Machine Learning Project: Analysis on Real-estate data

DATA602: Introduction to Data Analysis & Machine Learning

FINAL PROJECT REPORT

Srinivasa Akhil Vutukuri

University of Maryland, Baltimore County

v39@umbc.edu

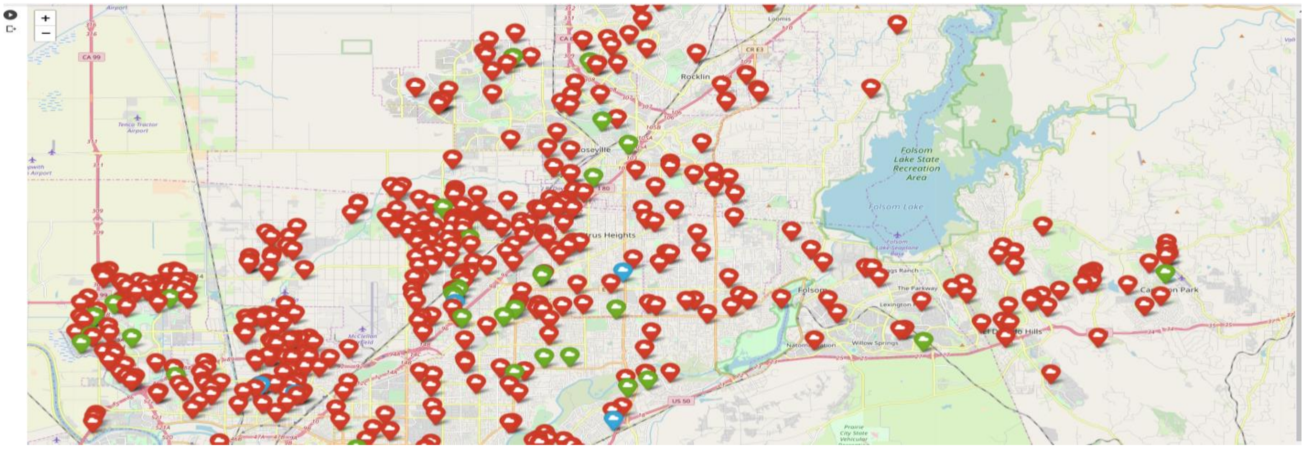
Abstract - We examine analysis on real-estate data. The contribution of this paper: (1) We explore the data using exploratory techniques and find the attributes affecting them. (2) We apply different machine learning techniques to analyze them and predict the outcomes.

Introduction:

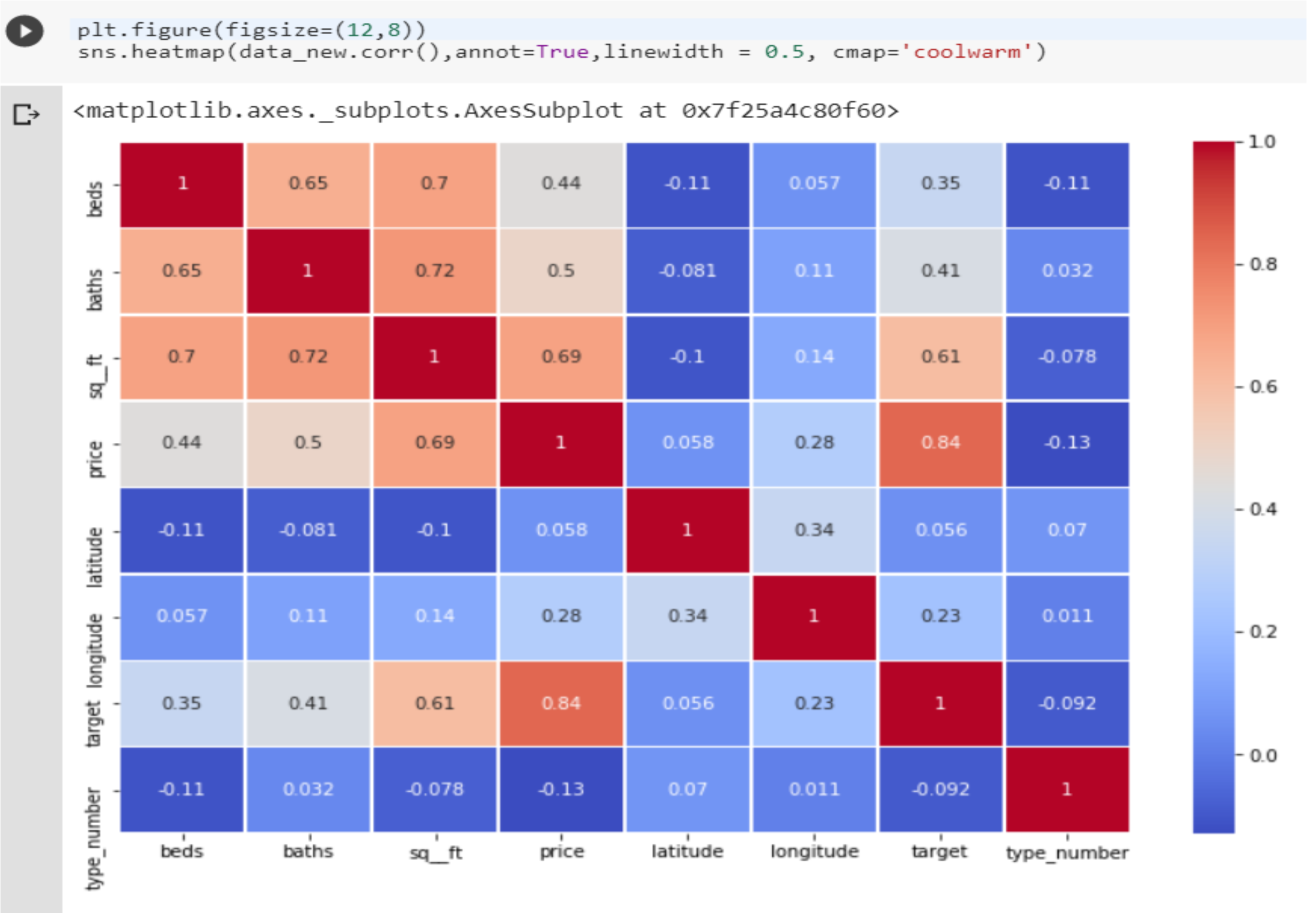
Real-estate is basically the best investment a person can do. This is because the value of properties with similar attributes can cost couple of thousands away. Since there is proper real estate data and the data can be used to predict the future price increase. The data I choose is about the real estate transactions in a place called Sacramento in a span of one week. So, I started to test various algorithms which can predict the price of the house and its range. Due to this algorithm we can also predict approximate cost of a property with your requirement in the area you would like to buy. Many companies are trying to predict to provide accurate price range for many people who are interested in investing their money in real-estate. What I tried to do in this project is to estimate the price range of a house you are looking for and predict them using machine learning techniques.

Analysis:

I started with collecting the data from my source (http://samplecsvs.s3.amazonaws.com/Sacramentorealestatetransactions.csv). I started to analyze my csv and found that it contains various columns like price, number of beds, square feet, are, state etc., I started my initial cleaning by dropping some of the columns which are not used for my analysis like state, city, zip because this dataset contains the transactions of the properties in a single city. I started to look for missing values but found out that there are none. While viewing the description of data I found that there are some entries with zero square area which cannot be true as there are no houses without any area. So, I dropped those rows and wanted to view the data. As there are latitude and longitude in my data I tried to plot them in a world map and differentiate based on the different types of property.



I did some exploratory analysis like correlation and found that bed rooms – number of bathrooms are highly correlated. This is logically true as the bedroom are more bathrooms would increase. Bedrooms and square area are also highly corelated.

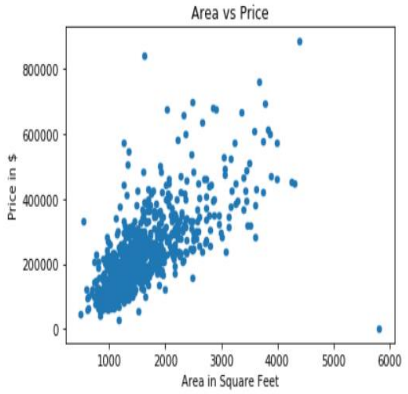


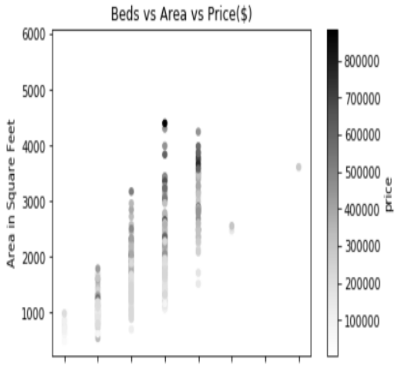
```
plt.figure(figsize=(12,8))
sns.heatmap(data_new.corr(),annot=True,linewidth = 0.5, cmap='coolwarm')
```

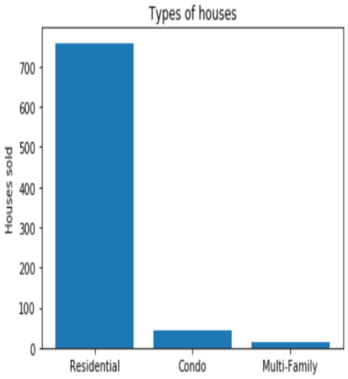
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f25a4c80f60>
```

	beds	baths	sq_ft	price	latitude	longitude	target	type_number
beds	1	0.65	0.7	0.44	-0.11	0.057	0.35	-0.11
baths	0.65	1	0.72	0.5	-0.081	0.11	0.41	0.032
sq_ft	0.7	0.72	1	0.69	-0.1	0.14	0.61	-0.078
price	0.44	0.5	0.69	1	0.058	0.28	0.84	-0.13
latitude	-0.11	-0.081	-0.1	0.058	1	0.34	0.056	0.07
longitude	0.057	0.11	0.14	0.28	0.34	1	0.23	0.011
target	0.35	0.41	0.61	0.84	0.056	0.23	1	-0.092
type_number	-0.11	0.032	-0.078	-0.13	0.07	0.011	-0.092	1

Then I did view the relation between price, beds and area







So, with this I got that with increase of area increases the price but there are some anomalies in this as we can get a 6-bed room house with in a range of 4 bed room house in a different area. Due to this I tried to predict the approximate estimate of house with the required attributes.

Then I did some feature engineering to match my attributes to machine learning techniques. I set my target variable based on my price. I divided my price range into three partitions and I separated all my entries into low, medium, high price houses.

1 - Low priced property

2 – Medium priced property

3 – High priced property

Then I changed all my attributes into numbers as regression models can't process text files so, I changed my type (types of properties) into numbers as it has only 3 different types of entries.

So, after this I applied various machine learning techniques like:

1. Linear Regression:
   linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression.

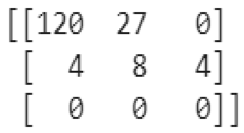
   Train accuracy: 0.0
   Test Accuracy: 0.0

   In this I got worst accuracy scores as my data is categorical and is not linear.

2. Logistic Regression:
   The logistic model is a widely used statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist.

   Train accuracy: 84.6 %
   Test Accuracy: 78.5 %

   Confusion Matrix:



```
[[120  27   0]
 [  4   8   4]
 [  0   0   0]]
```

   As my data is categorical data I got a better accuracy than linear as expected.

3. PCA:

   Basically, PCA is used to find the number of attributes on which your data is highly dependent. So, we can find principal components and the major components affect the model.

   Principal components can be found by explained variance ratio:

   ```
   Array([9.99998036e-01, 9.42993169e-07, 4.16442700e-07])
   ```

   So, we can understand that my primary component is highly dependent on the output.

   So, my principal component is Beds.

4. K means:

   k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis. So, we can estimate number of clusters your data can be divided.

   Train accuracy: 27.8 %
   Test Accuracy: 22.7 %

   I tried to check the k-means with different number of clusters. I tried to look for various number of clusters. I got maximum accuracy when number of clusters =3

5. Random Forest:

   I tried to work on random forest as I thought which would give me better results as there are categorical values which I want to estimate in my dataset.

   Train accuracy: 84.6 %
   Test Accuracy: 79.1 %

   I liked the results I got pretty good accuracy scores unlike my k-means and linear model.

6. Lasso Regression:

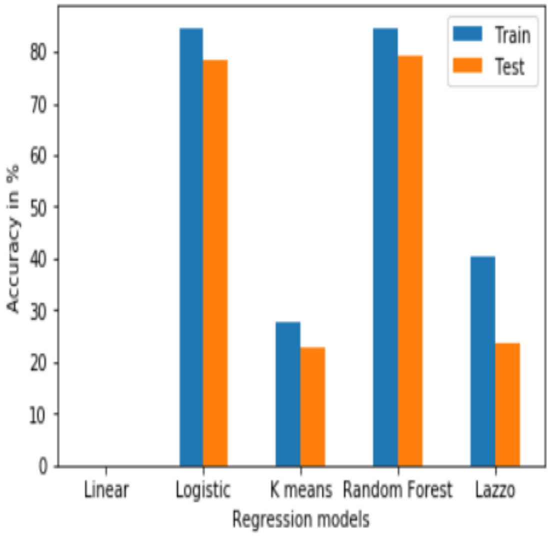
   Lasso Regression is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.

   Train accuracy: 40.4 %
   Test Accuracy: 23.8 %

   These results are not as good as others as the test accuracy is around 24%

Results:

So, by comparing all regression models we can see that Random Forest and Logistic regression techniques provide better accuracy for my dataset.



Conclusion:

As the data I choose has categorical values and I expected these models (Random Forest and Logistic regression) to work as these are most fetching models for data with categorical values. Coming to linear model I thought that I would get low accuracy results but didn't expect that to be zero. K-means and lasso are not quite performing as good as the other models. So, I conclude that Random forest and logistic regression models are better models for my dataset.