Subject Code: **CSE5ML**

Student Name: **Ashish Rawat**
Student Number: **20343966**

REGRESSION ANALYSIS ON HOUSING DATASET

INTRODUCTION

Regression analysis is a form of predictive modeling technique that investigates the relationship between a dependent (target) and the independent variable (s) (predictor). The task is formulated as to accurately predict median value of Boston houses using three different Regression model, and the goal is to find the most optimal and efficient trained model that provides the best relationship between the dataset predictors and label with high R-squared values and less Root Mean Square Error (RMSE) using different parameter settings.

DATASET

The provided Boston dataset consists of 13 predictors variables and one feature variable (House price) of continuous values. The data is divided into two sets; train data (to train our models) and test data (to evaluate and test against trained models). The data operations performed importantly using library NumPy, Pandas, Sklearn, and Seaborn visualization. The dataset contains 506 houses data with 13 properties which estimate the median price of property, where 20% of data is used for testing and 80% for training the model to accurately predict the house prices.

This dataset contains the following columns:

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 - nitrogen oxides concentration (parts per 10 million).
Rm - average number of rooms per dwelling.
Age - proportion of owner-occupied units built prior to 1940.
Dis - weighted mean of distances to five Boston employment centres.
Rad - index of accessibility to radial highways.
Tax - full-value property-tax rate per \$10,000.
Ptratio - pupil-teacher ratio by town.
Black - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
Lstat - lower status of the population (percent).
Medv - median value of owner-occupied homes in \$1000s.

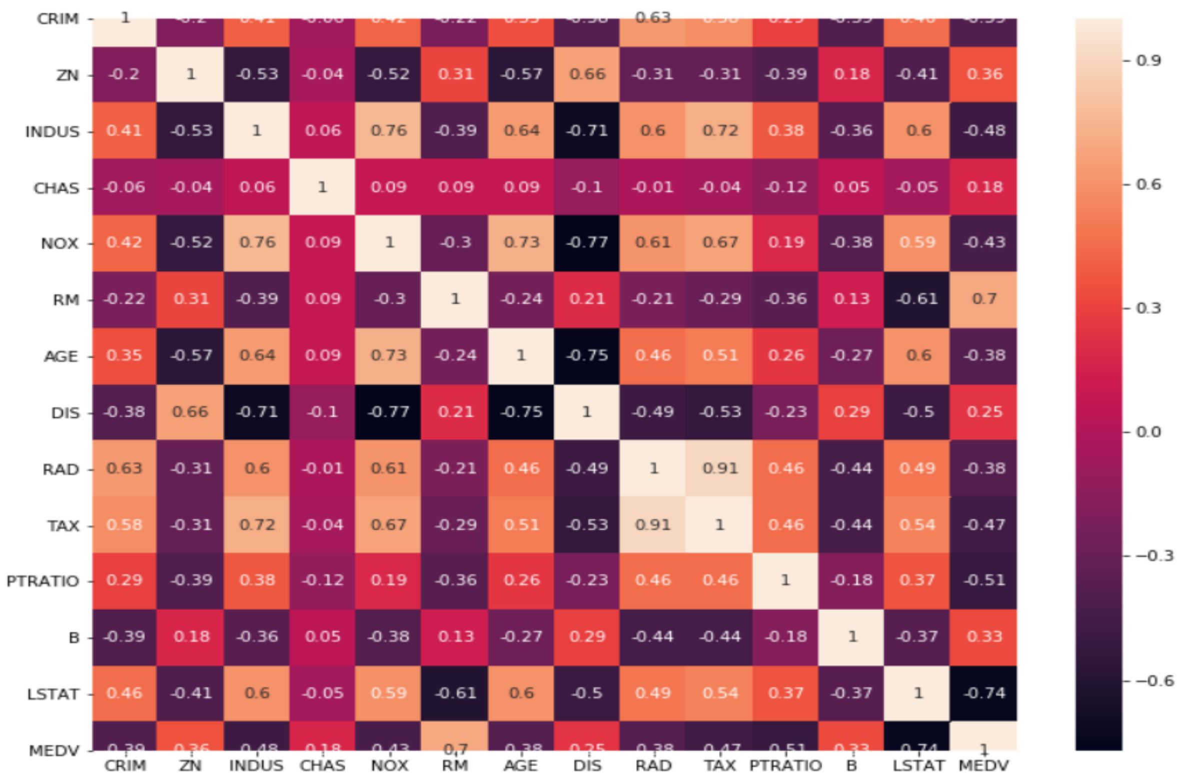


Figure 1: Heatmap of correlation between variables

MODELLING

Three end to end regression models are developed to predict Boston house prices: **Support Vector Machine (SVM), LinearRegression and K-Nearest Neighbors (KNN)**.

1) LinearRegression

Linear regression is widely used regression techniques also called the method of ordinary least squares that calculate the estimators of the regression coefficients or simply the predicted weights, denoted with b_0, b_1, …, b_r. This function should capture the dependencies between the inputs and output sufficiently well. To get the best weights, you usually minimize the sum of squared residuals (SSR) for all observations.

```
model = LinearRegression(normalize=True)
performance_metrices(model, X_test, Y_test, X_train, Y_train)
```

```
The model performance for training set
--------------------------------------
RMSE is 4.741000992236517
R2 score is 0.7383393920590519


The model performance for testing set
--------------------------------------
RMSE is 4.568292042303214
R2 score is 0.7334492147453069
```

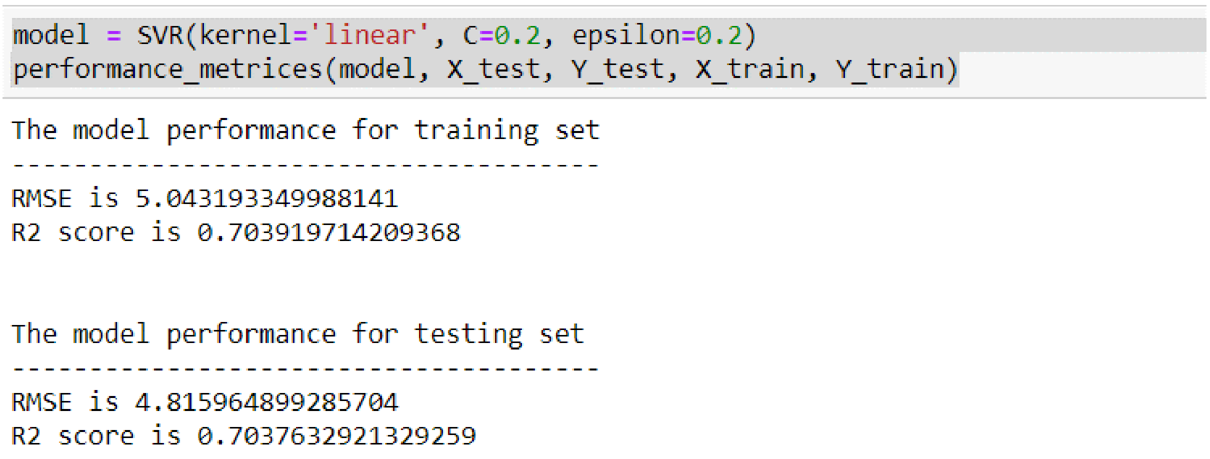
Steps taken to fine tune the model:

Step 1: Set the **normalize=True** parameter to normalize the dataset before applying regression to normally distribute it.
Step 2: Keep the **fit_intercept=True** parameter true to keep data is expected to be centered
Step 3: Kept the **n_jobs=None** since it is the simple dataset and job does not require more parallel jobs for computation

Result –
When fitting the data into the model with the above parameter and it produced an R-squared value of 0.73 on Training and 0.73 on the testing set as well while keeping low of 4.73 and 4.56 the Root mean squared error on both training and testing respectively.

2) Support Vector Regression (SVR)

Support Vector Regression used as a regression method, managing all the main features that characterize the algorithm (maximal margin Effective in high dimensional spaces). SVR gives us the flexibility to define how much error is acceptable in our model and will find a fitting line (or hyperplane in higher dimensions) to fit the data. The objective function of SVR is to reduce number of coefficients vector not the squared error. It has option of choosing different Kernel functions and still effective in cases where number of dimensions is greater than the number of samples while keeping memory efficiency.



```
model = SVR(kernel='linear', C=0.2, epsilon=0.2)
performance_metrices(model, X_test, Y_test, X_train, Y_train)
```

```
The model performance for training set
--------------------------------------
RMSE is 5.043193349988141
R2 score is 0.703919714209368


The model performance for testing set
--------------------------------------
RMSE is 4.815964899285704
R2 score is 0.7037632921329259
```

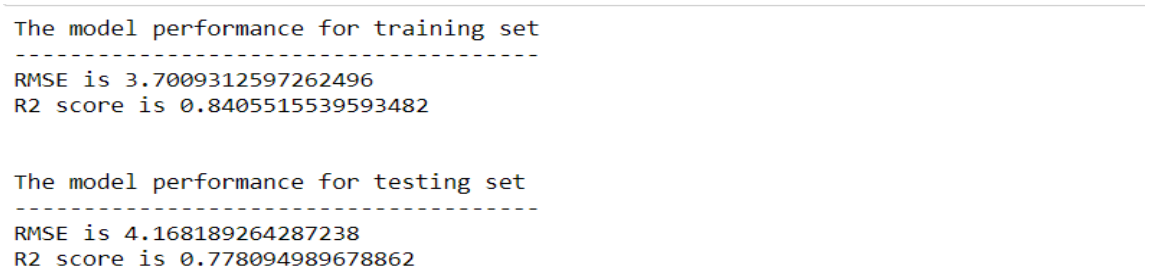
Steps taken to fine tune the model:

Step 1: Set the kernel to 'linear' instead of default 'rbf'
Step 2: Keep the *gamma* parameter value set to default 'scale' since we decided to go with kernel to linear and those parameters does not work with it
Step 3: Set Regularization parameter **C** to low value of 0.2 to prevent the overfitting
Step 4: Set epsilon=0.2 to reduce training loss function

Result –
When fitting the data into the model with the above parameter and it produced an R-squared value of 0.70 on Training and 0.70 on the testing set as well while keeping low of 5.043 and 4.81 the Root mean squared error on both training and testing respectively.

3) K-Nearest Neighbors

K-Nearest Neighbors algorithm for Regression (KNeighborsRegressor) is simplest yet powerful regression model that has proven to be incredibly effective and fast at certain tasks. It is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve regression problems with achieving good accuracy score. KNN Regressor accepts different parameters to tune the model to achieve high accuracy based on given data and the target is predicted with the nearest neighbors in the training set.



```
The model performance for training set
--------------------------------------
RMSE is 3.7009312597262496
R2 score is 0.8405515539593482


The model performance for testing set
--------------------------------------
RMSE is 4.168189264287238
R2 score is 0.778094989678862
```

<u>Steps taken to fine tune the model:</u>

1. Feature selection by using absolute value of highly correlated fields (>0.5) to be used.

2. Set n_neighbors=6 after testing with different number neighbours parameter.

Result –
When fitting the data into the model with the above parameter and it produced an R-squared value of 0.84 on Training and 0.77 on the testing set as well while keeping low of 3.7 and 4.168 the Root mean squared error on both training and testing respectively.
This model has highest chance of overfitting as compare to other models we build on comparisons since testing and training R-squared and RMSE values quite varies.

<u>FINAL RESULTS</u>

Although all models dispense good R-squared of over 70% and RMSE (Root mean squared error) rate less than 10% when fitted the training and test data in the model.

The analysis and evaluation of regression models suggests -

LinearRegression (**Model 1**) **is the best model** because of the evaluation report (RMSE: 4.56%, R-Squared: 73%) and the results shows training and testing data as R-squared value is quite similar which depicts the chances of

less overfitting or underfitting issue while keeping manageable computation cost
and model evaluation time.

K-nearest Regressor (RMSE: 4.168**%, R-squared: 0.77%),** despite its good accuracy and model evaluation speed, because evaluations report reveals the inconsistency of loss and accuracy rate which might result in overfitting in new data.

Support Vector Regression have similar accuracy as LinearRegression but computationally expensive and slow model building, and processing time as compare to other two models.

REFERENCES

Lib.stat.cmu.edu. 2020. [online] Available at: <http://lib.stat.cmu.edu/datasets/boston> [Accessed 4 June 2020].

Medium. 2020. *An Introduction To Support Vector Regression (SVR)*. [online] Available at: <https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2> [Accessed 4 June 2020].

Saedsayad.com. 2020. *KNN Regression*. [online] Available at: <https://www.saedsayad.com/k_nearest_neighbors_reg.htm> [Accessed 4 June 2020].

Medium. 2020. *Machine Learning Basics With The K-Nearest Neighbors Algorithm*. [online] Available at: <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> [Accessed 4 June 2020].