Machine Learning Linear Regression Model Using Mutual Information

SSIE 500 Computational Tools Final Project Report

Mitchell Keomoungkhoune, Nevin Nedumthakady, Vanessa Serna Villa

May 19th, 2021

1 Introduction

As discussed in class, information theory is a useful computational tool that allows us to quantify, store and communicate digital information (1). Two types of information theory include entropy and mutual; Information entropy is the average level of ‘surprise’ or ‘uncertainty’ inherent in the variable’s possible outcomes whereas mutual information measures the correlation between variables.

Machine learning, a method of data analytics upon which our code runs, depends on the concept of mutual information. Mutual information is the measure of dependency between variables; it explores the relationship between multiple probabilistic systems and captures it through the use of information measurements (1). Mutual information is a value of zero if and only if the variables are independent; it is always symmetric and non-negative. It’s important to emphasize that mutual information measures cause and effect. It focuses on relationships and correlation that exists between variables, but does not identify or assign causation. There are several applications of mutual information in which one may want to maximize information and thus increase dependencies. Examples include search engine technology where mutual information is used between phrases and contexts; in medical imaging where a reference image may be compared against a patients to detect discrepancies and indicate a diagnosis; within genes and expression microarray data to better understand and predict gene networks; and lastly in telecommunications where the channel capacity is equal to mutual information and is maximized over input distributions. In reference to our project, mutual information has been used as a criterion for feature selection and feature transformation in machine learning. It is used to characterize both the relevance and redundancy of variables which can help users draw meaningful conclusions (1).

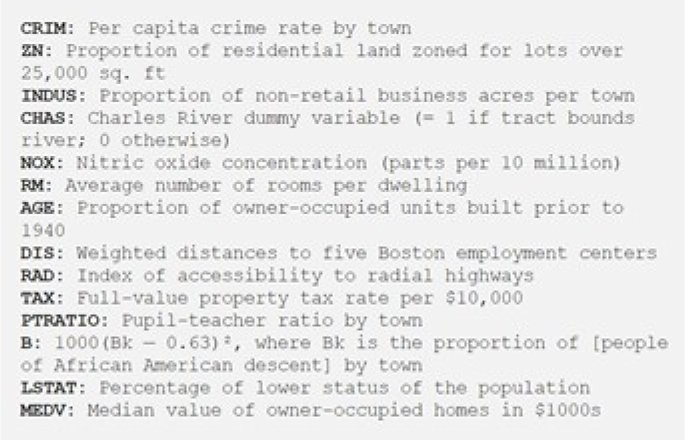
2 Dataset

The dataset we utilized in our Python code is the Boston Housing dataset. It contains information regarding houses in the Boston area. This data was originally a part of UCI Machine Learning

Repository and is no longer available. In order to access the data the scikit-learn library must be used. The data is quite small, as it only contains 506 samples and 13 features. It was first published in 1978 as a means to predict prices of houses using specific features.

First the required libraries are imported into Jupyter Notebook. Then the scikit-learn library is used to import the housing data. Next, we print the value of the boston_dataset and understand what it contains.

- Data: information for various houses
- Target: house price
- Feature_names: feature names
- DESCR: descriptoin of the dataset



CRIM: Per capita crime rate by town
ZN: Proportion of residential land zoned for lots over
25,000 sq. ft
INDUS: Proportion of non-retail business acres per town
CHAS: Charles River dummy variable (= 1 if tract bounds
river; 0 otherwise)
NOX: Nitric oxide concentration (parts per 10 million)
RM: Average number of rooms per dwelling
AGE: Proportion of owner-occupied units built prior to
1940
DIS: Weighted distances to five Boston employment centers
RAD: Index of accessibility to radial highways
TAX: Full-value property tax rate per $10,000
PTRATIO: Pupil-teacher ratio by town
B: $1000(Bk - 0.63)^2$, where Bk is the proportion of [people
of African American descent] by town
LSTAT: Percentage of lower status of the population
MEDV: Median value of owner-occupied homes in $1000s

Figure 1: Feature description of dataset

The prices of the house indicated by the variable MEDV is our target variable and the remaining are the feature variables based on which we will predict the value of a house. Now load the data into pandas dataframe. Print the first five rows of the data using head().

3 Feature Selection - Correlation and Mutual Information Statistics:

Before we confirm that the target value MEDV is missing from the data, we will perform feature selection for regression problems. Feature selection is the procedure of selecting a subset (some out of all available) of the input variables that are most relevant to the target variable (that we wish to predict). Target variable here refers to the variable that we wish to predict.

In order to find the correlation statistic we utilize the f_regression() function. This function can be used in a feature selection strategy, such as selecting the top k most relevant features (largest values) via the SelectKBest class.The plot below shows that feature 5 and 12 are more important than the other features. The y-axis represents the F-values that were estimated from the correlation values. The scikit-learn machine learning library provides an implementation of mutual information for feature selection with numeric input and output variables via the mutua_info_regression() function. The y-axis represents the estimated mutual information between each feature and the target variable. Compared to the correlation feature selection method we can clearly see many more features scored as being relevant.

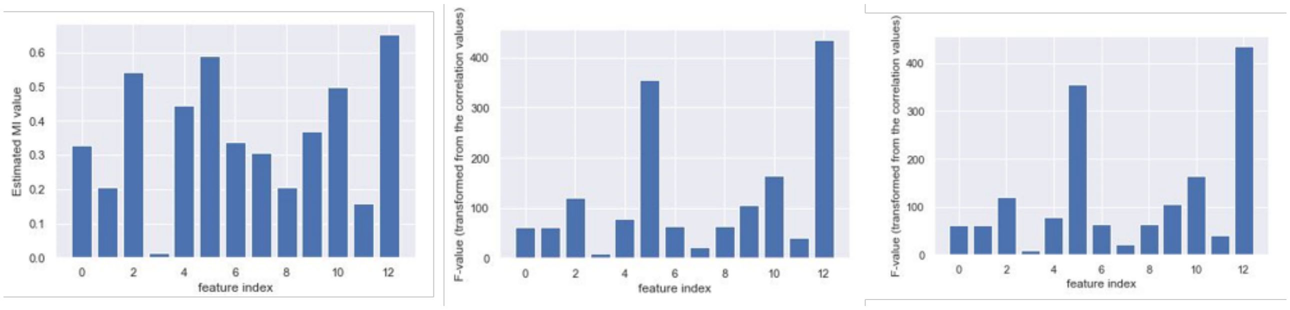
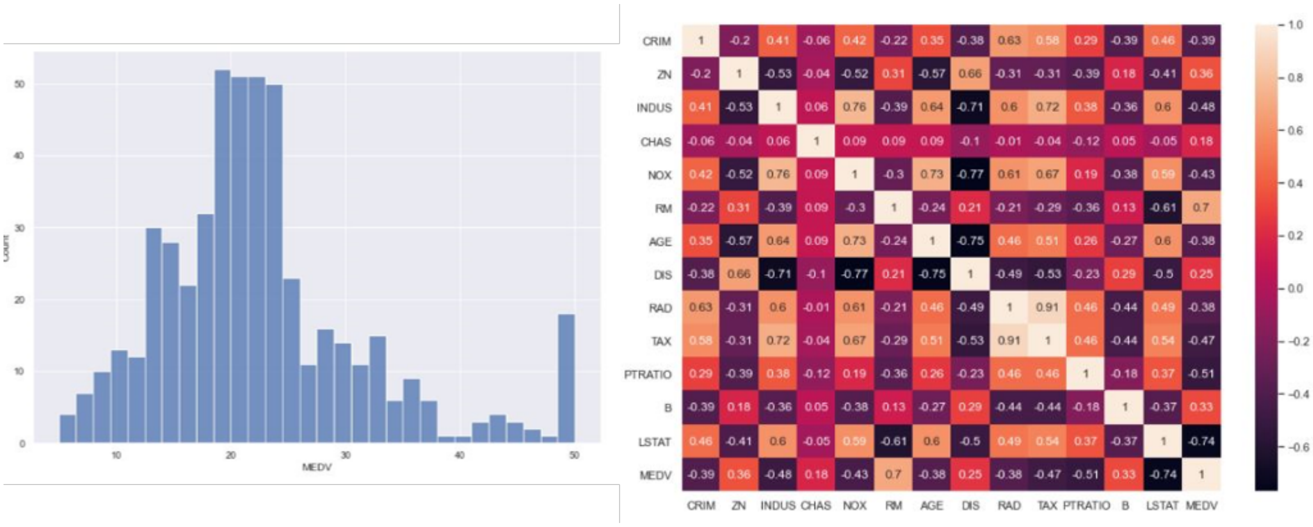


Figure 2: Mutal information visualization of features

4 Exploratory Data Analysis:

After loading the data, it is always a good idea to check for missing values in the data. In this dataset there are no missing values. Exploratory Data Analysis is a very important step before training the model. Visualizations are used to understand the relationship of the target variable with other features.

First plot the distribution of the target variable MEDV using this histplot function from the seaborn library. We see that the values of MEDV are distributed normally with few outliers. Next, we create a correlation matrix that measures the linear relationships between the variables. The correlation matrix can be formed by using the corr function from the pandas dataframe library. We will use the heatmap function from the seaborn library to plot the correlation matrix. The correlation coefficient ranges from -1 to 1. If the value is close to 1, it means that there is a strong positive correlation between the two variables. When it is close to -1, the variables have a strong negative correlation.



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
CRIM	1	-0.2	0.41	-0.06	0.42	-0.22	0.35	-0.38	0.63	0.58	0.29	-0.39	0.46	-0.39
ZN	-0.2	1	-0.53	-0.04	-0.52	0.31	-0.57	0.66	-0.31	-0.31	-0.39	0.18	-0.41	0.36
INDUS	0.41	-0.53	1	0.06	0.76	-0.39	0.64	-0.71	0.6	0.72	0.38	-0.36	0.6	-0.48
CHAS	-0.06	-0.04	0.06	1	0.09	0.09	0.09	-0.1	-0.01	-0.04	-0.12	0.05	-0.05	0.18
NOX	0.42	-0.52	0.76	0.09	1	-0.3	0.73	-0.77	0.61	0.67	0.19	-0.38	0.59	-0.43
RM	-0.22	0.31	-0.39	0.09	-0.3	1	-0.24	0.21	-0.21	-0.29	-0.36	0.13	-0.61	0.7
AGE	0.35	-0.57	0.64	0.09	0.73	-0.24	1	-0.75	0.46	0.51	0.26	-0.27	0.6	-0.38
DIS	-0.38	0.66	-0.71	-0.1	-0.77	0.21	-0.75	1	-0.49	-0.53	-0.23	0.29	-0.5	0.25
RAD	0.63	-0.31	0.6	-0.01	0.61	-0.21	0.46	-0.49	1	0.91	0.46	-0.44	0.49	-0.38
TAX	0.58	-0.31	0.72	-0.04	0.67	-0.29	0.51	-0.53	0.91	1	0.46	-0.44	0.54	-0.47
PTRATIO	0.29	-0.39	0.38	-0.12	0.19	-0.36	0.26	-0.23	0.46	0.46	1	-0.18	0.37	-0.51
B	-0.39	0.18	-0.36	0.05	-0.38	0.13	-0.27	0.29	-0.44	-0.44	-0.18	1	-0.37	0.33
LSTAT	0.46	-0.41	0.6	-0.05	0.59	-0.61	0.6	-0.5	0.49	0.54	0.37	-0.37	1	-0.74
MEDV	-0.39	0.36	-0.48	0.18	-0.43	0.7	-0.38	0.25	-0.38	-0.47	-0.51	0.33	-0.74	1

Figure 3: Uniform distribution of target value MEDV and correlation matrix of features

5 Observations

- Price increases as the value of RM increases linearly. Few outliers and data capped at 50

- Price decreases with increase in LSTAT. Does not look like it is following a linear line

- These features vary with MEDV, so using a scatter plot to visualize

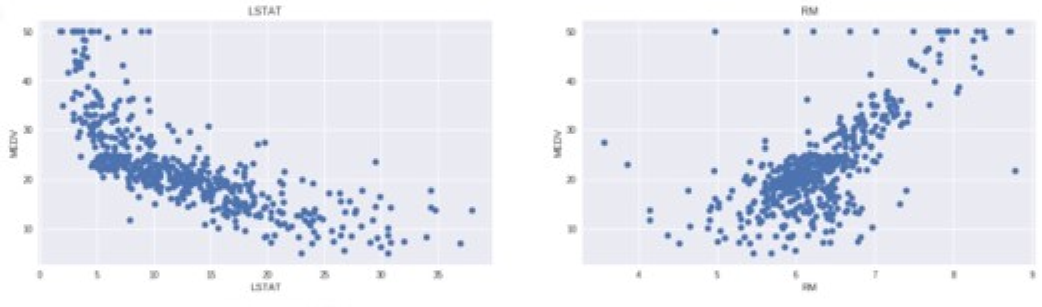
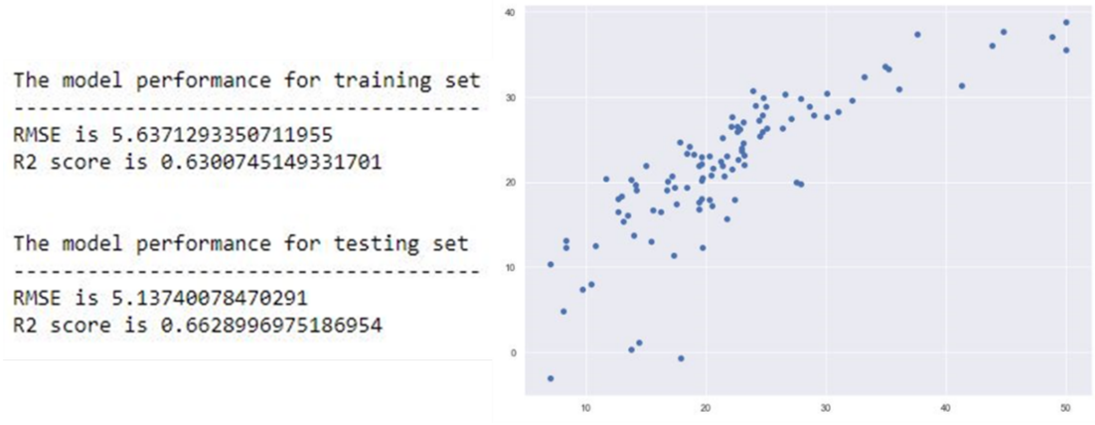


Figure 4: Relationship between MEDV with RM and LSTAT

6 Manipulation of Data and Model

Using np.c we concatenate the LSTAT and RM columns. Next, we split the data into training and testing sets. We train the model with 80% of the samples and test with the remaining 20%. We do this to assess the model's performance on unseen data. To split the data we use the train_test_split function. Finally, we print the sizes of our training and test set to verify if the splitting has occurred properly. LinearRegression is used to train the model on both the training and test sets. Below the model using RMSE and R2-score is evaluated and then the y_test vs y-pred is plotted.



```
The model performance for training set
--------------------------------------
RMSE is 5.6371293350711955
R2 score is 0.6300745149331701

The model performance for testing set
--------------------------------------
RMSE is 5.13740078470291
R2 score is 0.6628996975186954
```

Figure 5: Model peformance measures and scatter plot of y_test and y_pred

7 Conclusion

In conclusion we used concepts of information theory and mutual information to generate a linear regression model to predict median home values based on information from the boston housing dataset. Using mutual information we implemented a feature selection method to reduce the dimension of the dataset while still retaining core elements of the data for linear regression. Furthermore, we used exploratory data analysis to summarize the main charaterstics of the boston housing dataset and from this approach determined that RM (Average number of rooms per dwelling) and LSTAT (Percentage of lower status of the population) possessed the highest correlation or mutual information to MEDV. By using training and testing machine learning methods to build the linear regression model we were able to produce a model that computed a RSME of 5.64 and a R2-score of 0.63 for the training set and a RSME of 5.14 and a R2-score of 0.66 for the testing set.

References

1. SSIE 500 2021 Spring Course Materials