**CS\_31**

طارق محمد السيد الخطيب

2021170270 - CS

عبدالرحمن شريف مصطفى أحمد

2021170275 - CS

شهاب الدين عصام ابراهيم

2021170258 -CS

انس احمد محمد الفقي

2021170665

**Apartment Rent Prediction**

1 -Data preprocessing

**Null values**

* For object columns replacing columns amenities, pets allowed, city name , state with the mode value
* Fill address using ffill method
* For int/float columns replacing columns bathrooms, bedrooms with the mean value

**Outliers**

* the target column price display distribution:

A graph showing a curve

Description automatically generated with medium confidence

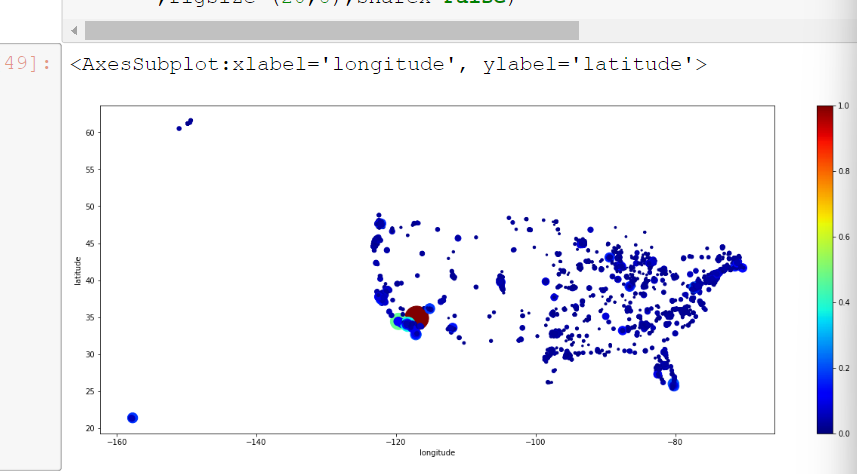
Solved using Winsorization (capping the high values into 99th percentile)

**data engineering**

* Extract road name from address column and replacing it with address column
* Creating a new column “price category” for better visualization between the target column and the object columns
* replacing title , body columns by it’s size (ex: long body ,short tittle)
* for the ‘amenities’ column we found the most frequent 20 word by splitting each value on (‘,’) then replacing it with the frequency of finding each item in the row **similar to frequency encoder**
* for the city name column, we have 1400 different city but we are only interested if the city regularly offers a high or low price so we

have created a new column ‘'city\_offer\_what'’ by grouping the cities and find the mean of the prices that see offers and assigning ‘high’,’avg’,’low’ to the new column.

* for longitude and latitude



We created a new column for this area longitude(-123,-114) latitude(30,38) -> “in area” column

* transform price display column to int using regex([^\d.])

**Encoding**

* apply label encoder for

['city\_offer\_what','category','source','pets\_allowed','has\_photo','cityname','state','citypstate','price\_category','currency','fee','price\_type','address','title','body']

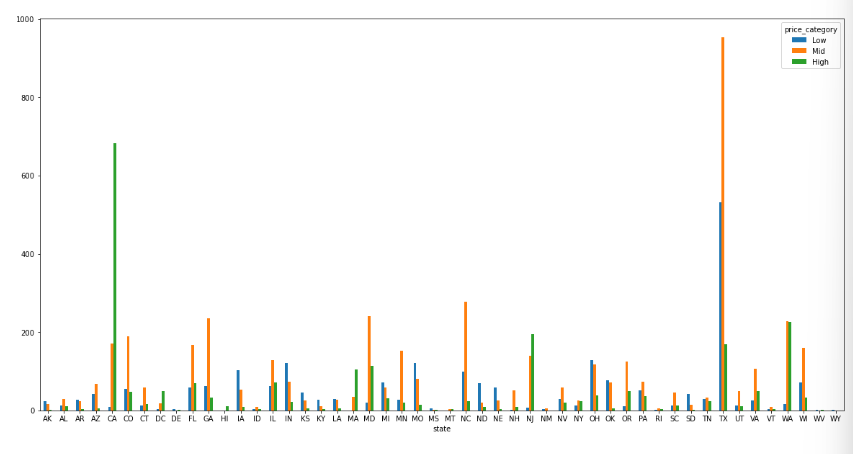
**Exploring the data**

* Features with no affect on the target column they are almost one value.

A screenshot of a computer screen

Description automatically generated

* For states we found that CA offers the highest price



* Looking for the most frequent cities we have Los Angeles dominates A graph of different colored bars

  Description automatically generated
* ‘amentities’ after encoding and preprocess. A graph with different colored bars

  Description automatically generated
* Square feet distribution A graph showing a number of numbers

  Description automatically generated with medium confidence
* Price outliers

A graph with numbers and lines

Description automatically generated

2 – Feature extraction

A screenshot of a graph

Description automatically generated

Features dropped :

['time', -> doesn’t have a context in the problem

'currency\_encoded', -> has one value

'fee\_encoded', -> has one value

'price\_type\_enc', - > has one value

'address\_encoded', - > a lot of different values and low corr score

'source\_encoded', -> low corr score

'body\_count', -> low corr score

'category\_encoded', -> low corr score and has almost on value

'pets\_allowed\_en', -> low corr score doesn’t affect the model

'has\_photo\_en', -> low corr score doesn’t affect the model

'title\_count', -> low corr score doesn’t affect the model

'city\_encoded', -> we have already created a new feature from this column

'price\_category'] -> was used only for visualization

Now the features are['amenities', 'bathrooms', 'bedrooms', 'square\_feet','latitude','longitude', 'in\_area', 'state\_encoded',

'city\_offer\_what\_encoded'],

3 – Modeling

Split the data into 80% : 20%

**Applying**

Linear regression

Polynomial regression

Ridge regression

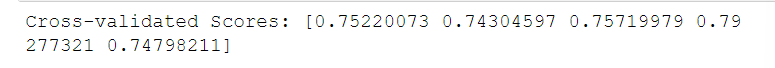
Decision tree regressor

Random forest regressor (n\_estimator = 100)

**We got meansqr\_errors:**

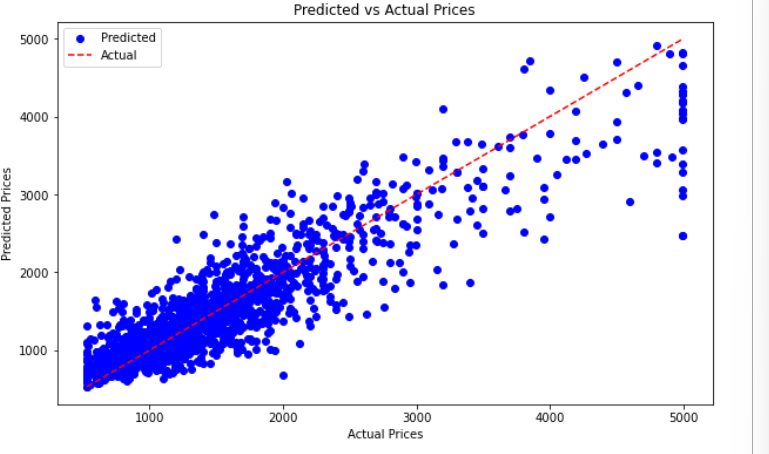
A close-up of text

Description automatically generated

**Using CV =5 in Random Forest we get** 

**And r2\_score**

**Visualizing the test acc Random forest**



**Conclusion:**

* Through the data preprocessing phase, we handled missing values, outliers, and engineered new features to improve model performance.
* Feature extraction helped in selecting the most relevant features for modeling while eliminating irrelevant ones.
* In the modeling phase, we experimented with multiple regression algorithms to find the best-performing model.
* By evaluating the models using mean squared error and R-squared scores, you gained insights into their predictive performance.
* Visualizations, such as plotting linear regression lines and random forest test accuracy, aided in understanding model behavior.
* Overall, this phase demonstrated a systematic approach to building and evaluating regression models for predicting apartment rent prices. The insights gained can guide further iterations to improve model accuracy and generalization.

**Apartment Rent Prediction Classification**

1 -Data preprocessing

Same as the regression column + encoding the target

column

2 – Feature extraction

Same

3- modeling

A graph of different colored bars

Description automatically generated

Random forest with the highest accuracy

training time

A graph of different colored bars

Description automatically generated with medium confidence

Testing time

A graph of different colored bars

Description automatically generated with medium confidence