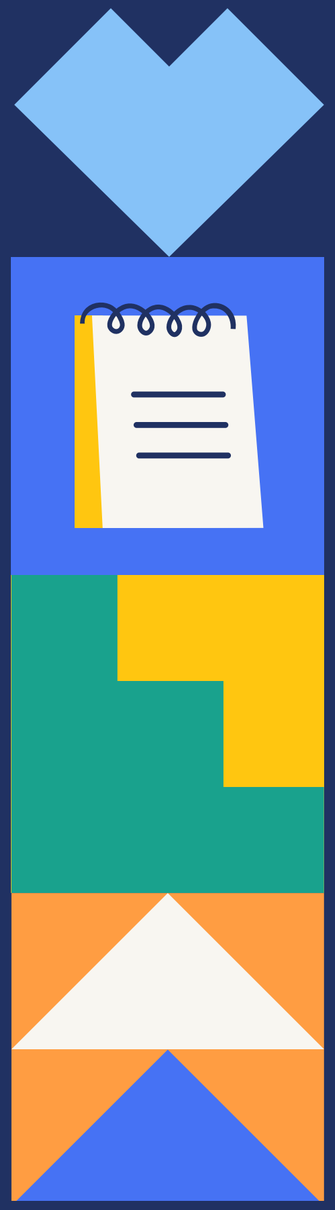
UK Housing Price Prediction Using Pyspark

A comparative study on Machine Learning Algorithms: Linear Regression and Decision Trees using a tabular dataset of real estate properties

Big Data Analytics Lab Mini-Project

By: Aarushi Dharna



Contents:

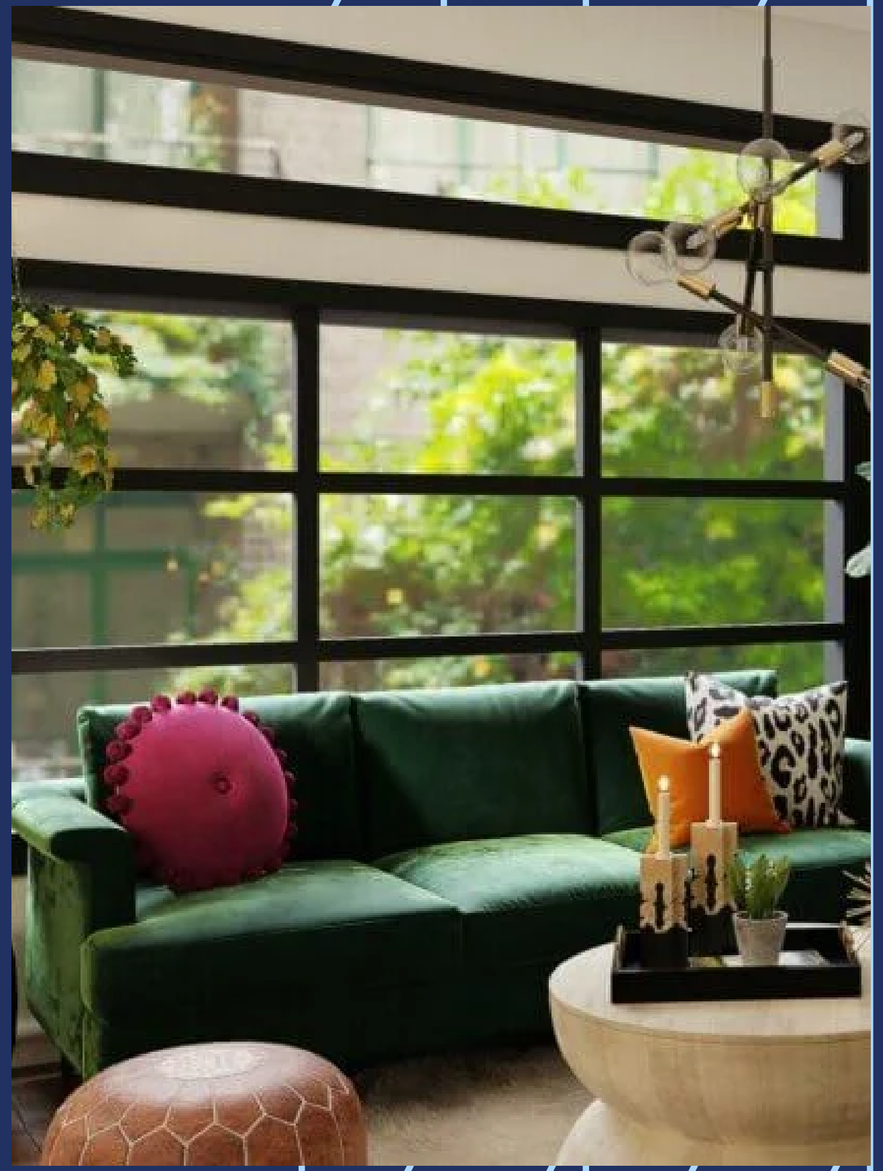
- Problem Statement
- Objectives
- Literature Review
- Data Source
- Methodology
- Results
- Conclusion

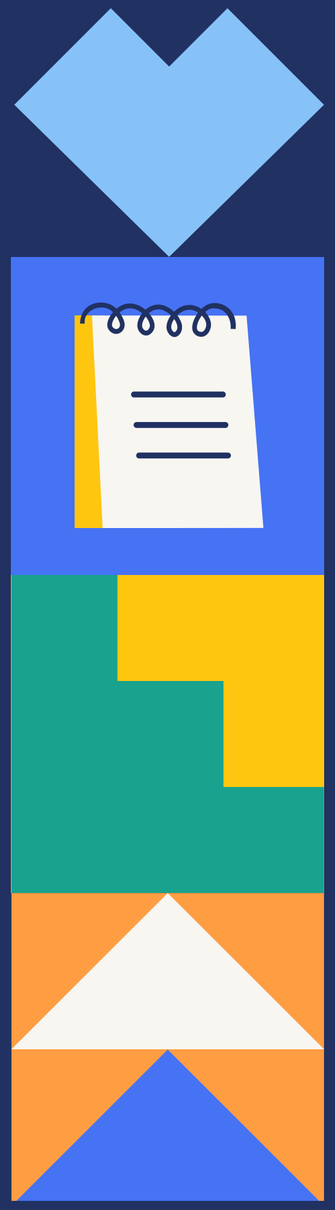
Problem Statement

Housing Price Prediction

- The housing market is a popular investment area for people of many backgrounds. Housing is a critical input in economic, social, and civic development. Many housing-related activities are known to contribute to achieving socio-economic development goals.

- It is critical to analyze and predict the factors that go into giving a particular property its estimated value. These factors may or may not be limited to descriptors regarding the neighboring area, such as street name, building name, postal code, country, and the time period of the sale.





Objective Extablishent

1. Compare supervised machine learning methods in the premise of regressive price prediction :
    a. Linear Regression
    b. Decision Tree Regressor
2. Evaluate methods on various metrics such as RMSE and R squared values.
3. Combat dataset challenges large size, the discrepancy between categorical IVs and continuous DV, and the high range of all variables.

Literature Review

- Preprocessing techniques in *Housing Price Prediction via Improved Machine Learning Techniques* include r**emoving missing data**, **removing ambiguous attributes** (number of category of rooms), **adding an attribute** to indicate the distance from the center of the city of Beijing, etc.
- This paper tested learning methods, including Random Forest, XGBoost, and LightGBM, and two techniques in machine learning, which include Hybrid Regression and Stacked Generalization Regression.

- The paper *Housing Price Prediction Based on CNN* compares ML and DL algorithms for housing price predictions. The experimental results in this paper show that the error of the CNN model converges to 0.01057, and the accuracy is close to **98.68%**, indicating that the CNN model is superior to the GM model and XGBoost model in both the prediction accuracy and the mean square error. **The authors hypothesize that a higher accuracy could be obtained using the ML algorithms with further preprocessing.**

Data Source

- Sourced from the UK government's HM Land Registry.
- 28,000,000 unique points (28276227) corresponding to property prices in the UK
- Only a subset of 200,000 points were used due to limitations in computation and memory capacity
- The dates of the transactions range from January 1995 to April 2023, and they correspond to houses with values ranging from 10,000 to 4,500,000.
- Each data point is associated with 15 attributes

Methodology

Preliminary Statistics

- Obtain minimum and maximum values and unique value counts for each attribute.

Name	Min	Max	Unique values
Transaction_unique_identifier	{00007F1A-EDE3-4EA1-982C-064180AFAC26}	{FFFF5180-5F89-4DDC-906F-E6579B00B9C9}	200000
price	500	4500000	6525
Date_of_Transfer	1995-01-01 00:00:00	1997-12-31 00:00:00	1079
postcode	AL1 1BH	YO8 9QR	58025
Property_Type	D	T	5
Old/New	N	Y	2
Duration	F	U	3
PAON	(MILTON), 38	ZULEIKA HOUSE, 235	31735
SAON	(ANDREWS)	YORK COTTAGE 2	4676
Street	AALTEN AVENUE	ZULLA ROAD	21946
Locality	ABBERTON	YSTALYFERA	6177
Town/City	ABBOTS LANGLEY	YORK	1065
District	ABERCONWY	YORK	453
County	AVON	YORK	128
PPDCategory_Type	A	B	2

Data Preprocessing (1)

Common preprocessing steps:

1. Drop rows that contain missing values for any attributes
2. Convert the Date_Of_Transfer column from data type timestamp to string and extract the year of transfer from it.

For decision trees exclusively:

- Using String Indexer to convert attributes with string type to a unique set of integers with a 1:1 mapping based on unique strings
- Filter attributes with high cardinality (high unique value count) due to limitations in memory and compute power
- Use one hot encoder in a pipeline with a string indexer to obtain one hot encoded vector for all categorical attributes

Data Preprocessing (2)

For linear regression exclusively:

- Using String Indexer to convert attributes with string type to a unique set of integers with a 1:1 mapping based on unique strings
- Obtain correlation coefficients between x and y variables in order to filter out those with low correlation coefficients. (< 0.1)
- min-max scale all values for linear regression to stabilize the learning process

```
The correlation between price and price is: 1.0
The correlation between price and Date_of_Transfer is: 0.07736112392334328
The correlation between price and postcode_index is: -0.039206651590024545
The correlation between price and Property_Type_index is: 0.06068293401736996
The correlation between price and Old/New_index is: 0.08338657061086012
The correlation between price and Duration_index is: 0.02861651482256954
The correlation between price and PAON_index is: -0.06314254054536322
The correlation between price and SAON_index is: 0.07712298571838136
The correlation between price and Street_index is: -0.07103708032651991
The correlation between price and Locality_index is: -0.07625870363489233
The correlation between price and Town/City_index is: -0.17226351079019528
The correlation between price and District_index is: -0.24599619576467302
The correlation between price and County_index is: -0.2263188139085693
The correlation between price and PPDCategory_Type_index is: -0.002198735808304641
```

Apply Machine Learning Algorithms

Linear Regression:

A statistical method to model the relationship between a dependent variable and one or more independent variables by fitting a straight line to the observed data points, aiming to minimize the differences between the observed and predicted values.

- After the aforementioned preprocessing stages, the remaining attributes were ['Town/City_index_Scaled2', 'District_index_Scaled2', 'County_index_Scaled2']
- 198,000 data points were used as part of the training set, whereas 2,000 points were used as part of the test set

price_Scaled2	Town/City_index_Scaled2	District_index_Scaled2	County_index_Scaled2
0.008000888987665296	0.09210526315789473	0.30309734513274333	0.45669291338582674
0.016335148349816645	0.11560150375939848	0.07743362831858407	0.015748031496062992
0.00277808645405045	0.08740601503759399	0.11504424778761062	0.023622047244094488
0.006545171685742861	9.398496240601503E-4	0.008849557522123894	0.05511811023622047
0.005778419824424936	0.02537593984962406	0.3163716814159292	0.22834645669291337

Apply Machine Learning Algorithms

Decision Tree Regressor:

A decision tree is a supervised machine learning model that predicts the value of a target variable by learning simple decision rules inferred from the data features, represented as a tree-like structure.

- After the aforementioned preprocessing stages, the remaining attributes were ['postcode', 'Old/New', 'PAON', 'SAON', 'Locality', 'Town/City', 'District', 'County', 'PPDCategory_Type', 'Date_of_Transfer']
- 198,000 data points were used as part of the training set, whereas 2,000 points were used as part of the test set
- There was an attempt to use ParamGridBuilder to tune hyperparameters however, due to limitations in RAM, individual decision trees were made and trained instead
- Hyperparameters maxBins, maxDepth and minEntropyGain were varied in this setup

Results

- The metrics RMSE and R Squared were used to evaluate the performance of the regression models, inferences will be discussed in the conclusion section

Model Algorithm	Max Depth	Max Bins	Attributes Used													R^2	
			Date of Transfer	Post code	Property Type	Old/ New	Duration	PAON	SAON	Street	Locality	Town /City	District	County	PDD Category	Train set	Test Set
Linear Regression	N/A	N/A										✓	✓	✓		0.068	0.051
Decision Tree Regressor	Not Set	30000										✓	✓	✓		0.225	0.183
	266	10	✓	✓		✓		✓	✓		✓	✓	✓	✓	✓	0.593	0.361
			✓	✓		✓		✓	✓		✓	✓	✓	✓	✓	0.664	0.352

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}}$$
$$= 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}.$$

Conclusion: Key Inferences

- Decision trees perform better than linear regression, even with the same number of features.
- Linear regression has a more even ratio of train: test fitting measure, whereas all the decision trees (with varying hyperparameters) showcase a significantly better test accuracy than train accuracy.
- Increasing the depth of the decision tree leads to a better train accuracy with a worse test accuracy (higher error), which points towards the possibility of overfitting.
- Using more variables in decision trees helps it learn the training set better (increases train accuracy)

Thank You