**PROJECT REPORT**

NAME :- UBAID AHMAD KHANDAY

ROLL NO :- 29

**SECTION 1**

**1.1 Lasso Regression for House Price Prediction**

The real estate market is a dynamic and complex environment where property prices fluctuate based on various factors such as location, size, number of bedrooms, amenities, economic conditions, and more. Accurate house price prediction is essential for both buyers and sellers to make informed decisions. Machine learning techniques, like Lasso Regression, can be employed to build predictive models that estimate house prices based on historical data.

**i) Project Objective:**

The main objective of this project is to develop a predictive model using Lasso Regression that can accurately estimate the prices of houses based on a set of relevant features. By doing so, this project aims to provide valuable insights into the real estate market and assist both potential buyers and sellers in making informed decisions.

ii) **Project Goals:**

1. Data Collection: Gathered a comprehensive dataset containing information about various attributes of houses, including size, location, number of bedrooms, and more.
2. Data Preprocessing: Cleaned and prepared the dataset by handling missing values, categorical encoding, and scaling numerical features.
3. Model Development: Implemented the Lasso regression algorithm to create a predictive model, leveraging its feature selection capabilities.
4. Hyper-parameter Tuning: Fine-tuned the Lasso regression hyperparameters to optimize model performance.
5. Visualization: Visualized key insights and model results using charts and graphs for better understanding.
   1. **Data-Set Used:**

The dataset we are using has been taken from the kaggle.com , The original dataset contains 18

Columns (features) and 4600 instances.

**SECTION 2**

**2.1 Lasso Regression Code With Description:**

* Importing all the necessary libraries for the code

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import warnings

## Reading the dataset

data = pd.read\_csv('data.csv')

## Dropping some columns not important

data=data.drop(['date','street','statezip'],axis=1)

data['city']=pd.factorize(data['city'])[0]

data=data.drop(['country'],axis=1)

## Setting the ‘price’ as the target column

X=data.drop(['price'],axis=1)

y=data['price']

## splitting the dataset for training & testing

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=0)

## importing the lasso regression model from sklearn libraray

from sklearn.linear\_model import Lasso

## taking hyperparameter alpha=0.50

lasso= Lasso(alpha=0.50)

## Training the model

lasso.fit(X\_train, y\_train)

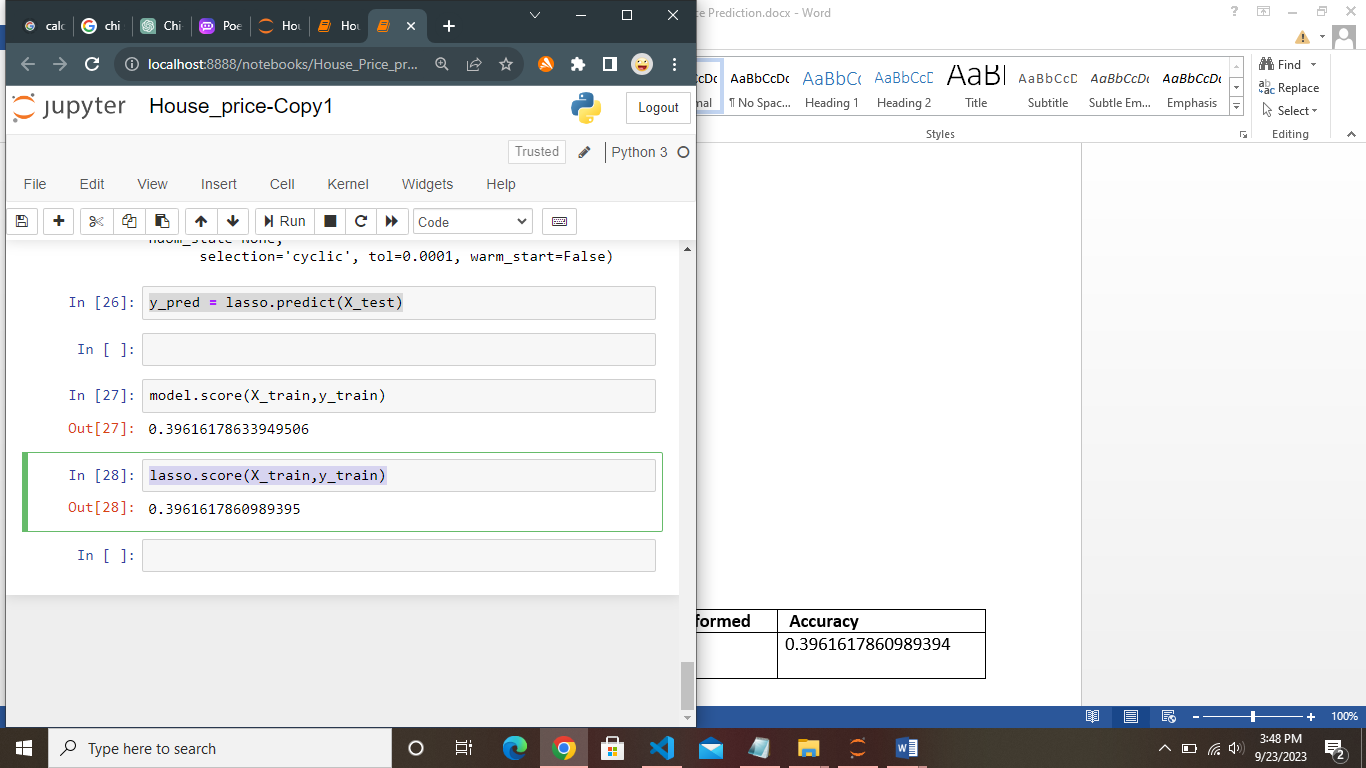
## Making the prediction

y\_pred = lasso.predict(X\_test)

## Finding the accuracy score of the model

lasso.score(X\_train,y\_train)

**2.2 Result**

****

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Operations Performed** | **Accuracy** |
| Lasso Regression | Without Any Preprocessing | 0.3961617860989394 |

**SECTION 3**

**3. Approaches Used to Improve Results:**

**3.1**  Hyper Parameter Tunning:- like alpha and random state

**3.2** Second Version of Code.

* Importing all the necessary libraries for the code

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import warnings

## Reading the dataset

data = pd.read\_csv('data.csv')

## Dropping some columns not important

data=data.drop(['date','street','statezip'],axis=1)

data['city']=pd.factorize(data['city'])[0]

data=data.drop(['country'],axis=1)

## getting some correlation between the columns of the data set

data.corr()

## Finding the null values in the columns

data.isnull().sum()

## Setting the ‘price’ as the target column

X=data.drop(['price'],axis=1)

y=data['price']

## splitting the dataset for training & testing, setting the random\_state=42 so that at every time we run the code we get the same result. Also setting the train:test ratio 80:20

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

## importing the lasso regression model from sklearn libraray

from sklearn.linear\_model import Lasso

## taking hyperparameter alpha=0.50

lasso= Lasso(alpha=0.50)

## Training the model

lasso.fit(X\_train, y\_train)

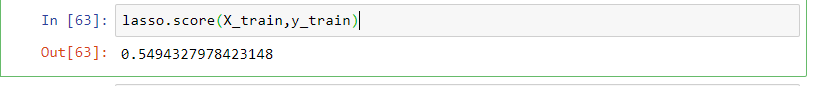
## Making the prediction

y\_pred = lasso.predict(X\_test)

## Finding the accuracy score of the model

lasso.score(X\_train,y\_train)

**3.3 Results**



|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Operations Performed** | **Accuracy** |
| Lasso Regression | Without Any Preprocessing | 0.3961617860989394 |
| Lasso Regression | Hyper\_parameter tunning, test:train ratio=80:20,random\_state=42 | 0.5494327978423148 |

**SECTION 4**

**4. LASSO REGRESSION WITH EXPLORATORY DATA ANALYSIS.**

**4.1 PERFORMING EDA**

**4.2 REMOVING OUTLIERS**

**4.3 REMOVING NULL VALUES AND DUPLICATES**

**4.4 CHANGING DATATYPES OF SOME COLUMNS**

# Importing the required libraries and modules

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

## Reading the dataset

data = pd.read\_csv('data.csv')

## Dropping some columns not important

data=data.drop(['date','street','statezip'],axis=1)

data['city']=pd.factorize(data['city'])[0]

data=data.drop(['country'],axis=1)

## Changing the datatype of columns from object to integer type

data['bedrooms'] = data['bedrooms'].astype('int64')

data['floors'] = data['floors'].astype('int64')

## Plotting the graph for how price is distributed

data['price'].hist(bins=100)

plt.xlabel("Price")

plt.ylabel("frequency")

plt.title("histogram of price")

plt.legend("Price")

plt.show()

# Finding & removing the outliers in price

(data['price'] > 0.3e7).sum()

data = data[~(data['price'] > 0.3e7)]

## finding the number of bedrooms and floors

bedrooms=data['bedrooms'].value\_counts()

bedrooms

floors=data['floors'].value\_counts()

floors

## Replacing the 0 values in price with Nan

(data['price']==0).sum()

data['price'].replace(0,np.nan,inplace=True)

data.dropna(inplace=True)

## Removing duplicate values if any

duplicates = data[data.duplicated()]

duplicates

data.drop\_duplicates()

## Finding the correlation between columns

corr=data.corr()

corr

## splitting the dataset for training & testing, setting the random\_state=1 so that at every time we run the code we get the same result.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

## importing the lasso regression model from sklearn libraray

from sklearn.linear\_model import Lasso

## taking hyperparameter alpha=0.50

lasso= Lasso(alpha=0.50)

## Training the model

lasso.fit(X\_train, y\_train)

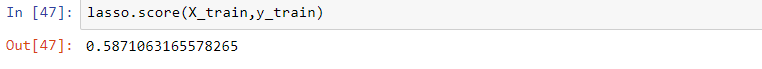
## Making the prediction

y\_pred = lasso.predict(X\_test)

## Finding the accuracy score of the model

lasso.score(X\_train,y\_train)

**4.5 Result**

****

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Operations Performed** | **Accuracy** |
| Lasso Regression | Without Any Preprocessing | 0.3961617860989394 |
| Lasso Regression | Hyper\_parameter tunning, test:train ratio=80:20,random\_state=42 | 0.5494327978423148 |
| Lasso Regression With EDA | EDA,Removing outliers,random\_state=1,Removing Null values | 0.5871063165578265 |