Ain Shams University
Faculty of Computer & Information Sciences
Computer Science Department

Machine Learning Project Report

Selected Project Idea:

"House Price Prediction"

Under Supervision of:

Dr. Dina Khattab

Team Members:

1st Team Member Name:

Kareem Sherif Fathy

1st Team Member ID:

2018170283

2nd Team Member Name:

Kareem Saeed Ragab

2nd Team Member ID:

2018170282

3rd Team Member Name:

Abanoub Asaad Azab

3rd Team Member ID:

2018170001

4th Team Member Name:

Nada El Sayed Anies

4th Team Member ID:

2018170430

5th Team Member Name:

Nada Mohamed Abdelhamed

5th Team Member ID:

2018170434

Milestone #1: "Preprocessing and Regression"

1- Preprocessing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

Before we deep into our preprocessing techniques, there are a few things to clarify:

- We are passing *two separate objects* to the preprocessing methods one is representing the **training data** and the other represents **the unseen data** for testing. All the preprocessing techniques are mainly applied to the training data while the testing data are just keeping up with the training data.
- We are saving all the models used in preprocessing using the ***'pickle'*** library, for later use.

1.1 Dropping useless Features

Starting the Pre-processing by neglecting the unnecessary columns either which don't affect the target value (such as case index) or contain too many missing values.

- Dropping *ID* column as it only refers to row index.
- Eliminating those which have more than **80%** of total data missing or showing no values. For this dataset, we are Dropping *PoolQC*, *MiscFeature*, and *Fence*. Their Missing values percentages were very high, 99%, 96%, and 81% respectively. Which they cover almost the entire column, eliminating these features won't cause any significant loss of values or even affects the prediction process. The right evaluation was also considered as it is important to be aware of dealing with such missing values because there may be a chance of losing important data.

1.2 Solving missing values

Imputing missing data is very important in the Feature Engineering process. There are many techniques to impute the missing values in such an accurate manner. What attracted us were the ***Iterative Imputer*** *(MICE)* and the ***mean*** replacement techniques.

The Iterative Imputer is such an effective algorithm which refers to a process where each feature is modeled as a function of the other features, e.g. a regression problem where missing values are predicted. Each feature is imputed sequentially, one after the other, allowing prior imputed values to be used as part of a model in predicting subsequent features and it uses the mean as the **initial_strategy.** In our trials it was found that the Iterative Imputation is not stable with changing the random seed. But it was an experience that has to be tested.

- Numeric Data is imputed using the mean strategy.
- Categorical Data is imputed using the most-frequented value.

1.3 Label encoding

House Pricing Dataset contains **35 categorical features** which we can't ignore (drop) or even let them as they are. We started the encoding using ***LabelEncoder*** and ***OneHotEncoder***.

OneHotEncoder converts each categorical value into a new categorical column and assigns a binary value of 1 or 0 to those columns. But after applying that and then observing the correlation (1.4 Feature Selection) was found that ***all the correlated features are non-categorical data***, then for preserving memory there was no need to use ***OneHotEncoder***. So with the help of *sklearn,* we used *LabelEncoder* instead of *OneHotEncoder* to encode the textual data in a sensible manner.

1.4 Feature Selection

By using the Pearson Coefficient of Correlation we are getting the top 50% correlation features with the target (***SalePrice***).

There are features (from the top features) for Example *1stFlrSF* and *TotalBsmtSF*, both are strongly correlated with each other and also have a strong correlation with the target. So for preserving the memory and runtime, there is no need to consider both features, we consider selecting from each strongly correlated features the most correlated one with the target (*SalePrice*) and dropping the rest.

1.5 Feature Scaling

Feature scaling is a method used to normalize the range of independent variables or features of data. It makes the flow of ***gradient descent*** smooth and helps algorithms quickly reach the minima of the cost function.

We used **Standardization** which is a very effective technique that re-scales a feature value so that it has distribution with *0 mean* value and *variance* equals *1*.

Although Decision trees and ensemble methods do not require feature scaling to be performed as they are not sensitive to the variance in the data, we kept it as a general process to lower the headache that is going to happen in saving the data as a side effect.

2- Regression Techniques Used:

a. **Gradient Boosting Regressor.**
b. **Random Forest Regressor.**
c. **Ridge Regression.**

3- Differences Between Each Model:

3.1 Gradient Boosting Regressor

Builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage, a regression tree is fit on the negative gradient of the given loss function.

- Accuracy: **91.97%**
- Mean Square Error: **712024004.23**
- Training Time: **0.18s**

3.2 Random Forest Regressor

An ensemble learning algorithm based on decision tree learners. The estimator fits multiple decision trees on randomly extracted subsets from the dataset and averages their prediction.

- Accuracy: **89.34%**
- Mean Square Error: **945551389.10**
- Training Time: **0.5s**

3.3 Ridge Regression

A model tuning method used to analyze any data that suffers from multicollinearity. This method performs L2 regularization. When the issue of multicollinearity occurs, least-squares are unbiased, and variances are large, this results in predicted values being far away from the actual values.

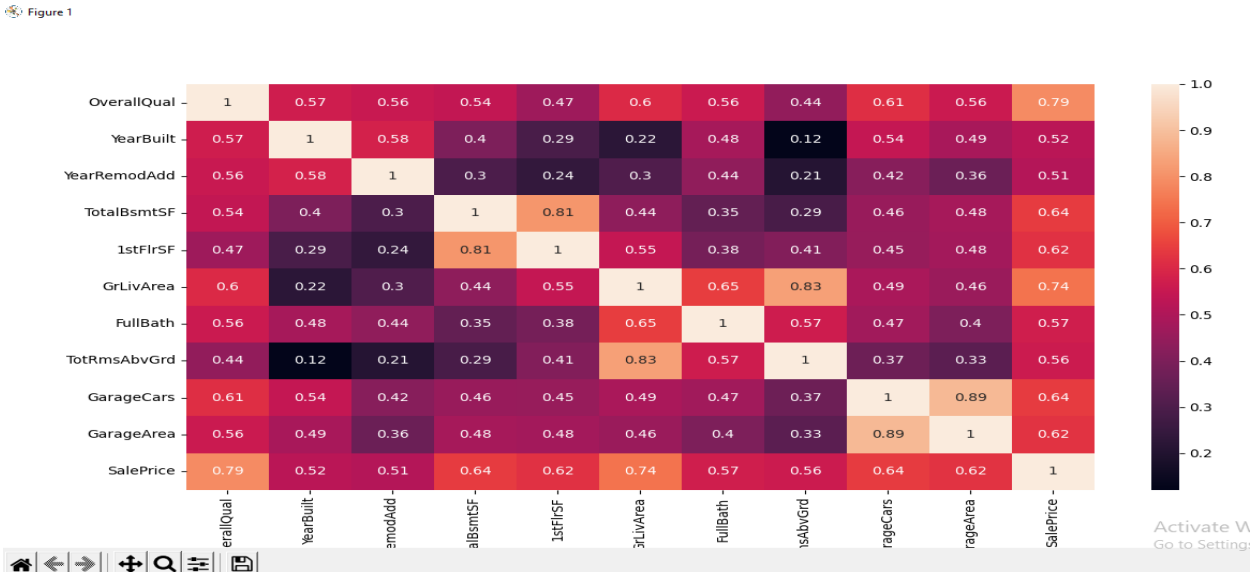
- Accuracy: **80.12%**
- Mean Square Error: **1762930579.35**
- Training Time: **0.08s**

4- Training & Testing Sizes:

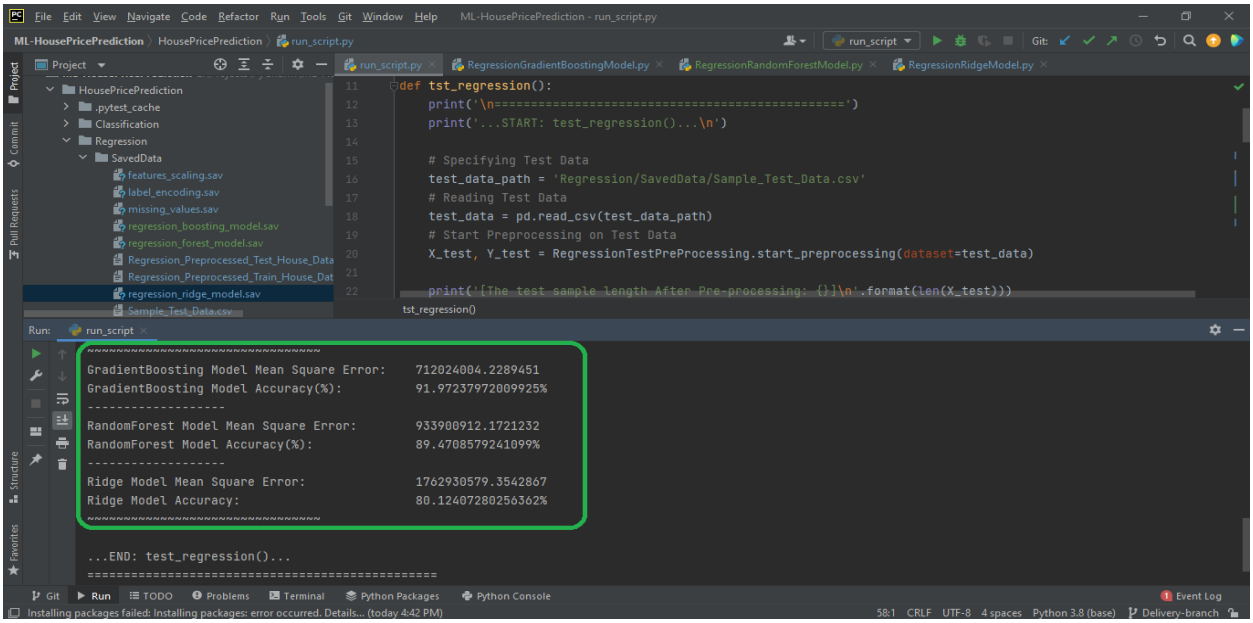
- **Training set:** is 80% of the total data size.
- **Testing set:** is 20% of the total data size.
- **Validation set:** no validation set was used.

5- Work Screenshots:

5.1 Dataset Features Correlation:



5.2 Running Script of All Regression Models Output:



6- Conclusion:

In most practices, the best metrics to optimize have been obtained by the decision trees followed by random forest and extreme gradient boosting. But in the House Price Prediction, It differs. So it is always related to the dataset and how their features relate with each other, and there is no static way to obtain the best metric.

This dataset example demonstrates **Gradient Boosting** to produce a predictive model from an ensemble of weak predictive models. Gradient boosting can be used for **regression and classification problems**.

Random Forest is a very powerful model both for **regression and classification.** It can give its own interpretation of feature importance as well, which can be plotted and used for selecting the most informative set of features according, for example, to a Recursive Feature Elimination procedure. Properly used, feature importance can give us very good and easy-to-understand deliverables (the bar plot) and efficient optimization (feature selection). That's why I think that feature importance is a necessary part of every machine learning project.

Ridge Regression alpha value, which is a **hyperparameter** of Ridge, which means that they are not automatically learned by the model instead they have to be set manually. The optimum alpha for Ridge Regularization can be found by applying *GridSearchCV*.

Milestone # 2: "Classification and Hyperparameter Tuning"

1- Classification Techniques Used:

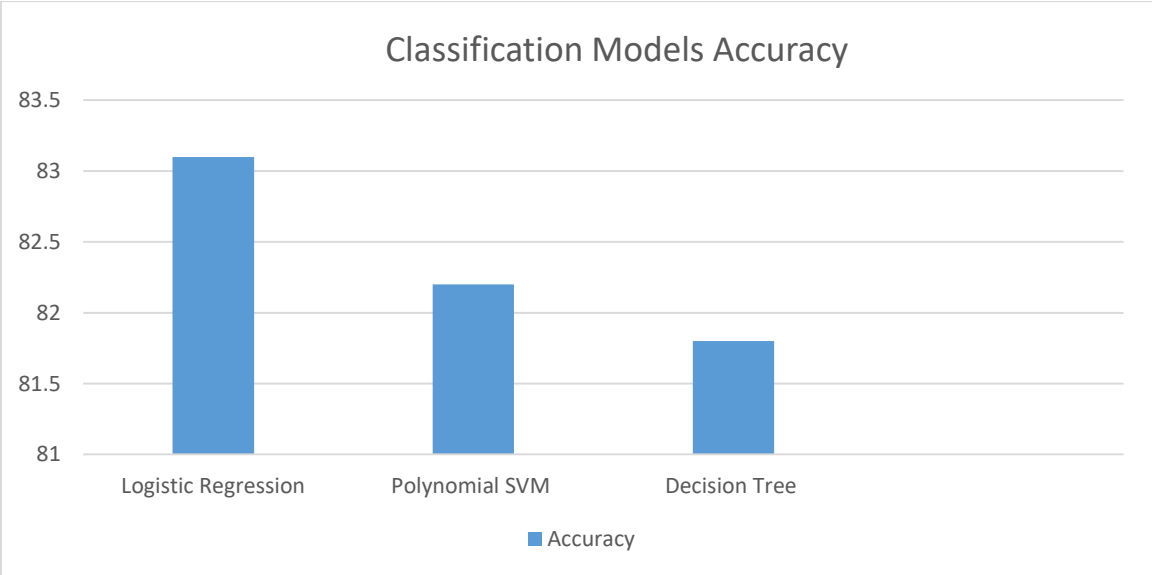
a. **Polynomial SVM**

b. **Decision Tree**

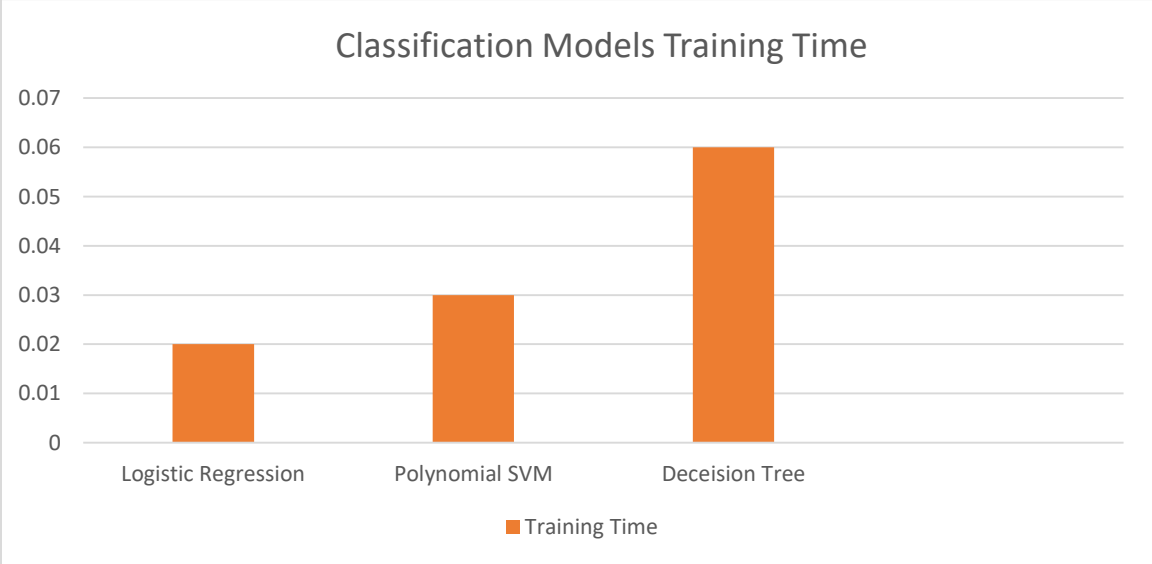
c. **Logistic Regression**

2- Three Classification Models Summary:

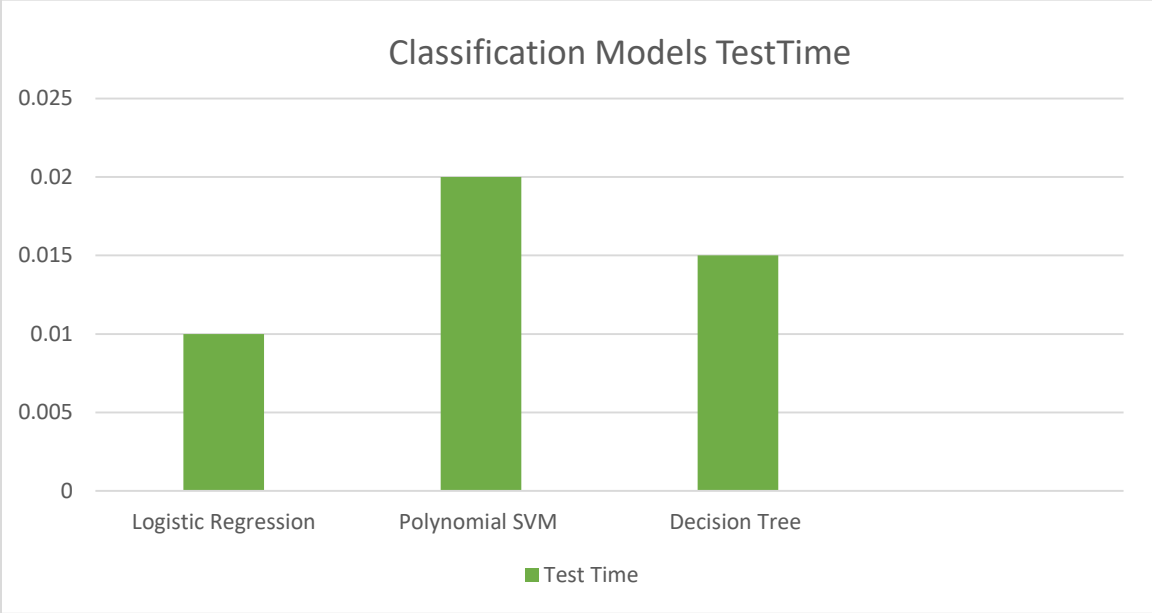
2.1 Accuracy



2.2 Training Time



2.3 Testing Time



3- Feature Selection:

In this phase, the feature selection differs from the previous phase, the top 50% correlation features with the *PriceRate* (target) were *OverallQual*, *YrarBuilt*, *FullBath* and *GarageCars* features which are having a high correlation with *PriceRate* (target).

4- Hyperparameter Tuning:

In machine learning, hyperparameter optimization or tuning is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

4.1 According to Polynomial SVM Model.

There are two hyperparameters:

[Regularization parameter **(C)**, Polynomial Degree (**D**egree)]

C	Degree	Accuracy
0.1	3	72.2%
10	3	81.4%
50	3	83.1%

C	Degree	Accuracy
0.1	2	81.8%
0.1	3	72.2%
0.1	4	83.1%

4.2 According to Decision Tree Model.

There are two hyperparameters:

[Max Tree Depth (**max_depth**), Max Leaf Nodes (**max_leaf_nodes**)]

max_depth	max_leaf_nodes	Accuracy
3	6	80.9%
4	6	81.8%
5	6	81.9%

max_depth	max_leaf_nodes	Accuracy
3	2	78.1%
3	3	78.9%
3	5	80.9%

4.3 According to Logistic Regression Model.

There are two hyperparameters:

[Regularization parameter (**C**), number of iterations (**epochs**)]

C	epochs	Accuracy
0.1	1000	73.5%
50	1000	83.8%

C	epochs	Accuracy
0.5	1000	76.1%
0.5	5000	80.5%

5- Conclusion:

In most practices, the best model that gives a high accuracy is a logistic regression classifier that works with the sigmoid function.

The correlation in this phase between the features and PriceRate (target) is low because the target has value in (cheap, moderate, expensive) So, the three values doesn't have a strong correlation with other features that have various values. This problem is solved when we begin to scale feature values between 0 and 2 then make the correlation. In this case, the Accuracy was updated to high and correlation became strong.