

PART:3

Self-contained Setup Guide

To use our system user first needs to install Anaconda for Python which contains most of the libraries pre-installed which we have used.

Steps to run our system:

1. Install Anaconda for your operating system from this [link](#)
2. After installing it, open Jupyter Notebook
3. Navigate to the folder where our project is saved
4. Open Jupyter Notebook from that folder
5. Install required python packages as mentioned in requirements section from anaconda shell

Requirements

User needs to install certain packages before running our project as follows:

- Pandas
 - To install pandas run `pip install pandas` in anaconda shell
 - We have used pandas for exploring dataset in a form of dataframe
- Numpy
 - We have used numpy for data pre-processing
 - To install numpy run `pip install numpy` in anaconda shell
- Matplotlib
 - We have used pyplot from matplotlib to visualize data in form of graphs.
 - To install matplotlib run `pip install matplotlib` in anaconda shell
- Seaborn
 - We have utilized seaborn to visual data in form of graphs.
 - To install seaborn run `pip install seaborn` in anaconda shell
- Scikit-learn
 - We have extensively used scikit learn library for feature selection, feature scaling, splitting dataset, and performing PCA.
 - We have also used various machine learning models from `sklearn.ensemble` and `sklearn.tree` like Random Forest, GradientBoosting, Extra Trees and Decision Tree.

- To install scikit-learn library user needs to run `pip install scikit-learn` in anaconda shell
- XGBoost
 - We have used XGBRegressor from XGBoost library to build XGBRegressor model.
 - To install XGBoost user needs to run `pip install xgboost` in anaconda shell.
- Keras
 - We have used keras to build Artificial neural network.
 - To install keras run `pip install keras` in anaconda shell.
 - User also need to install TensorFlow using `pip install TensorFlow` in anaconda shell.

Code/Setup Files

- To run our project first user needs to assure that they have all required files in a same folder i.e. dataset and jupyter notebook.
- To run our project user can click on cell and Run all which will run all cells of Jupyter notebook.
- User can run individual cells from the notebook by click Run cells from cells or can use a shortcut key CTRL + Enter.

User-guide for Analysts

- First of all, you make sure that the project file and data you want perform this method remains in the same directory.
- After running all the required pre-processing and model training cells, you can proceed with the Final export of results section where you can analyze the dataset and calculate the loss function.
- Run all the cells and user will get the results of different models but the best performance considering MAE would be an Extra Tree model.
- At last, this method would return the results and calculate the capitalization rate which will be reflected directly in the output dataset.