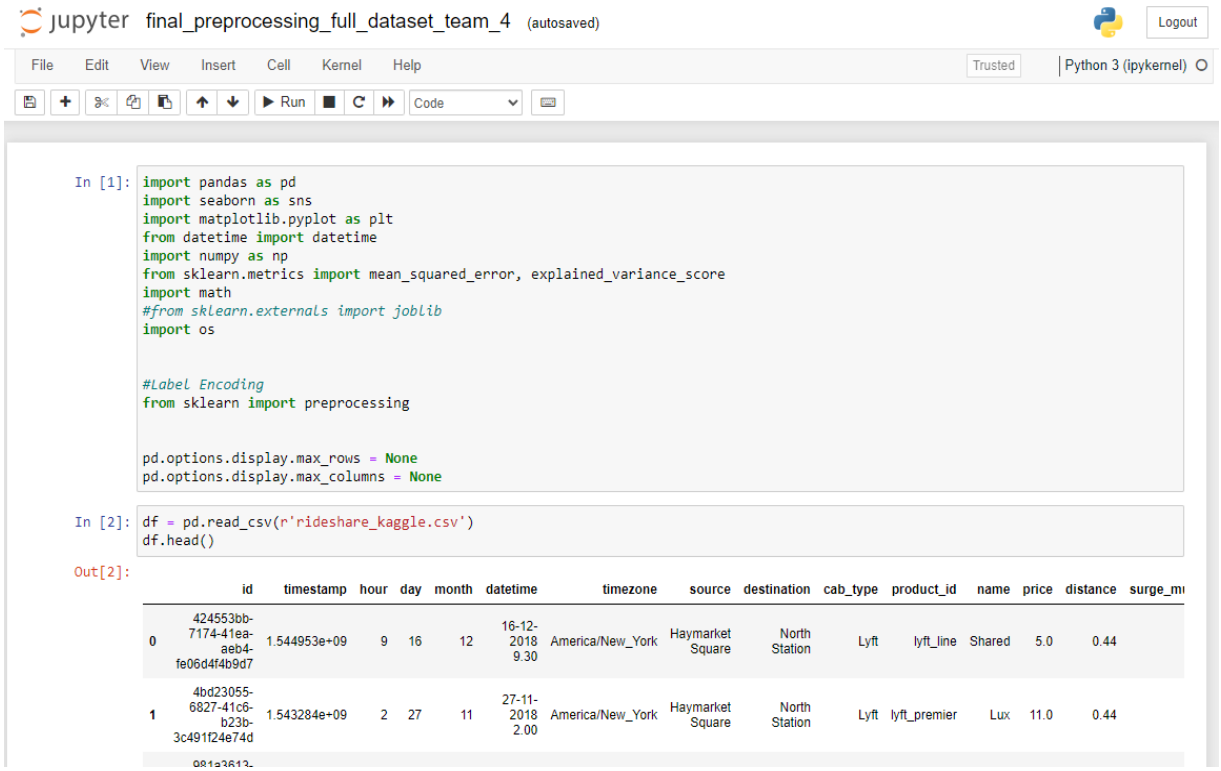


- Fig 5. Here shows that the libraries have been installed.

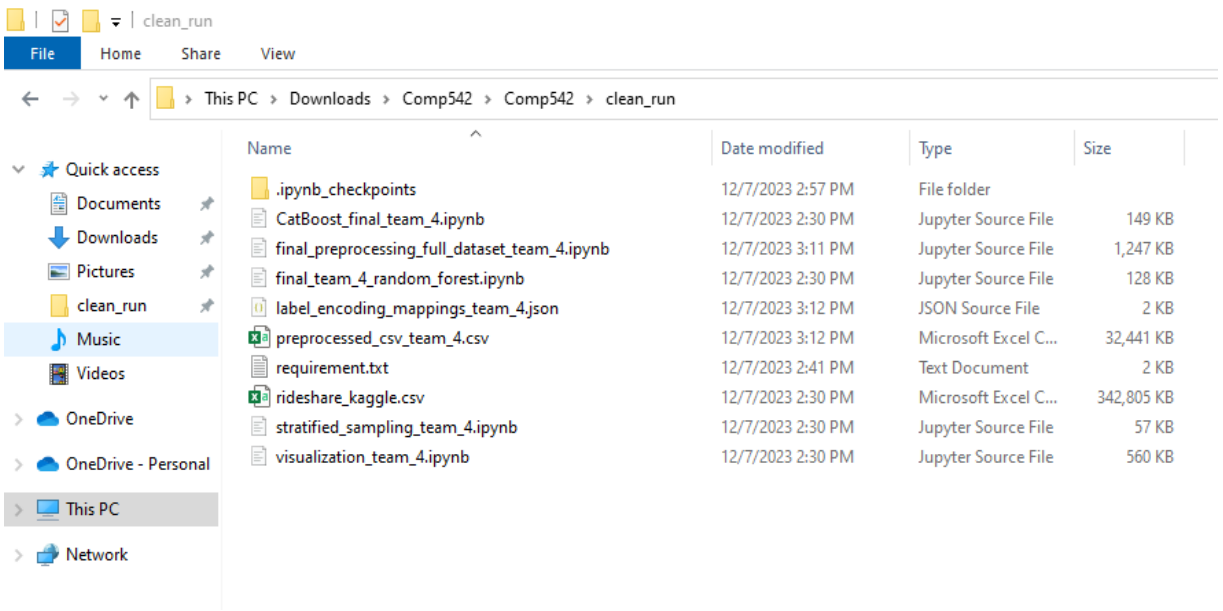
- **For the next step, we start with the PREPROCESSING.**
  - In the same CMD, open Jupyter notebook by typing:
  - ***Note that the path of the CMD is still the Clean Run folder.***
  - Jupyter home screen will looks as follows:



  - **Fig 6. Here shows the jupyter homescreen.**
  - Open the script : final_preprocessing_full_dataset_team_4.ipynb by clicking on it.
  - Execute each cell in the order it appears by pressing Ctrl + Enter

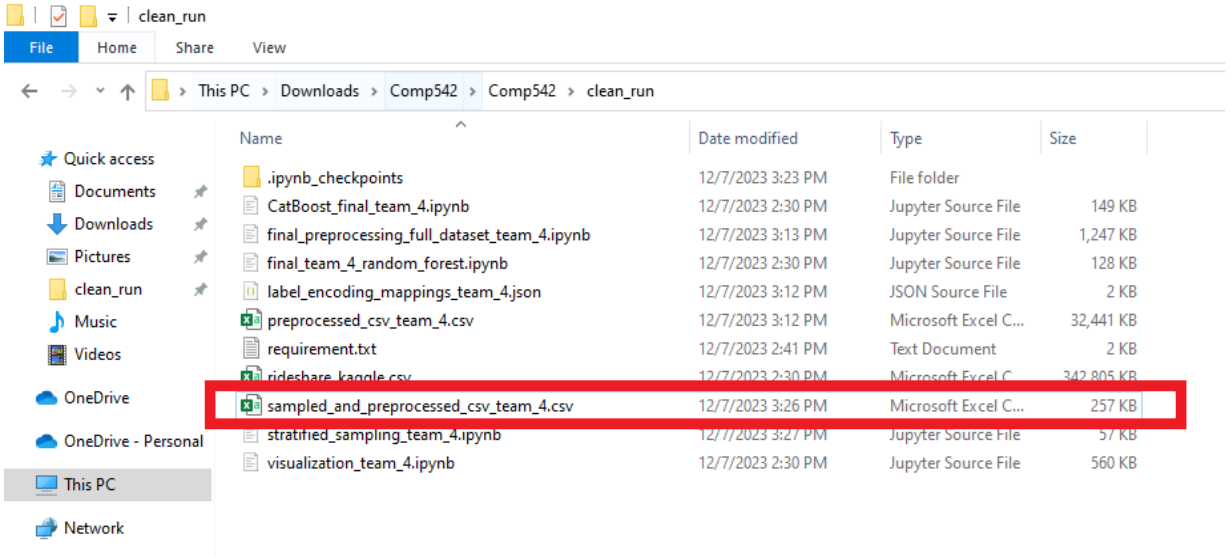


  - Execution of each cell in the script.
  - After executing each cell, the contents of the Clean Run folder will look as follows:



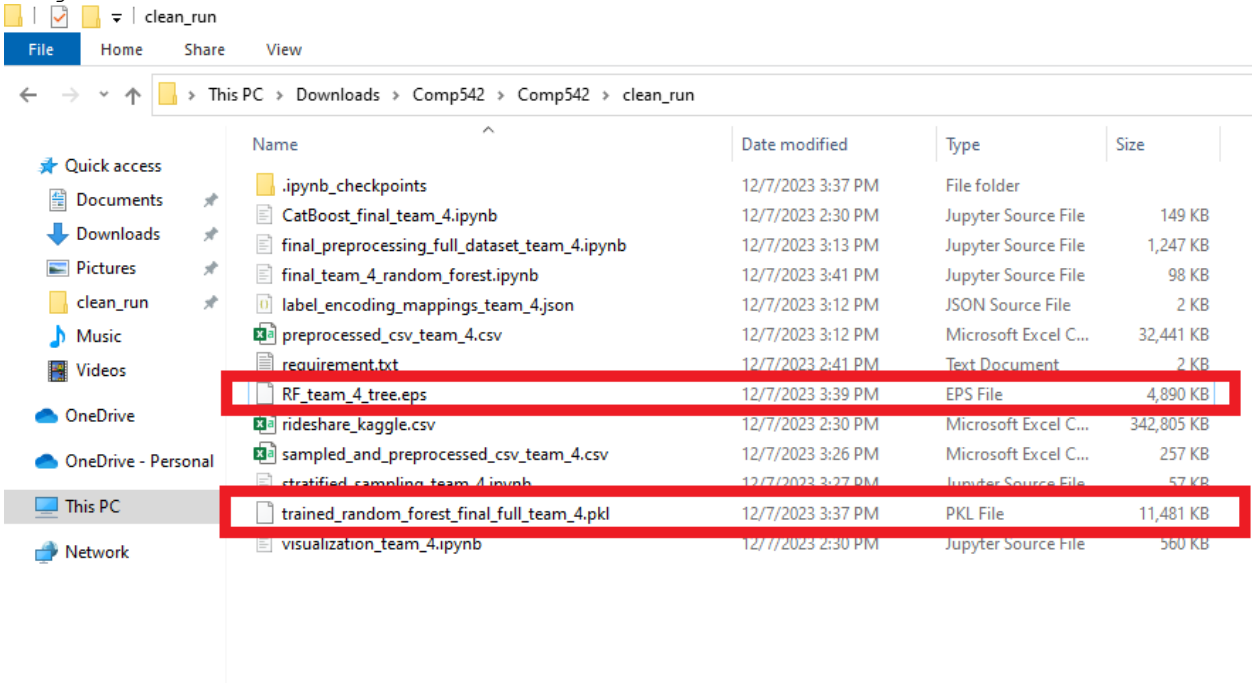
- Fig 7. Shows the updated contents of the Clean Run folder.
- The label_encoding_mappings_team_4.json contains the Label Encoded mappings for the categorical data from the dataset.
- Preprocessed_csv_team_4.csv will be the input to the next script.

- **Now, we run the STRATIFIED SAMPLING.**
  - Refer to Fig. 6. Go here and open the script: stratified_sampling_team_4.ipynb
  - Execute each cell in the script by pressing Ctrl + Enter
  - Once the script is fully executed, the Clean Run folder will have an additional file: sampled_and_preprocessed_csv_team_4.csv



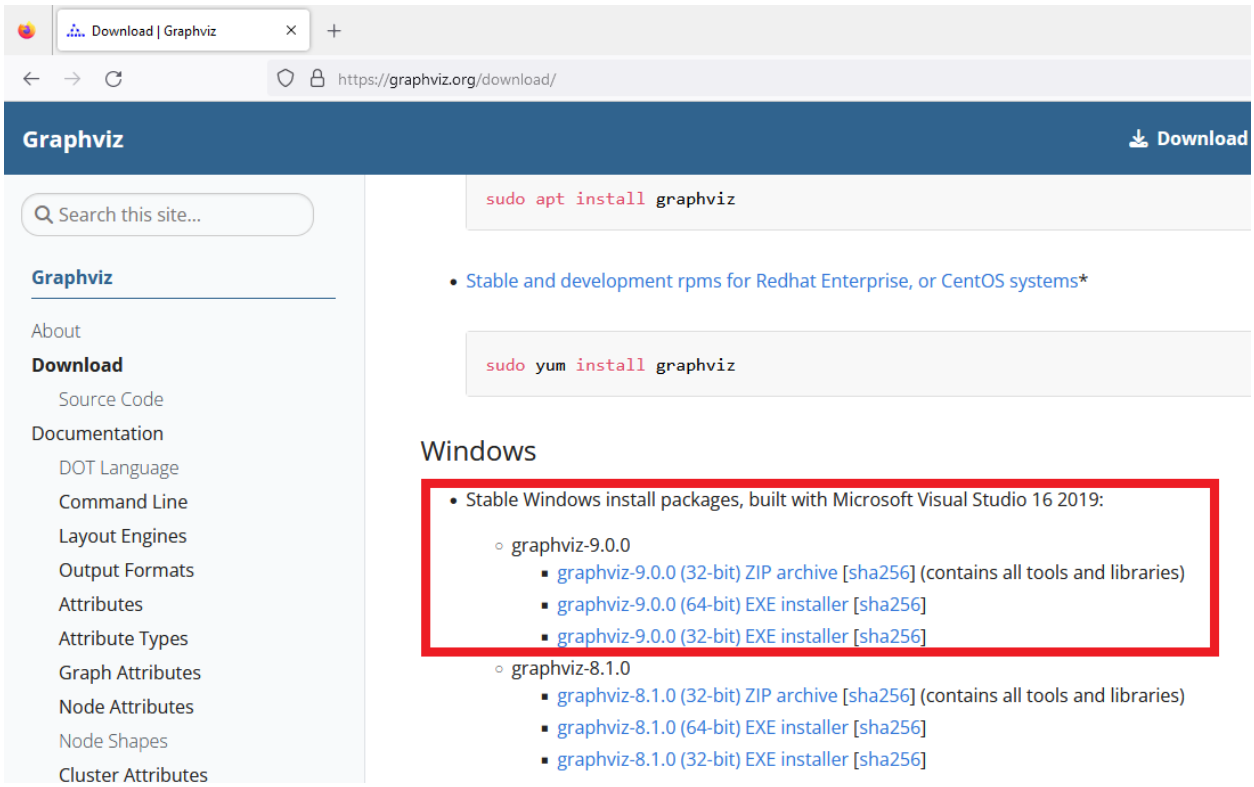
- Fig. 8 shows the new file available in our base directory. This file will be used as an input to later stages.

- **TRAINING OF RANDOM FOREST MODEL**
  - Refer to Fig. 6 again. Go here and open the file named: final_team_4_random_forest.ipynb
  - Execute each cell in the script by pressing Ctrl + Enter
  - Note – The tree plotting will take nearly 150 seconds.
  - Once the script is fully executed, the Clean Run folder will have additional files: trained_random_forest_final_full_team_4.pkl which is the model dump and RF_team_4_tree.eps which is the tree image.



  -
  - Fig 9. Shows the newly generated files in the clean_run folder.

- **INSTALLATION OF GRAPHVIZ**
  - Before proceeding further with other models, we need to install the Graphviz software to visualize Trees for XGboost.
  - Go to: https://graphviz.org/download/
  - And download the files for graphviz-9.0.0
  - Select any of the files based on the system specifications and install.



- Fig. 10 shows Graphviz packages available for download.

- **TRAINING OF XGBOOST MODEL**
  - Refer to Fig. 6 again. Go here and open the file named: final_team_4_random_forest.ipynb
  - Execute each cell in the script by pressing Ctrl + Enter
  - **Update the path to Graphviz binary here:**

  - Fig 11. Shows the cell where the Graphviz path needs to be updated.
  - If the Graphviz path is not updated, there will be an error in the second last cell.

- **TRAINING OF CATBOOST MODEL**
  - Refer to Fig. 6 again. Go here and open the file named: CatBoost_final_team_4.ipynb
  - Execute each cell in the script by pressing Ctrl + Enter
  - **For the CatBoostRegressor, if GPU is present in your hardware, choose that. It gives a better result.**
  - If GPU is not present the default code will work.
  - If GPU is present, update the following to

```
In [ ]: #Model Training

        start_time = time.time()

        model = CatBoostRegressor(task_type='GPU', random_seed=13, verbose=200, loss_function='RMSE', learni

        model.fit(X_train, y_train)

        end_time = time.time()
```

- Here, **task_type='GPU'** has been set.
- The default code with lower accuracy is:

```
In [8]: #Model Training

        start_time = time.time()

        model = CatBoostRegressor(task_type=None, random_seed=13, verbose=200,

        model.fit(X_train, y_train)

        end_time = time.time()

        print("CatBoost Model Training Time (seconds) : ", round(end_time - st
```

- **Visualization of data**
  - Refer to Fig. 6 again. Go here and open the file named: visualization_team_4.ipynb
  - Execute each cell in the script by pressing Ctrl + Enter

For any issues, please reach out via email or cell.