Build Real Estate Price Prediction Model with NLP and FastAPI
Project Overview

Business Overview

Our client is a Real Estate aggregator company that lists properties across the country on their platform. Property owners can enlist their properties in the platform, and the customers can directly contact the owners if they like to enquire about a property.
The customers found that similar properties in the same area were significantly different in price. They have contacted the support team and raised the issue multiple times. This inconsistency in pricing is creating a lack of trust on the platform and hence the company called us to build a price discovery and regulation model that would estimate the price range of property given its attributes like area, apartment type, amenities, etc.
This project involves building a regression model for price prediction, developing a web application for the same using the FAST API framework, and deploying it on Heroku.

Aim

To predict the price range of a new listed property based on attributes like area, apartment type, amenities, etc.

Data Description

The dataset contains information about 200 properties in Pune, Maharashtra, India, on various attributes such as area, amenities, description, apartment type, etc.

Tech Stack

- ➔ Language: Python
- ➔ Libraries: pandas, numpy, scipy, matplotlib, seaborn, sklearn, nltk, statsmodel

Approach

1. Data Reading
2. Data Preprocessing
    - Categorical Data Cleaning
    - Continuous Data Cleaning
    - Using Regex Library
    - Univariate Data Analysis
    - Multivariate Data Analysis
    - Outlier Treatment

- Feature Extraction
- Text Data Processing
- Parts of Speech Tagging
- Count Vectorization and N-grams

3. ML Model Building
    - Linear Regression
    - Confidence Interval
    - Regularization
        - Ridge Regression
        - Lasso Regression
    - Voting Regressor
4. Model Deployment
    - APIs
    - Web Application Development using FastAPI
    - Heroku Deployment
    - Model Inference Pipeline

Modular code overview:

```
input
|_Pune Real Estate Data.xlsx

lib
|_data
|_model
|_EDA.ipynb
|_Feature Engineering_Extraction.ipynb
|_ML Model Building.ipynb
|_Model Inference Pipeline.ipynb

ML_pipeline
|_model_training.py
|_preprocessing.py
|_utils.py

output

engine.py

app.py

Procfile

property_price_prediction_voting.sav

PropertyVariables.py

readme.md

requirements.txt
```

Once you unzip the modular_code.zip file, you can find the following folders within it.

1. input
2. lib
3. ML_pipeline
4. output
5. engine.py
6. app.py
7. Procfile
8. property_price_prediction_voting.sav
9. PropertyVariables.py
10. readme.md
11. requirements.txt

1. The input folder contains the raw data that we have for analysis. In our case, it contains Pune Real Estate Data.xlsx

2. The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. The engine.py script then calls these python functions to run the steps in one go. It can be used to train the model or use the web application as mentioned in the readme.md file.

3. The output folder contains the models and training data python objects saved for inference while training the model through engine.py.

4. The lib folder is a reference folder, and it contains the original ipython notebooks we saw in the lectures, it has its own system with data and models as used in the videos.

5. The FastAPI for the model can be accessed by running the app.py script and Procfile along with app.py, saved voting regressor model, propertyvariables.py and requirements.txt will be used to deploy the application on Heroku.

6. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command pip **install** -r **requirements**.**txt**

7. **All the instructions for running the code are present in readme.md file**

Project Takeaways

1. What is Regression?
2. What are the different Regularization models and their importance?
3. How to preprocess the data in various ways and use Regex to create patterns?
4. How to detect outliers and clip them?
5. How to clean and preprocess textual data?
6. What are Parts of Speech Tagging, and how is it implemented?
7. What is Count Vectorization?
8. What are N-grams, and how to use them to identify the context?
9. What are Confidence Intervals?
10. How are Confidence Intervals used to identify a range for the prediction?
11. Building Regression models - Linear, Ridge, and Lasso Regression
12. What is Ensemble Learning?
13. How to build a Voting Regressor?
14. Building web application using FAST API
15. Deploying the application on Heroku