Housing Value Predictor

Team B:

Jonathan Cook

Sasha Munir

Harshal Shelare

Bezawit Tafesse

Babacar Tall

Understanding the task

- The purpose is to predict the final sale price of houses
- Set of data available with features for Texas
- Finding features correlated with the price factor
- One feature could be safety
- Correlation between crime rate and price

Project Overview

Understanding problem

We wanted to analyze what features could cause changes in prices

Acquiring Data

Collected data that is publicly available and free to use

Cleaning DataSet

We used graphs to find null values, remove outliers and clean data for regression use

Finding Correlations

For features that we wanted to analyze we wanted to see what factors could affect prices.

We did not account for unforeseen events like Covid

Regression Models

Understanding the Dataset

- Our Dataset features monthly data all the way back to 1990.
- We have used 2 different sources to acquire the data.
- We merged the data collected for the different crime rates of different areas and house prices into one data file.

Data Collection

- The first data set was from an official source that has collective data of real estate sales over the past years for each city: Texas A&M University's Texas Real Estate Research Center
    - https://www.recenter.tamu.edu/data/housing-activity/#!/activity/State/Texas
    - This includes data about housing activity trends in Texas since January of 1990.
- The second data set was crime rates of cities for each month over the last few decades: FBI's Crime Data Explorer
    - https://crime-data-explorer.app.cloud.gov/pages/explorer/crime/crime-trend
    - Data about crime trends and statistics in Texas since 1990.

Cleaning the Data

After collecting and joining the data for different areas, we cleaned the data:

- Eliminated duplicates
- Took out null or incomplete entries
- Transformed the data to make it usable. For example converted the object datatypes to int, float, and str datatypes.

```
root
 |-- Sales: double (nullable = true)
 |-- DollarVolume: double (nullable = true)
 |-- AveragePrice: integer (nullable = true)
 |-- MedianPrice: integer (nullable = true)
 |-- TotalListings: integer (nullable = true)
 |-- MonthlyInventory: double (nullable = true)
 |-- Year: integer (nullable = true)
 |-- Month: integer (nullable = true)
 |-- City/MSA: string (nullable = true)
 |-- YearlyCrimesReportedByArea: double (nullable = true)
```

Finding Correlation

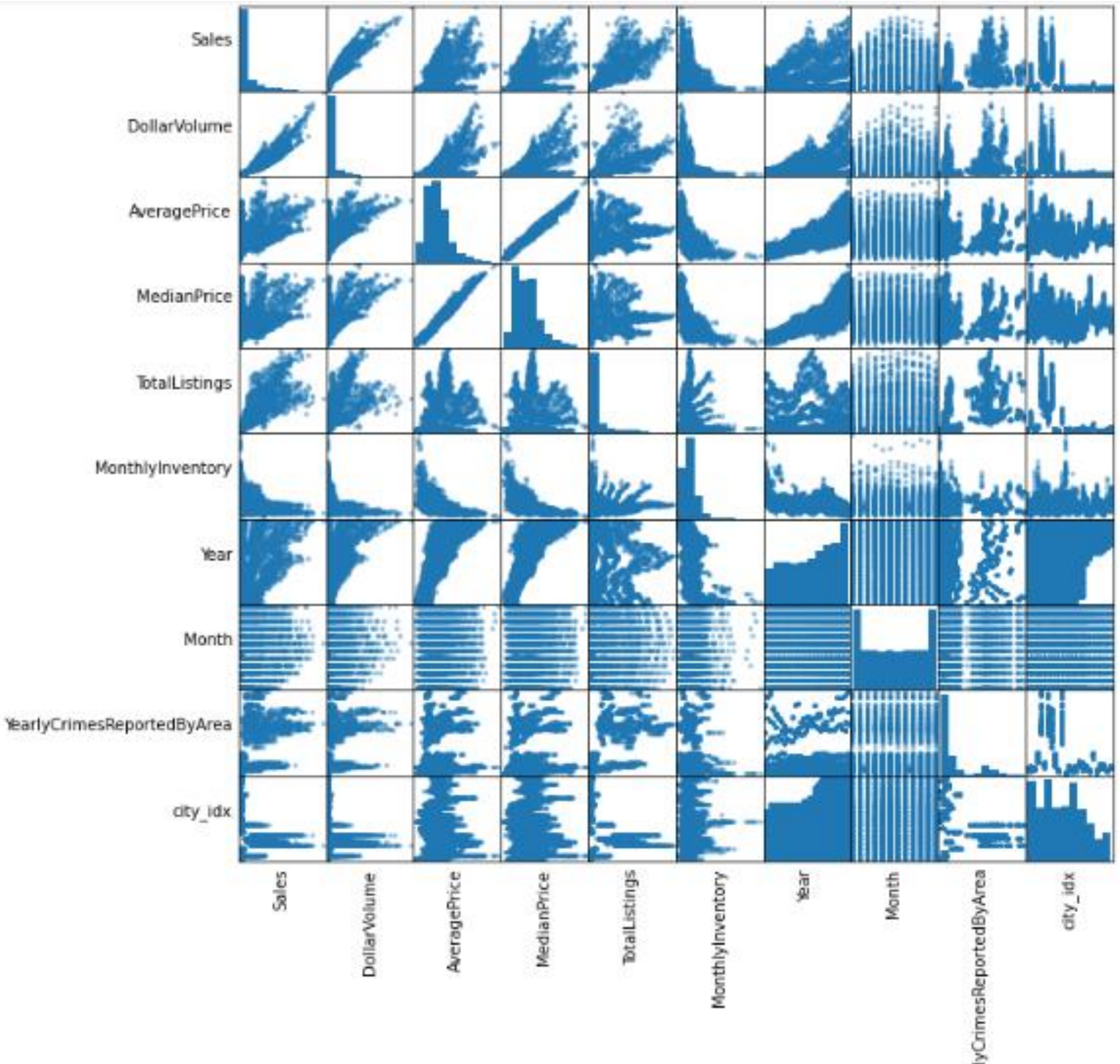
- Correlations between the different attributes and the median price of homes in each area
- 9 factors that were included in the dataset
- Some have strong correlation and others are not strongly correlated

```
for i in df.columns:
    if not( isinstance(df.select(i).take(1)[0][0], six.string_types)):
        print( "Correlation to MedianPrice for ", i, df.stat.corr('MedianPrice',i))
```

```
Correlation to MedianPrice for  Sales 0.41893803077457777
Correlation to MedianPrice for  DollarVolume 0.5416708530093095
Correlation to MedianPrice for  AveragePrice 0.9798320225029512
Correlation to MedianPrice for  MedianPrice 1.0
Correlation to MedianPrice for  TotalListings 0.1779888649917788
Correlation to MedianPrice for  MonthlyInventory -0.5203159144436832
Correlation to MedianPrice for  Year 0.7900672017310437
Correlation to MedianPrice for  Month 0.0545017180044795
Correlation to MedianPrice for  YearlyCrimesReportedByArea 0.1272429745105624
Correlation to MedianPrice for  city_idx -0.02879406369296482
```

Visual representation of Correlation

- We excluded features that did not result in a better model.
- Therefore, we ended up selecting sales total listings, inventory, year, month, crime rate, and city as features.
- These gave us a better accuracy



Vectorization

- The features are converted into vectors using Vector Assembler.
- These are then grouped into one column named features to be used in the feature parameter in our regression model and create another data frame.
- So, as a result, we get a data frame with a features column with all the features and a dependent variable column.
- Then it is split into training and testing data frames for our regression model.

Machine Learning Model

Regression Model

- Model: Multiple Linear Regression
- Assumptions:
  - The distribution is probabilistic. This assumption is often violated.
- Why:
  - The parameters of this distribution define the shape of the of the distribution.
  - If you know these parameters, then you also know the probability associated with the distribution.
- Probabilistic modeling is about estimating the parameters of the distribution.

Regression Model

- Predicting “Median Price” as our dependent variable to see what factors influence it.
- We had tried different ways of building our model.
- The R squared value for our train model is 77.5%.
- Our RMSE value is 24,300. This shows the range of prices that it includes and indicates a good model.

```
lr = LinearRegression(featuresCol = 'features', labelCol='MedianPrice', maxIter=10, regParam=0.3, elasticNetParam=0.8)
lr_model = lr.fit(train)
print("Coefficients: " + str(lr_model.coefficients))
print("Intercept: " + str(lr_model.intercept))
```

```
Coefficients: [10.728746314079904,-0.8013307328105947,-1568.8961489273383,4231.423686449378,595.8045634756928,-0.1157162301918548,-1039.5124707425193]
Intercept: -8351169.332897272
```

```
trainingSummary = lr_model.summary
print("RMSE: %f" % trainingSummary.rootMeanSquaredError)
print("r2: %f" % trainingSummary.r2)
```

```
RMSE: 24299.620812
r2: 0.775865
```

Output

- We were able to attain an R squared value of 77.3% with our test model.
- The RMSE value we got is 24,299.
- •Normalized RMSE = RMSE / (max value – min value)
- 24299/(300000-55000)= approx. 0.099
- This explains that the housing prices predicted by our model could be higher or lower than the actual value.

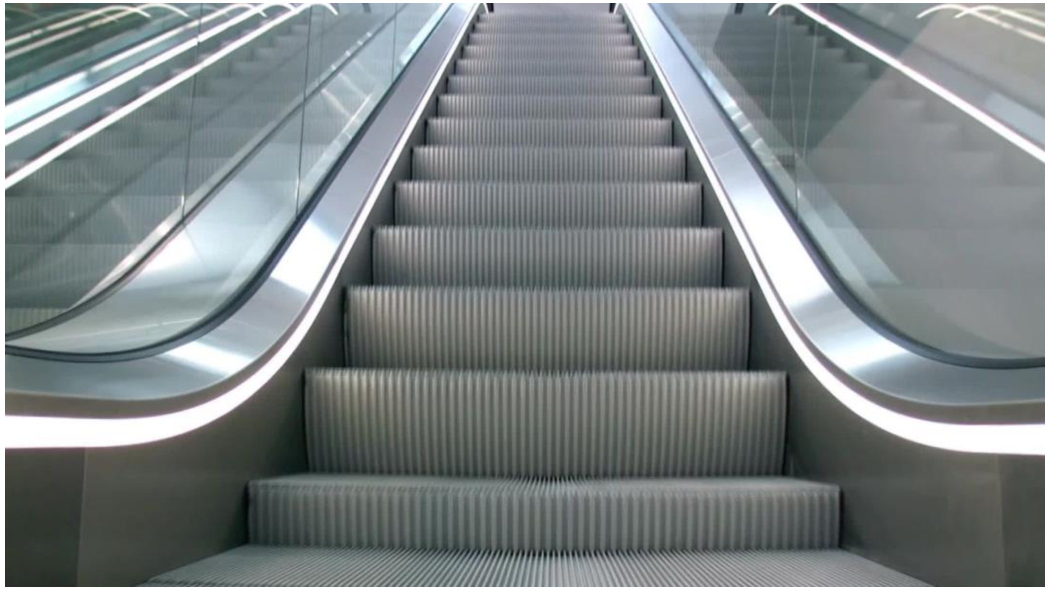
prediction	MedianPrice	features
16627.42180417478	57355	[20.0,1092.0,27.8,1990.0,5.0,43.0,11.0]
53255.0657143835	65290	[25.0,290.0,7.4,1992.0,1.0,689.0,13.0]
47824.669786276296	64178	[27.0,569.0,10.7,1991.0,1.0,495.0,9.0]
70295.70567333698	72489	[28.0,203.0,4.7,1995.0,1.0,616.0,13.0]
127074.19643089361	117722	[29.0,378.0,6.2,2009.0,1.0,431.0,13.0]
60671.84897136688	60329	[30.0,259.0,6.9,1992.0,12.0,689.0,13.0]
52217.06512439065	64220	[30.0,320.0,8.8,1991.0,10.0,706.0,13.0]
150926.264121918	122000	[30.0,454.0,7.9,2017.0,1.0,478.0,20.0]
57761.11494839378	65049	[30.0,559.0,10.5,1990.0,12.0,638.0,2.0]
70766.00553013012	63929	[31.0,204.0,4.8,1995.0,2.0,616.0,13.0]
42930.16474346444	57049	[31.0,821.0,18.2,1993.0,2.0,49.0,11.0]
29561.161746699363	44774	[31.0,875.0,21.3,1991.0,2.0,43.0,11.0]
58868.841009132564	67139	[33.0,279.0,7.3,1992.0,10.0,689.0,13.0]
156019.46297323704	129400	[33.0,442.0,7.4,2018.0,1.0,171.0,20.0]
137505.52483135462	139000	[33.0,635.0,12.1,2014.0,11.0,325.0,20.0]
134786.10344102606	114125	[34.0,370.0,7.1,2011.0,2.0,386.0,13.0]
91683.8504902143	89804	[35.0,270.0,5.3,1999.0,10.0,442.0,13.0]
47034.62962760776	60329	[35.0,356.0,10.6,1991.0,6.0,706.0,13.0]
147955.66248200275	154950	[36.0,239.0,3.8,2013.0,1.0,397.0,13.0]
59617.67866769992	76866	[36.0,260.0,6.9,1993.0,3.0,576.0,13.0]

only showing top 20 rows

R Squared (R2) on test data = 0.773298

Conclusion

- Importance of Data Cleaning and Transformation
- Model Improvement to get best model
- Our model considered sales, total listings, monthly inventory, year, month, city and crime rates reported by area.
- Based on the RMSE value we got, we chose that the prediction from the linear regression reflects the best range.



Thank you!