­­­

**Module 6: Final Project Report**

# Lakshmi Sravya Vedantham   (002989684 )

Northeastern University -Toronto

Instructor – Mohammad Shafiqul Islam

ALY6110– Data Management and Big Data

June 28th, 2022

# **Table of Contents**

Table of Contents 2

[Introduction 3](bookmark://_bookmark1)

Exploratory Data Analysis 4

Data Modelling 8

Conclusion 13

References 14

|  |
| --- |
| **Introduction**  People nowadays are eager to invest in a variety of items. Property is one of the few things in which everyone is interested in investing. Property can refer to a variety of things, including homes, land, and other assets. This may be beneficial not only in the future but also in long-term planning. This piqued my curiosity, and I decided to concentrate on how to analyze housing data and what knowledge I might glean from it. I wanted to anticipate the worth of a house based on my analysis. California housing is the dataset I have chosen for future investigation. The StatLib database was used to obtain this data. My main objective is to forecast housing values in the California region. This can be advantageous to individuals who want to invest in real estate with caution. My major objective is to examine the provided dataset using Python and machine learning models and to forecast the housing value based on various parameters.  All of California's block groups from the 1990 Census were used to compile the data for the variables. According to the data, 1425 people in this California sample block group live in a geographically congested area. The population density determines the size of the geographical area covered. The distances between each block are calculated using latitude and longitude. Consequently, there are 20,640 observations and 9 features in the final dataset. The dependent variable is the median house value because my major purpose is to forecast house prices. A better understanding of this data is seen in the next section |

**Exploratory data Analysis**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The dataset with almost 20640 observations contains 10 variables as follows. For each block group, longitude and latitude relate to the angular distance of a geographic point north or south, east, or west of the equator. The housing median age refers to the average age of residents in a block group. Total Rooms refers to the total number of rooms in each block group's homes. Total Bedrooms refers to the total number of bedrooms in each block group of houses. The population of a block group refers to the number of individuals who live there. Individual residences and their inhabitants, as well as the median income of people in a block group, are referred to as households. Figure 1 shows the structure of the data set.    Figure 1  The mean, minimum, maximum, standard deviation, and other dataset statistics are shown in Figure 2. As can be seen, the highest housing value is 500001.00, the lowest is 20640, and the average is 206855.  Figure 2  Figure 3  All the histograms for all the variables in the data set are shown in Figure 3. We can observe that the home has a maximum of 1300 members over the age of 50. The overall number of rooms was discovered to be 2000, including 500 bedrooms, in 5000 California blocks. In California, the population is estimated to be 1500 people in 9000 blocks. The median income is around $4000, and the average house value is around $200,000.    Figure 4  Now that we have seen most of the numbers, it is time to look at the correlation. The correlation map for each variable in the given dataset is shown in Figure 4. Low correlation is represented by the darker hue, whereas high correlation is shown by the lighter tint. Population, the total number of rooms, the total number of bedrooms, and households are all connected, as can be shown. All of them are inextricably linked. Since I am more concerned with predicting the house's value, everything hinges on the median income of each block, which is tied to the value of 0.69.    Figure 5  Figure 5 depicts how the value of homes is spread across California. Most of the colors appear to be blue, and the average housing value ranges from $100,000 to $200,000. We can also see that housing values are higher along the coast. As we get closer to the water, the value of homes rises. California's population is represented by the size of the bubble. The larger the bubble, the more people there are. It has been noticed that most of the population lives in coastal locations with high housing values. Figure 6 shows that there are double rooms available for the given population. That means a single person will require at least two rooms. Also, there are 1/3 of the bedrooms for the population count. According to the numbers in the dataset, three people can share a bedroom.   |  |  | | --- | --- | | Figure 6 | Figure 7 |   Figure 7 depicts the relationship between median income and house value. As one's income rises, so do the value of their home. As a result, in this dataset, both variables are significantly connected.  As we did EDA, the upcoming session helps us to predict the house value in California.  **Data Modelling**  The moment has come to handle and model the data. I attempted to model the given data in Google Collab using Spark and Python. To elaborate, I attempted to add a few more columns to the provided dataset and produced the result depicted in Figure 8. We can observe that it appears that three new columns—rooms per household, people per household, and bedrooms per total room have been added. Even better features for the modeling will be provided by these columns.  Figure 8  I tried to use the spark function's vector assembler to generate the features column, as seen in figure 9.  Figure 9    Figure 10  As we are supposed to anticipate housing values, I have now chosen the characteristics column as column features for X and the column medhv (housing value) for Y. The first 5 columns of X and Y are clearly displayed in Figure 10. With a random seed of 42, this dataset is divided into an 80/20 split (80% training set and 20% testing set), and it is ready for modeling.  **Models Used**  *Linear Regression:*  When modeling the relationship between a scalar answer and one or more explanatory variables in statistics, linear regression is a linear method (also known as dependent and independent variables). Simple linear regression is the situation where there is only one explanatory factor.  *Generalized Linear Regression:*  A generalized linear model is a versatile expansion of traditional linear regression in statistics. By allowing the linear model to be connected to the response variable via a link function and by allowing the size of each measurement's variance to be a function of its predicted value, the GLM generalizes linear regression.  *Decision tree Regression:*  A decision tree creates tree-like models for classification or regression. It incrementally develops an associated decision tree while segmenting a dataset into smaller and smaller sections. The outcome is a tree containing leaf nodes and decision nodes.  *Gradient Boost Regression:*  A machine learning method called gradient boosting is used, among other things, for classification and regression tasks. It provides a prediction model in the form of an ensemble of decision trees-like weak prediction models. The resulting technique, known as gradient-boosted trees, performs better than random forest when a decision tree is a weak learner.  **Modeling**  I tried using linear regression and generalized linear regression for modeling as the data is regression and not categorization. I started by thinking about a few characteristics that had extremely low r square values, therefore I added a few more features to raise the r square values. The improvement in both models is clearly shown in table 1 below.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Linear Regression**  **(6 features)** | **Generalized Linear Regression**  **(6 features)** | **Linear Regression**  **(All features)** | **Generalized Linear Regression**  **(All features)** | | RMSE (Root Mean Square Error) | 75987.175 | 75987.191 | 69775.453 | 69775.538 | | MSE (Mean Squared Error) | 5774050821.023 | 5774053260.508 | 4868613839.630 | 4868625728.770 | | MAE (Mean Absolute Error) | 55345.304 | 55345.360 | 50561.772 | 50561.807 | | R square | 0.564 | 0.564 | 0.648 | 0.648 |   **Table 1**  As shown in Table 2, I attempted to apply the decision tree regression model and the gradient boost regression model to improve the performance of the model. As we can see, gradient boost has the best performance of all the models, with a r square value of 0.72, which is better than the other models.   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Decision tree**  **regression**  **(All features)** | **Gradient boost regression**  **(All features)** | **Generalized Linear regression**  **(All features)** | **Linear Regression**  **(All features)** | | RMSE (Root Mean Square Error) | 71901.5 | 60336.7 | 67314.030 | 67313.985 | | MSE (Mean Squared Error) | 5.16983e+09 | 3.64051e+09 | 4531178569.615 | 4531172636.615 | | MAE (Mean Absolute Error) | 52508.4 | 42168.7 | 49268.667 | 49268.671 | | R square | 0.614517 | 0.728549 | 0.662 | 0.662 |   **Table 2**    Figure 11  We can view the test r2 score and train r2 score from figure 11 above. We can observe that none of these models were either overfitted or underfitted. We can also see that gradient boosting regression outperforms all other models in terms of training and testing results. I, therefore, choose to fine-tune this model to achieve higher performance. Using param grid and TrainValidationSplit, I further tuned this model and raised the r2 value to be a little bit more, approximately 0.73. This strategy is much more effective. The next section gives the summary of the analysis I have done so far |

|  |
| --- |
| **Conclusion and Future work**  According to the EDA (Exploratory Data Analysis), there are nine variables that are responsible for forecasting house value in California. In 1997, the average house value in California was $200,000. The value of houses rises as we get closer to the coast. The population density in coastal locations is high, as are housing costs. As a result, most high-income people live in coastal locations. There are double rooms available for the given population, as well as 1/3 of the bedrooms available for the given population. I made the decision to go deeper and use Spark to build machine learning models using gradient boosting regression, decision tree regression, generalized linear regression, and linear regression to estimate these home prices. I was able to predict house values using these ML models and improved my understanding of the Spark big data infrastructure. In comparison to other models for this dataset, the gradient boosting regression appeared to produce better results.  I would like to investigate ways to improve model results in the future and work on other models that can produce better results. To develop a more effective method for this dataset, I would like to go in further deeper using pipelines and neural networks. References are provided after this session |

**References**

Pace, Kelley, R., & Ronald Barry. *California Housing Dataset*. Dcc. Retrieved June 1, 2022, from <https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html>

*Build software better, together*. GitHub. Retrieved June 18, 2022, from <https://github.com/ageron/handson-ml2/blob/master/>

F. (2020, February 21). *Pyspark ML (Machine Learning) tutorial for beginners*. Kaggle. <https://www.kaggle.com/code/fatmakursun/pyspark-ml-tutorial-for-beginners/notebook>

MachineLearningPlus. (2022, March 8). *Pandas Dropna - How to drop missing values?* <https://www.machinelearningplus.com/pandas/pandas-dropna-how-to-drop-missing-values/>

L., & L. *Bigdata/6110\_project.ipynb at main · LakshmiSravya123/Bigdata*. GitHub. Retrieved June 18, 2022, from <https://github.com/LakshmiSravya123/Bigdata/blob/main/6110_project.ipynb>

Li, S. (2018, June 27). *Building A Linear Regression with PySpark and MLlib*. Medium. <https://towardsdatascience.com/building-a-linear-regression-with-pyspark-and-mllib-d065c3ba246a>

A., & A. *handson-ml2/02\_end\_to\_end\_machine\_learning\_project.ipynb at master · ageron/handson-ml2*. GitHub. Retrieved June 26, 2022, from <https://github.com/ageron/handson-ml2/blob/master/02_end_to_end_machine_learning_project.ipynb>

*Classification and regression - Spark 3.3.0 Documentation*. Spark. Retrieved June 26, 2022, from <https://spark.apache.org/docs/latest/ml-classification-regression.html#decision-tree-regression>

*Generalized Linear Model - an overview | ScienceDirect Topics*, <https://www.sciencedirect.com/topics/mathematics/generalized-linear-model>.

*pyspark.pandas.DataFrame.plot.line — PySpark 3.2.0 documentation*. Spark. Retrieved June 28, 2022, from <https://spark.apache.org/docs/3.2.0/api/python/reference/pyspark.pandas/api/pyspark.pandas.DataFrame.plot.line.html>

A. *spark/examples/src/main/python/ml at master · Apache/spark*. GitHub. Retrieved June 29, 2022, from <https://github.com/apache/spark/tree/master/examples/src/main/python/ml>

*Classification and regression - Spark 3.3.0 Documentation*. Spark. Retrieved June 29, 2022, from <https://spark.apache.org/docs/latest/ml-classification-regression.html#gradient-boosted-tree-regression>

*How to find optimal parameters using GridSearchCV for Regression in ML in python -* ProjectPro. Retrieved June 29, 2022, from <https://www.projectpro.io/recipes/find-optimal-parameters-using-gridsearchcv-for-regression>