Regression Task for House Price Prediction

Nehal Malap | MSc. Data Analytics | 21237207

Introduction

The task given, is to develop a regression model using any two of the regression algorithms (one of Multiple Linear Regression /K Nearest Neighbour Algorithm) on the given test and training data sets using any machine learning packages. The final aim of the regression algorithm is to analyze a house price prediction dataset for finding out the price of the house on different parameters. This prediction consists of houses rent ranging from 580 to 3700 euro per month. This report describes the methodology followed in achieving the best performance of the regression model.

Selection of Machine Learning Package and Regression Algorithm

There are a total of 8 open-source machine learning packages are available which are Tensor Flow, Keras, Scikit-learn, Microsoft Cognitive Toolkit, Theano, Caffe, Torch, and Accordnet. The purpose of every package is different. TensorFlow and PyTorch library are majorly used for deep learning and Neural Networks. Keras library supports both convolutional and recurrent networks. Scikit learn focuses on data mining and analysis.

For the given regression task, I have used a Scikit-learn package over another machine learning packages. The entire package is heavily written in python, and it becomes very easy to understand and implement the various algorithms. The scikit-learn library is very versatile and helps to split the entire dataset into train and test datasets and provides different machine learning algorithms to solve different problems like Classification, clustering, Regression also it provides different algorithms to solve different selection Dimensionality Reduction and Pre-processing. It provides a wide selection of supervised and unsupervised algorithms. This package is composed of three main libraries which are NumPy(mainly used for Scientific Computation), Pandas (to process Tabular Data), Geo, and SciPy(for sparse matrices). Within a few lines of code, it is easy to accomplish classification and many other tasks. I have used the scikit package in NLP and in previous ML assignments and observed the improvement in the implementation of various machine learning models. Efficiency and versatility of use make scikit-learn one of the prime choices of academic and industrial organizations for performing various operations.

Data Pre-processing and Data Cleaning

The following pre-processing steps are followed before developing the data models.

- The given dataset text file(galway_rentals.txt) is converted to .csv files using the text import wizard on Microsoft Excel.
- As data is in the CSV file, will read the CSV file using pandas read_csv function and check the first 5 rows of the data frame using head(). The dataset has 395 records and different 12 columns(attributes)
- To analyze the data correctly and to understand the nature of the data has performed a few analysis steps such as checking the info of the data frame by using the .info function. In the output of the function, it can be seen that distance_eyresquare, distance_salthill, and two more attribute types are float and the other 3 attribute type is int, and the remaining 5 attributes are of the object type. Describe function is used to understand statistical data like percentile, mean, standard deviation, minimum and maximum value of each attribute.
- Explored the dataset using plots, based on the plot we can determine that distribution is nonnormalized. The histogram is in bimodal shape. To know the correlation between the data used heat map plotting. Each column has a 100% correlation with itself. There is a high correlation between num_of_bathrooms and price_per_month attribute that is 55%.

Figure : Distribution of parameters

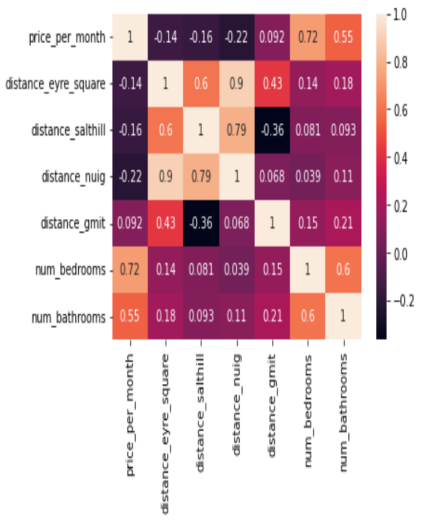

Figure : Correlation matrix

- Identified the type of attribute value as there are 7 attributes that have continuous value and 5 that have categorical. For eg. column type and heating contain a type of property and type of heating system values respectively which is not in natural ordering. Column type values studio, apartment, house, townhouse, and heating have central, electric, NA values hence to convert these categorical values into indicator values used get_dummies function. For the rest 3 columns such as ber balcony and floor, the possible values of this attribute have natural ordering categorical values hence used Ordinal Encoder to map each unique variable to an integer value. For eg. ber column has the values like a, b1, b2, b3, c1, c2, c3, d1, d2, e1, e2, f, g, exempt (ordered from most desirable to least desirable) using ordinal encoder it converts into (1.0,2.0.,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0,13.0).
- Split the dataset into a training set and testing set using sklearn train_test_split(). The training set is used for training the model and the testing set for testing the model. Split the data into 40% testing data and 60% of the training set. The target attribute(price_per_month) and the feature attributes were separated for both the training and test datasets.
- The feature attributes in the training and test datasets are standardized using the StandardScaler() function to normalize the value

Algorithms Description and parameter settings:

A brief description of the Linear Regression and K Nearest Neighbour algorithms are as follows:

- **K-Nearest Neighbours Algorithm:**

1. KNN algorithm is a supervised learning algorithm that is used for both classification and regression.
2. Uses the similarity feature to predict the group that the new point will fall into.
3. In KNN, K is the number of nearest neighbors. The number of neighbors is the main deciding factor. It finds the distance between the query points and all the dataset points. It looks at K closet query points in the dataset points whatever is the majority class in the group of K data it is the predicted class of algorithm. The k-value is chosen by analyzing the R2 score for various k-values in a specific range.[1]
4. The distance can be calculated using 'Euclidean distance', 'Manhattan distance', 'Hamming Distance', and 'Minkowski Distance'. The euclidian distance used by default in the scikit-learn package and it is can be calculated as –

   Distance between (x1,y1) and (x2,y2) = √(x1-x2)^2 + (x2- y2)^2
5. In the regression, the task used the Euclidean distance method and predicted the house

rent based on the other attributes.

6. Pre-processed the train data set and imported KNeighborsClassifier from sklearn neighbors. Developed a KNN model with .fit() method. Passed trained dataset at x-axis and Price_per_month column at the y-axis
7. Predict the result of the model, pass preprocessed test data, and label data.
8. From the analysis of the k –values, it can be identified that the model predicts with the good R2score when k=5. Hence, this value is used for developing the model.
9. Calculated Mean Squared error = 134263.81, Root Mean Squared Deviation = 366.42, and R2 score = 53.60

- **Multiple Linear Regression**

1. Linear Regression is a Supervised Machine Learning Model for finding the relationship between independent variables and dependent variables.
2. Linear regression performs the task to predict the response (dependent) variable value (y) based on a given (independent) explanatory variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output).
3. In Multiple Linear Regression, there are more than one independent variables for the model to find the relationship.
4. Equation of Multiple Linear Regression, where bo is the intercept, b1,b2,b3,b4…, bn are coefficients or slopes of the independent variables x1,x2,x3,x4…,xn and y is the dependent variable.

$$y = b_o + b_1x_1 + b_2x_2 + b_3x_3 \ldots . + b_nx_n$$

5. The main aim of the Linear Regression model is to find the best fit linear line and the optimal values of intercept and coefficients such that the error is minimized.
   Error is the difference between the actual value and Predicted value and the goal is to reduce this difference.
6. In the given regression task created the linear regression model using scikit's LinearRegression() package. There are a few optional parameters like ***fit_intercept*** is a Boolean that decides whether to calculate the intercept b_0 (True) or consider it equal to zero (False). ***normalize*** is a Boolean that decides whether to normalize the input variables (True) or not (False). ***copy_X*** is a Boolean (True by default) that decides whether to copy (True) or overwrite the input variables (False). ***n_jobs*** is an integer or None (default) and represents the number of jobs used in parallel computation. None usually means one job and -1 to use all processors.[2]
7. Created the model with fit_intercept is True, normalize is false, copy_X is True, n _jobs is none.
8. After creating the model, used a fit() method that fits the created model. The fit method calculates the optimal value of b0 and b1 using the existing input and output (x and y) as the arguments.
9. The model was fitted with training data and training labels and determined the outcome. To get the corresponding predicted response use predict() method. The predict function output the values by multiplying each column of the input with the appropriate weight, summing the results, and adding the intercept to the sum. The R2 score = 61.08 ,RMSE = 328.58 and MSE = 107968.424 are calculated to evaluate the performance of the model.

Overfitting and Underfitting

Overfitting occurs when the model performs well on training data but fails to generalize well to new, unseen data points. It happens due to noisy data or model learned to predict specific inputs rather than the predictive parameters that could help it make correct predictions. When the model is very complex then there is a high chance that the model will be overfitted.[5]

Underfitting occurs when the model has poor performance even on the training dataset. It happens

because the model is not suitable for the problem you are trying to solve. this means that the model is less complex than required in order to learn those parameters that can be proven to be predictive. This issue occurs while splitting the dataset into training and test data. The training data is used to train the model and test data is used to evaluate whether the model can generalize well to new, unseen data. There is common practice to assign ⅔ data points to the training data and the remaining ⅓ data points to the test data. While developing a model the training data is used to train the model and then test data is used to evaluate. If the training accuracy is greater than testing data accuracy then it is an indicator of the overfitted model. There are some other situations like when both training and testing data don't contain patterns that do not exist in real-world data then the model would still have poor performance. This can be avoided by creating an extra set known validation set. It helps to select a model which performs better among the others. The developed model is neither overfitted nor under fitted.

Evaluation

Evaluated the model by using below evaluation metrics:

1. **Mean Squared Error (MSE)**: Another Common metric for evaluation is Mean squared error which is the mean of the squared difference of actual vs predicted values
2. **Root Mean Squared Error (RMSE):** It is the root of MSE i.e Root of the mean difference of Actual and Predicted values. RMSE penalizes the large errors whereas MSE doesn't.
3. **R squared or Coefficient of Determination**: It can be defined as a Ratio of variation to the Total Variation. The value of R squared lies between 0 to 1, the value closer to 1 the better the model.

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \qquad \text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \qquad MSE = \frac{1}{n}\Sigma\underbrace{\left(y - \widehat{y}\right)}_{\text{The square of the difference between actual and predicted}}^2$$

4. The R2 score of both the models are not having a huge difference. The R2 score is similar with the difference of 8% that is the R2 score of the linear model is – 61% and the R2 score of the KNN model is 53% with K = 5.
5. KNN and Linear Regression algorithms both are easy to interpret prediction.
6. The linear Regression algorithm finds the linear relationship between the dependent and independent variables.
7. Whereas in KNN, it finds the nearest neighbor of query data point and assigns the class/labels based on the nearest neighbor's value.
8. The accuracy of both the models is 53% similar, as data preprocessing steps performed on both the models and train dataset are the same. In KNN the accuracy varies as the k values changes.
9. KNN is sensitive, due to noise and irrelevant data the accuracy gets impacted.

References

[1]https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn

[2]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[3]https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-your-first-machine-learning-model-linear-regression/

[4]https://towardsdatascience.com/how-to-split-a-dataset-into-training-and-testing-sets-b146b1649830#:~:text=The%20simplest%20way%20to%20split,the%20performance%20of%20our%20model.