Contents

Prediction of Cumulative Grade Point Average

Dataset:

A dataset is given to us named as '**The_Grade_Dataset.csv**'. This dataset contains grade points of some student of first year, second year, third year and fourth year courses. We have to create several models to predict final CGPA of a student at the end of fourth year.

By seeing this dataset, we concluded that this problem/task is of supervised learning which contain known features and known target against particular features (known inputs and known outputs).

By seeing this dataset's target variable, it is very clear that our problem of dataset is a regression-based task. Thus, we will implement regression algorithms for each model.

Regression Problem:

Regression, like classification, is a predictive problem setting where we want to use past information to predict future observations. But in the case of regression, the goal is to predict numerical values instead of categorical values. The variable that you want to predict is often called the response variable. For example, we could try to use the number of hours a person spends on exercise each week to predict their race time in the annual Boston marathon. As another example, we could try to use the size of a house to predict its sale price. Both of these response variables—race time and sale price—are numerical, and so predicting them given past data is considered a regression problem.

Data Pre-Processing:

Data preprocessing in Machine Learning is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models. In simple words, data preprocessing in Machine Learning is a data mining technique that transforms raw data into an understandable and readable format.

Data Encoding:

Since we know that machine learning algorithms works (do calculations) with numerical data but here we have whole dataset presents as string data. Therefore, we have to apply some encoding technique to convert these strings into numerical form.

Here, dataset contain grade points in for of A+, A, A-, B+… ....., D, F, WU, W and I. We encode these values by using NED University of Engineering and Technology' s grade points criteria.

Original Values	Encoded Values
A+	4.0
A	4.0
A-	3.7
B+	3.4
B	3.0
B-	2.7
C+	2.4
C	2.0
C-	1.7
D+	1.4
D	1.0
F	0.0
I	0.0
W	0.0
WU	0.0

We are assigning 0.0 to WU, W and I (stands for withdrawal unofficially, withdrawal and incomplete respectively) because these courses will also count as not attempt which is similar to the case of fail.

Feature Engineering:

In the given dataset, there are a lot of missing values in some particular columns and that's a big problem. We have deal with these missing values somehow. That's why we are using value filling method called forward filling and backward filling by using python pandas library.

Forward Fill:

Python is a great language for doing data analysis, primarily because of the fantastic ecosystem of data-centric python packages. Pandas is one of those packages and makes importing and analyzing data much easier. Pandas dataframe.ffill() function is used to fill the

missing value in the dataframe. 'ffill' stands for 'forward fill' and will propagate last valid observation forward.

Backward Fill:

Pandas dataframe.bfill() is used to backward fill the missing values in the dataset. It will backward fill the NaN values that are present in the pandas dataframe.

Dropping Unnecessary Features:

There is a column named '**Seat No.**' which contain roll numbers of particular students. Roll number of students doesn't affect their CGPA. Therefore, we are dropping this column from given dataset.

Model Selection:

We have built three different models as follow,

- 1st year model.
- 2nd year model.
- 3rd year model.

Models Description:

1st year Model:

This model will contain courses of 1st year with grade points of some students and predict final CGPA based on grade points of 1st year courses.

2nd year Model:

This model will contain courses of 1st year and 2nd year with grade points of some students and predict final CGPA based on grade points of up to 2nd year courses.

3rd year Model:

This model will contain courses of 1st year, 2nd year and 3rd year with grade points of some students and predict final CGPA based on grade points of up to 3rd year courses.

Algorithm Selection:

We are required to apply at least two algorithms for each model.

Model 1:

We have applied two algorithms for model 1 as follows,

- KNN regressor (K – nearest – neighbor).

- Linear regression.

Model 2:

Two algorithms that we selected for model 2 are,

- Polynomial regression.
- Support vector regressor.

Model 3:

Model 3 is tested with 2 different algorithms, named as follow,

- Decision tree regressor.
- Multi-layer regressor.

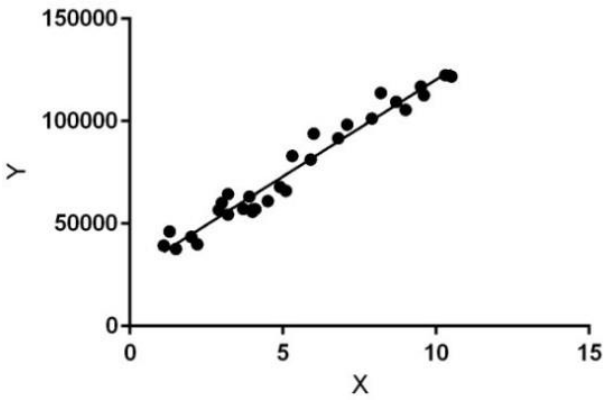
Algorithms Description.

Algorithms for 1st year Model:

Linear Regression:

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables.

The graph below shows the working of simple Linear regression.



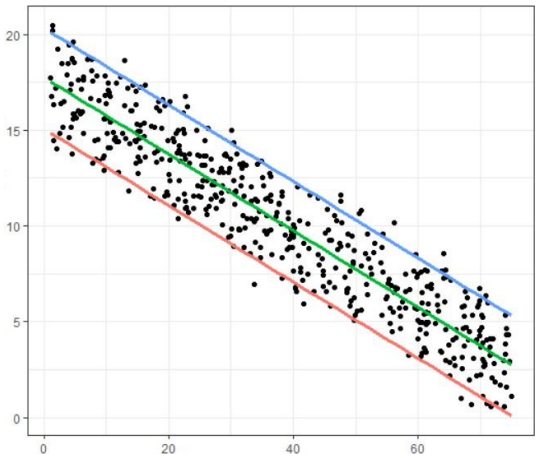
Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.

In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

We apply Linear regression on 1st year model which contains 11 features and 1 target. Therefore, we have to use Multiple linear regression. The main difference is Simple linear regression has only one input feature and one target variable. Multiple linear regression has one target variable and two or more input features. Thus, hypothesis will also be different for multiple linear regression.

Hypothesis => $y(h) = theta_0 + x_1 * theta_1 + x_2 * theta_2 \ldots\ldots\ldots x_n * theta_n$

Where $theta_0$, $theta_1$, $theta_2 \ldots\ldots\ldots theta_n$ are learning parameters.



KNN Regressor:

K nearest neighbors is a simple algorithm that stores all available cases and predict the numerical target based on similarity measures (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

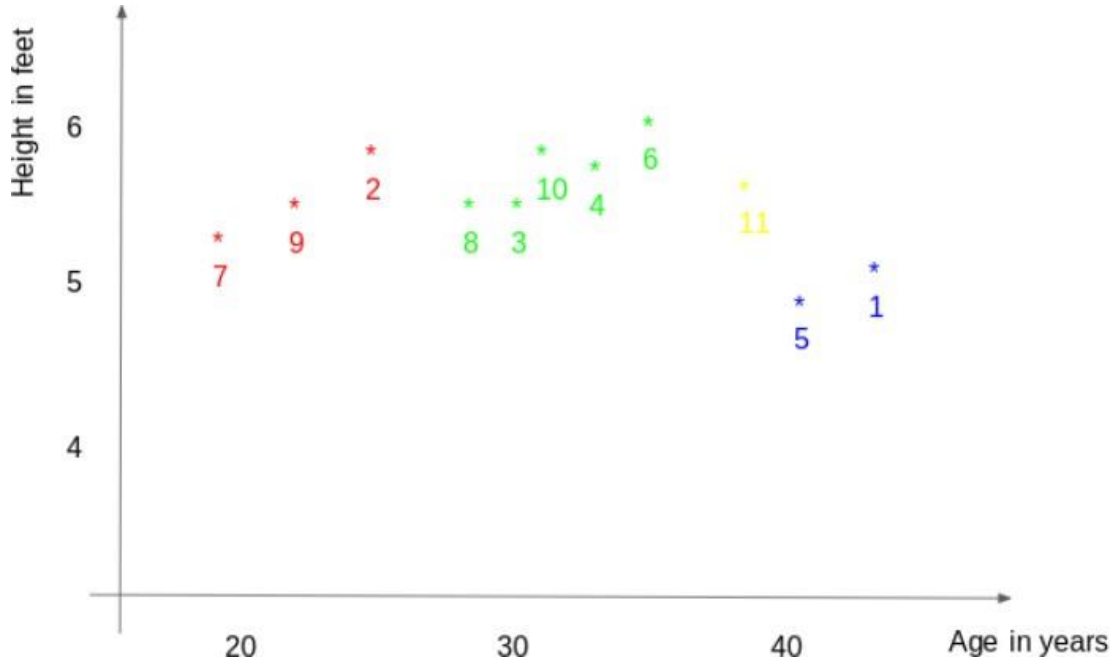
A simple implementation of KNN regression is to calculate the average of numerical target of the K nearest neighbors. KNN regression uses the same distance functions as KNN classification.

Let us start with a simple example. Consider the following table – it consists of the height, age and weight (target) value for 10 people. As you can see, the weight value of ID11 is

missing. We need to predict the weight of this person based on their height and age.

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

For a clearer understanding of this, below is the plot of height versus age from the above table:



In the above graph, the y-axis represents the height of a person (in feet) and the x-axis represents the age (in years). The points are numbered according to the ID values. The yellow point (ID 11) is our test point.

Working of KNNalgorithm:

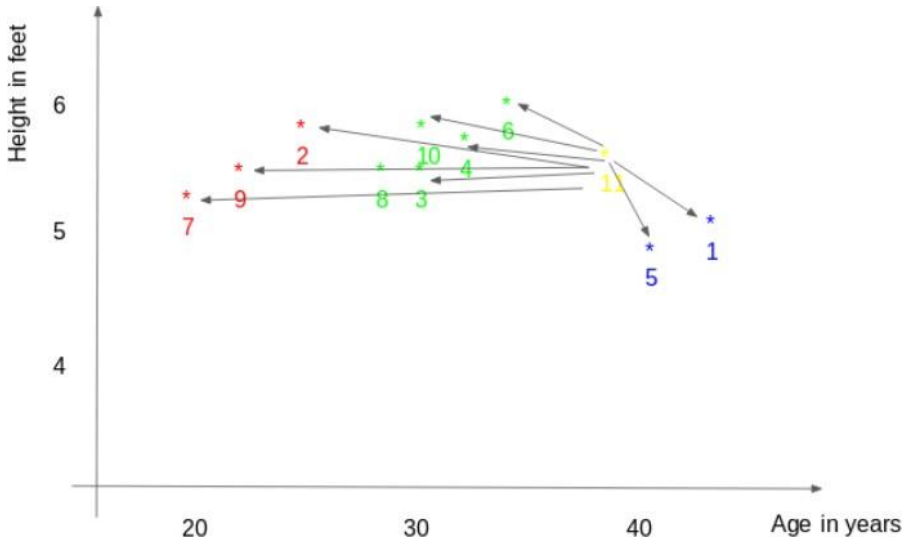
As we saw above, KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses 'feature similarity' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles

the points in the training set. From our example, we know that ID11 has height and age similar to ID1 and ID5, so the weight would also approximately be the same.

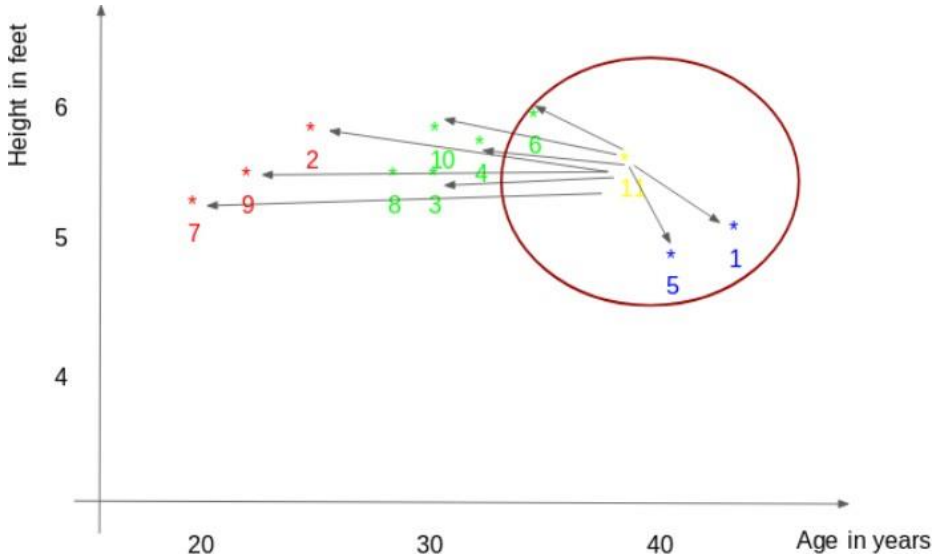
Had it been a classification problem, we would have taken the mode as the final prediction. In this case, we have two values of weight – 72 and 77. Any guesses on how the final value will be calculated? The average of the values is taken to be the final prediction.

Below is a stepwise explanation of the algorithm:

- First, the distance between the new point and each training point is calculated.



- The closest k data points are selected (based on the distance). In this example, points 1, 5, 6 will be selected if the value of k is 3. We will further explore the method to select the right value of k later in this article.



- The average of these data points is the final prediction for the new point. Here, we have weight of ID11 = (77+72+60)/3 = 69.66 kg.

Now, let's discuss the methods for calculating distance between points. These methods are,

- **Euclidean Distance:** Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

- **Manhattan Distance:** This is the distance between real vectors using the sum of their absolute difference.

Distance functions

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\sum_{i=1}^{k}|x_i - y_i|$

Algorithms for 2nd year Model:

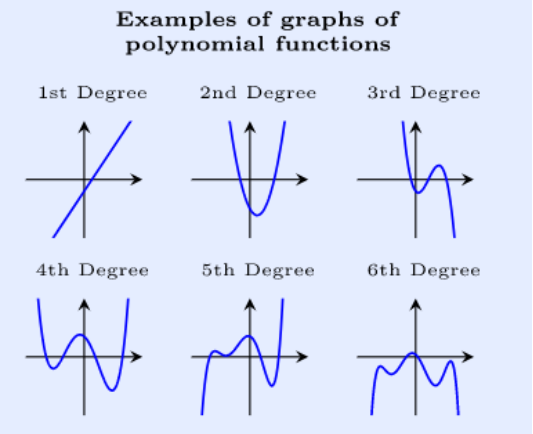
Polynomial Regression:

In polynomial regression, the relationship between the independent variable x and the dependent variable y is described as an nth degree polynomial in x. Polynomial regression, abbreviated E (y |x), describes the fitting of a nonlinear relationship between the value of x and the conditional mean of y. It usually corresponded to the least-squares method. According to the Gauss Markov Theorem, the least square approach minimizes the variance of the coefficients. This is a type of Linear Regression in which the dependent and independent variables have a curvilinear relationship and the polynomial equation is fitted to the data.

Hypothesis of polynomial regression is like:

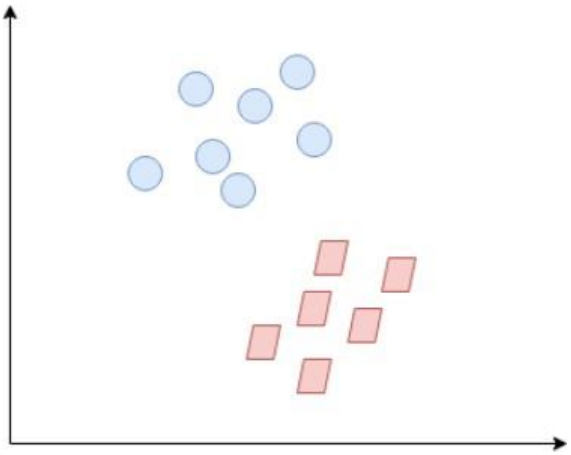
y (h) => $theta_0$ +theta1*x_1^1 + $theta_2$*x_2^2……. thetai*x_i^n

If we take polynomial degree as 2, then hypothesis will be quadratic and I we take degree as 3, then hypothesis will be cubic. Some graphical examples are,

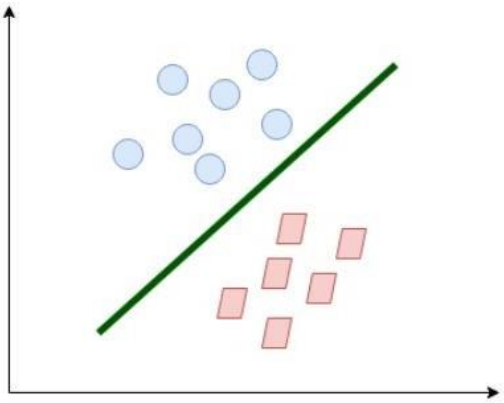


Support Vector Regressor:

First of all, lets discuss about SVM (support vector machine). So, what exactly is Support Vector Machine (SVM)? We'll start by understanding SVM in simple terms. Let's say we have a plot of two label classes as shown in the figure below:

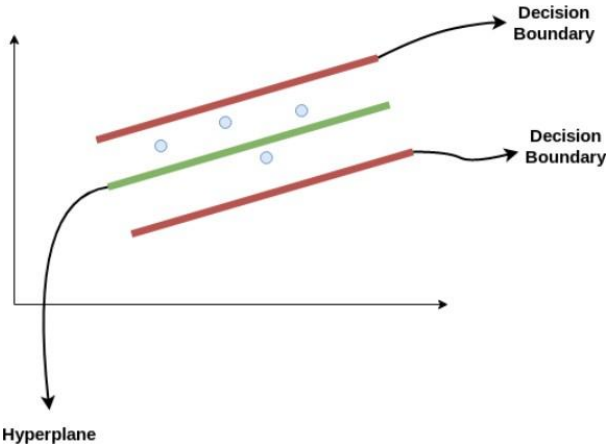


Can you decide what the separating line will be? You might have come up with this:



The line fairly separates the classes. This is what SVM essentially does, simple class separation.

Support Vector Regression (SVR) uses the same principle as SVM, but for regression problems. Let's spend a few minutes understanding the idea behind SVR. Below is the diagram showing the working of support vector regressor.



Consider these two red lines as the decision boundary and the green line as the hyperplane. Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line. Our best fit line is the hyperplane that has a maximum number of points.

The first thing that we'll understand is what is the decision boundary (the danger red line above!). Consider these lines as being at any distance, say 'a', from the hyperplane. So, these are the lines that we draw at distance '+a' and '-a' from the hyperplane. This 'a' in the text is basically referred to as epsilon.

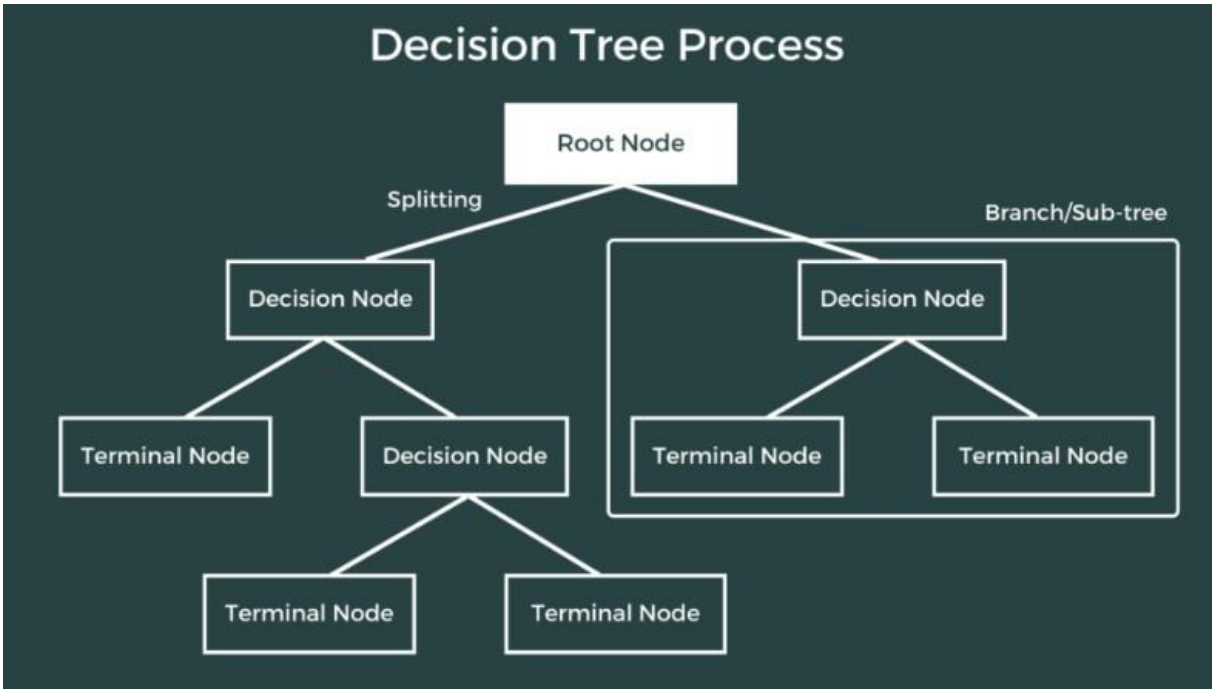
So, concluding that, we can think of Support Vector Regression as the counterpart of SVM for regression problems. SVR acknowledges the presence of non-linearity in the data and provides a proficient prediction model.

Algorithm for 3rd year Model:

Decision Tree Regressor:

A decision tree is one of the most frequently used Machine Learning algorithms for solving regression as well as classification problems. As the name suggests, the algorithm uses a tree-like model of decisions to either predict the target value (regression) or predict the target class (classification).

General structure of decision tree looks like,



- Root Node: This represents the topmost node of the tree that represents the whole data points.
- Splitting: It refers to dividing a node into two or more sub-nodes.
- Decision Node: They are the nodes that are further split into sub-nodes, i.e., this node that is split is called a decision node.
- Leaf / Terminal Node: Nodes that do not split are called Leaf or Terminal nodes. These nodes are often the final result of the tree.
- Branch / Sub-Tree: A subsection of the entire tree is called branch or sub-tree.

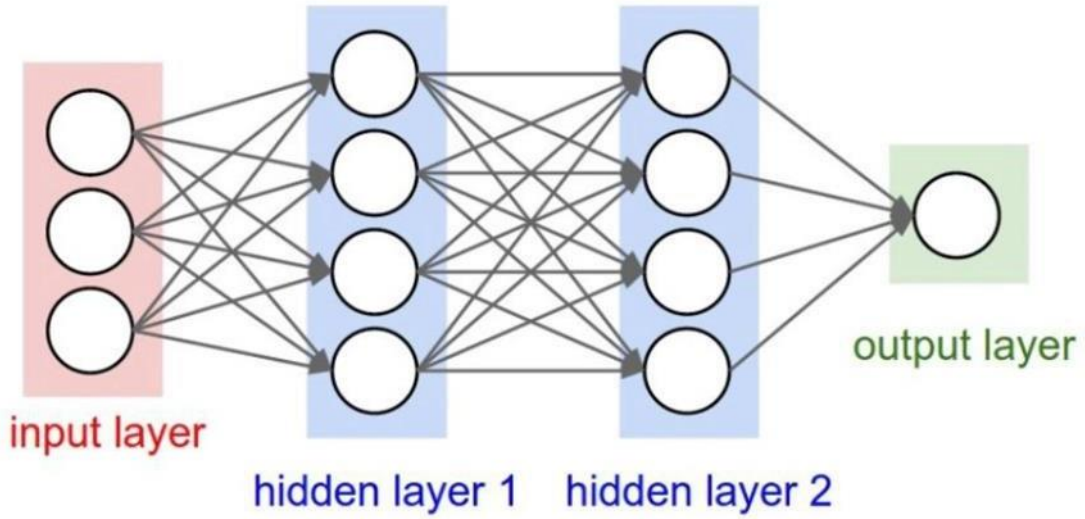
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of the parent node. In the figure above, the decision node is the parent of the terminal nodes (child).
- Parent and Child Node: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of the parent node. In the figure above, the decision node is the parent of the terminal nodes (child).

The process of splitting starts at the root node and is followed by a branched tree that finally leads to a leaf node (terminal node) that contains the prediction or the final outcome of the algorithm. Construction of decision trees usually works top-down, by choosing a variable at each step that best splits the set of items. Each sub-tree of the decision tree model can be represented as a binary tree where a decision node splits into two nodes based on the conditions.

Decision trees where the target variable or the terminal node can take continuous values (typically real numbers) are called regression trees.

Multi-layer Perceptron:

Multi-layer perceptron (MLP) is an artificial neural network that has 3 or more layers of perceptron. These layers are- a single input layer, 1 or more hidden layers, and a single output layer of perceptron. The data flows in a single direction, that is forward, from the input layers-> hidden layer(s) -> output layer. Backpropagation is a technique where the multi-layer perceptron receives feedback on the error in its results and the MLP adjusts its weights accordingly to make more accurate predictions in the future. MLP is used in many machine learning techniques like classification and regression.



A neural network executes in two phases: Feed-Forward and Back Propagation.

Feed Forward Propagation:

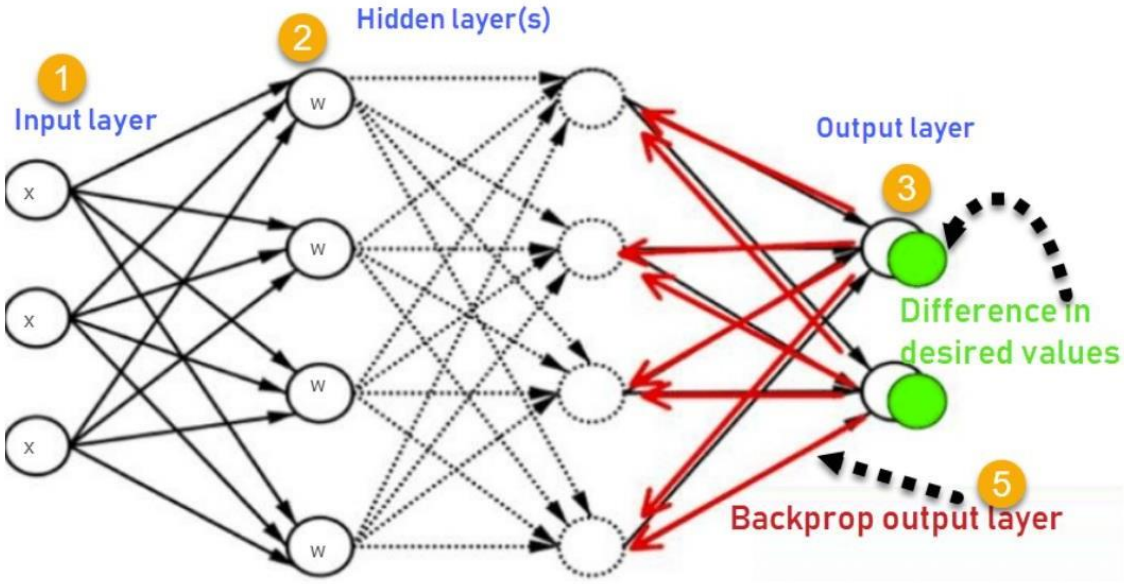
Following are the steps performed during the feed-forward phase:

- The values received in the input layer are multiplied with the weights. A bias is added to the summation of the inputs and weights in order to avoid null values.
- Each neuron in the first hidden layer receives different values from the input layer depending upon the weights and bias.
- The outputs from the first hidden layer neurons are multiplied with the weights of the second hidden layer; the results are summed together and passed to the neurons of the proceeding layers. This process continues until the outer layer is reached. The values calculated at the outer layer are the actual outputs of the algorithm.

The feed-forward phase consists of these three steps. However, the predicted output is not necessarily correct right away; it can be wrong, and we need to correct it. The purpose of a learning algorithm is to make predictions that are as accurate as possible. To improve these predicted results, a neural network will then go through a back-propagation phase. During back propagation, the weights of different neurons are updated in a way that the difference between the desired and predicted output is as small as possible.

Back Propagation:

Backward Propagation is the preferable method of adjusting or correcting the weights to reach the minimized loss function. Backward propagation is basically used for calculating the errors. It is a standard method of training artificial neural networks. This method helps calculate the gradient of a loss function with respect to all the weights in the network.



How Backpropagation Algorithm Works

This one cycle of feed-forward and back propagation is called one "epoch".

NOTE: MLP regressor doesn't have any activation function in the output layer.

Overfitting problem:

Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the model's ability to generalize.

In short, a model overfit if it goes high training accuracy but low testing accuracy.

Algorithms that Overfit on specific Model:

In our case, there are two algorithms that are getting overfitting issues.

- Polynomial Regression.
- Decision Tree Regression.

Techniques to prevent/avoid/reduce Overfitting:

Polynomial Regression:

There are several ways to prevent polynomial regression from overfitting. Either we can reduce the degree of polynomial or we can give our model more training data, can reduce some features or we can use some techniques which can help in avoiding overfitting such as regularization.

Regularization:

Regularization is a technique to discourage the complexity of the model. It does this by penalizing the loss function. This helps to solve the overfitting problem. There are three different types of regularization.

- L1 or Lasso.
- L2 or Ridge.
- L3 or Elastic Net.

We used ridge regularization technique to prevent polynomial regression from overfitting. Therefore, we will discuss Ridge regularization.

Ridge Regularization:

To overcome the problem of overfitting we use a machine learning algorithm called Ridge Regression. Ridge regression addresses some of the problems of Ordinary Least Squares by imposing a penalty on the size of the coefficients.

$$min \sum_{i=1}^{i=n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{j=D} w_j^2$$

Cost function for Ridge

In Ridge regression we square the weight, multiply it by a l2_penalty term(lambda) and add it to the RSS (Residual sum of square). Then we try to minimize this value. The lambda parameter controls the shrinkage of the term. If it's set to 0 then the entire equation becomes like normal Linear Regression curve.

Decision Tree Regressor:

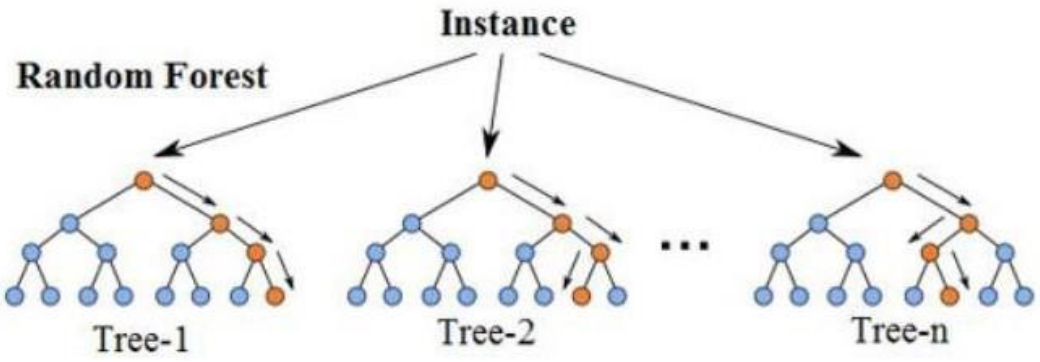
Decision Trees are a non-parametric supervised machine learning approach for classification and regression tasks. Comparing to other machine learning algorithms, decision trees can easily overfit. There are various methods to prevent decision tree from overfitting which are very common as follows.

- Pruning.
  - Pre-pruning.
  - Post-pruning.
- Ensemble.
  - Random Forest.

We have applied the ensemble technique Random forest. Therefore, we are discussing Random forest only.

Random Forest:

Random forest is a type of supervised learning algorithm that uses ensemble methods (bagging) to solve both regression and classification problems. The algorithm operates by constructing a multitude of decision trees at training time and outputting the mean/mode of prediction of the individual trees and follows bootstrap sampling and aggregation techniques to prevent overfitting.



The fundamental concept behind random forest is the wisdom of crowds wherein a large number of uncorrelated models operating as a committee will outperform any of the individual constituent models.

The reason behind this is the fact that the trees protect each other from their individual errors. Within a random forest, there is no interaction between the individual trees. A random forest acts as an estimator algorithm that aggregates the result of many decision trees and then outputs the most optimal result.

Input Function:

This input function has been created for user if he/she want his/her CGPA to be predicted from three different models.

This function will first ask you which model you want to select as per following rules.

- Press 1 to select 1st year model.
- Press 2 to select 2nd year model.
- Press 3 to select 3rd year model.

If user press any other key apart from 1,2 and 3, an error message will appear as "Please press valid key!"

```
Hello People! welcome to CGPA predictor.

=============================================================================================================================

There are 3 different models according to which your CGPA will be predicted.

1) 1st Model: This model will take your 1st year Grade Points of individual courses as input.
2) 2nd Model: This model will take your Grade Points of individual courses of 1st and 2nd year as input.
3) 3rd Model: This model will take your Grade Points of individual courses of 1st, 2nd and 3rd year courses as input.

=============================================================================================================================

Press 1 to select 1st model.
Press 2 to select 2nd model.
Press 3 to select 3rd model

Press Key: 4

Please press valid key!
Press Key: q

Please press valid key!

Press Key:
```

If user press 1 than 1st year model will be selected user will get the message "**You have selected 1st year model. Now give us your individual grade points of 1st year courses.**"

```
Hello People! welcome to CGPA predictor.

=============================================================================================================================

There are 3 different models according to which your CGPA will be predicted.

1) 1st Model: This model will take your 1st year Grade Points of individual courses as input.
2) 2nd Model: This model will take your Grade Points of individual courses of 1st and 2nd year as input.
3) 3rd Model: This model will take your Grade Points of individual courses of 1st, 2nd and 3rd year courses as input.

=============================================================================================================================

Press 1 to select 1st model.
Press 2 to select 2nd model.
Press 3 to select 3rd model

Press Key: 1
You have selected 1st year model. Now give us your individual grade points of 1st year courses.

Enter GP of Subject PH-121:
```

If user press 2 than 2nd year model will be selected and user will get the message "**You have selected 2nd year model. Now give us your individual grade points of 1st year and 2nd year courses.**"

```
Hello People! welcome to CGPA predictor.

=============================================================================================================

There are 3 different models according to which your CGPA will be predicted.

1) 1st Model: This model will take your 1st year Grade Points of individual courses as input.
2) 2nd Model: This model will take your Grade Points of individual courses of 1st and 2nd year as input.
3) 3rd Model: This model will take your Grade Points of individual courses of 1st, 2nd and 3rd year courses as input.

=============================================================================================================

Press 1 to select 1st model.
Press 2 to select 2nd model.
Press 3 to select 3rd model

Press Key: 2
You have selected 2nd year model. Now give us your individual grade points of 1st year and 2nd year courses.

Enter GP of Subject PH-121:
```

If user press 3 than 3rd year model will be selected and user will get the message "**You have selected 2nd year model. Now give us your individual grade points of 1st year, 2nd year and 3rd year courses.**"

```
Hello People! welcome to CGPA predictor.

=============================================================================================================

There are 3 different models according to which your CGPA will be predicted.

1) 1st Model: This model will take your 1st year Grade Points of individual courses as input.
2) 2nd Model: This model will take your Grade Points of individual courses of 1st and 2nd year as input.
3) 3rd Model: This model will take your Grade Points of individual courses of 1st, 2nd and 3rd year courses as input.

=============================================================================================================

Press 1 to select 1st model.
Press 2 to select 2nd model.
Press 3 to select 3rd model

Press Key: 3
You have selected 2nd year model. Now give us your individual grade points of 1st year, 2nd year and 3rd year courses.

Enter GP of Subject PH-121:
```

After selecting any model user will be asked for his/her grade points of particular year's courses and their CGPA will be predicted.

NOTE: User can only give following grades as input,

A+	4.0
A	3.7
A-	3.4
B+	3.0
B	2.7
B-	2.4
C+	2.0
C	1.7
C-	1.4
D+	1.0
D	0.0
F	
I	
WU	
W	

These values are in string and float class. If user give lower case alphabetic value and integer number as input then the program will convert lower case to upper case alphabets and integer to float values.

If user give any value other than values mention in above table, then he/she will get the message "**Please enter valid key**"

```
Hello People! welcome to CGPA predictor.

==========================================================================================================================================

There are 3 different models according to which your CGPA will be predicted.

1) 1st Model: This model will take your 1st year Grade Points of individual courses as input.
2) 2nd Model: This model will take your Grade Points of individual courses of 1st and 2nd year as input.
3) 3rd Model: This model will take your Grade Points of individual courses of 1st, 2nd and 3rd year courses as input.

==========================================================================================================================================

Press 1 to select 1st model.
Press 2 to select 2nd model.
Press 3 to select 3rd model

Press Key: 1
You have selected 1st year model. Now give us your individual grade points of 1st year courses.

Enter GP of Subject PH-121: q

Please enter valid key
Enter GP of Subject PH-121: 2.5

Please enter valid key

Enter GP of Subject PH-121:
```

1st year model:

Let's see an example of how predicted CGPA will look like if user select 1st year model.

```
You have selected 1st year model. Now give us your individual grade points of 1st year courses.

Enter GP of Subject PH-121: q

Please enter valid key
Enter GP of Subject PH-121: 2.5

Please enter valid key
Enter GP of Subject PH-121: A
Enter GP of Subject HS-101: a
Enter GP of Subject CY-105: A+
Enter GP of Subject HS-105/12: a+
Enter GP of Subject MT-111: 4
Enter GP of Subject CS-105: 4.0
Enter GP of Subject CS-106: 2.7
Enter GP of Subject EL-102: 0
Enter GP of Subject EE-119: 0.0
Enter GP of Subject ME-107: 1.4
Enter GP of Subject CS-107: c

=============================================================================================================================

Your CGPA should be [2.52396576] according to Linear Regression.
Your CGPA should be [2.69946667] according to KNN Regressor.
```

2nd year Model:

Now, see an example of how predicted CGPA will look like if user select 2nd year model.

```
You have selected 2nd year model. Now give us your individual grade points of 1st year and 2nd year courses.

Enter GP of Subject PH-121: A+
Enter GP of Subject HS-101: a-
Enter GP of Subject CY-105: A
Enter GP of Subject HS-105/12: B+
Enter GP of Subject MT-111: b-
Enter GP of Subject CS-105: b
Enter GP of Subject CS-106: C
Enter GP of Subject EL-102: 2
Enter GP of Subject EE-119: D
Enter GP of Subject ME-107: 1
Enter GP of Subject CS-107: 0.0
Enter GP of Subject HS-205/20: F
Enter GP of Subject MT-222: f
Enter GP of Subject EE-222: wu
Enter GP of Subject MT-224: W
Enter GP of Subject CS-210: I
Enter GP of Subject CS-211: i
Enter GP of Subject CS-203: 4
Enter GP of Subject CS-214: 4
Enter GP of Subject EE-217: a+
Enter GP of Subject CS-212: A
Enter GP of Subject CS-215: a

=============================================================================================================================

Your CGPA should be [2.13350974] according to Polynomial Regression.
Your CGPA should be [2.46540028] according to Support Vector Regressor.
```

3rd year Model:

At last, let's see an example of how predicted CGPA will look like if user select 3rd year model.

```
Press Key: 3
You have selected 2nd year model. Now give us your individual grade points of 1st year, 2nd year and 3rd year courses.

Enter GP of Subject PH-121: A+
Enter GP of Subject HS-101: a+
Enter GP of Subject CY-105: A
Enter GP of Subject HS-105/12: a
Enter GP of Subject MT-111: A-
Enter GP of Subject CS-105: a-
Enter GP of Subject CS-106: B+
Enter GP of Subject EL-102: b+
Enter GP of Subject EE-119: B
Enter GP of Subject ME-107: b
Enter GP of Subject CS-107: B-
Enter GP of Subject HS-205/20: b-
Enter GP of Subject MT-222: C+
Enter GP of Subject EE-222: c+
Enter GP of Subject MT-224: C
Enter GP of Subject CS-210: c
Enter GP of Subject CS-211: C-
Enter GP of Subject CS-203: c-
Enter GP of Subject CS-214: D+
Enter GP of Subject EE-217: d+
Enter GP of Subject CS-212: D
Enter GP of Subject CS-215: d
Enter GP of Subject MT-331: 4
Enter GP of Subject EF-303: 3.7
Enter GP of Subject HS-304: 3.4
Enter GP of Subject CS-301: 3
Enter GP of Subject CS-302: 2.7
Enter GP of Subject TC-383: 2.4
Enter GP of Subject EL-332: 2
Enter GP of Subject CS-318: 1.7
Enter GP of Subject CS-306: 1.4
Enter GP of Subject CS-312: 1
Enter GP of Subject CS-317: 0

================================================================================================================================

Your CGPA should be [2.48364667] according to Decision Tree Regressor.
Your CGPA should be [2.42172332] according to Multi Layer Perceptron.
```

Model Comparison:

There are different performance measure metrics for checking the performance of regression algorithm. The comparison of models can be done by seeing R_2 score, mean square error, root mean square error, mean absolute error etc.

We have used R_2 score, mean square error, root mean square error and mean absolute error for each and every algorithm that we implemented for each model.

The tabular comparison of these values is:

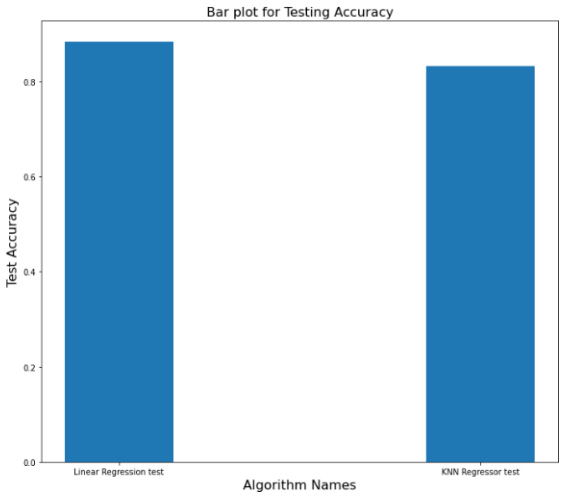
Tabular Comparison:

Algorithms Name:	R_2 Score	Mean Squared Error	Root Mean Squared Error	Mean Absolute Error
Linear Regression	88.35	4.33	20.82	16.15
KNN Regression	83.28	6.22	24.94	18.65
Polynomial Regression	89.35	3.96	19.91	14.22
Support Vector Regressor	90.28	3.61	19.01	12.55
Decision Tree Regressor	88.42	4.31	20.76	10.38
Multi-layer Perceptron	89.82	3.79	19.46	11.43

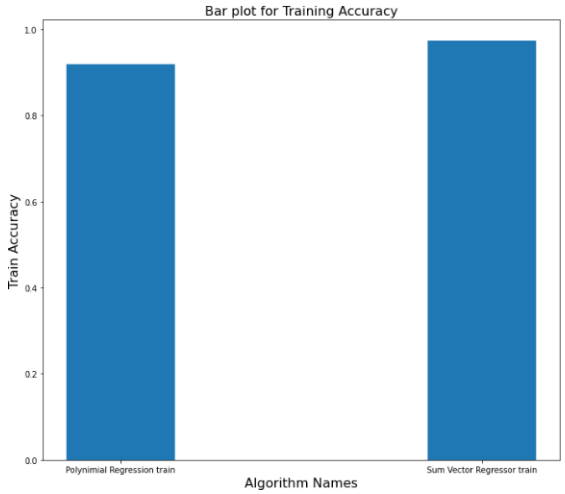
Graphical Comparison:

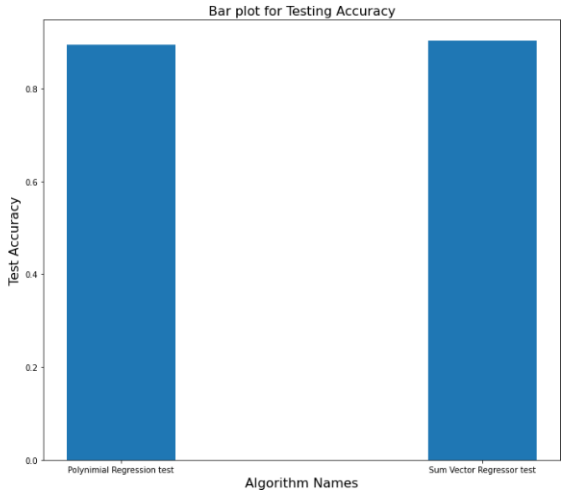
For graphical comparison of algorithms, we used bar plot to plot train and test accuracy of individual algorithms.

1st year model and its implemented algorithms comparison in terms of scoring.

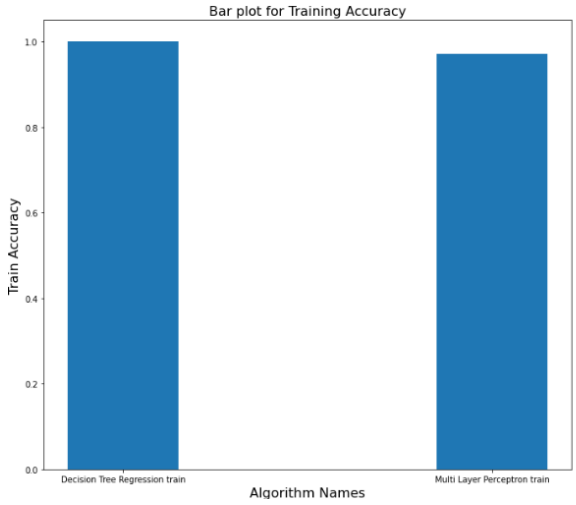


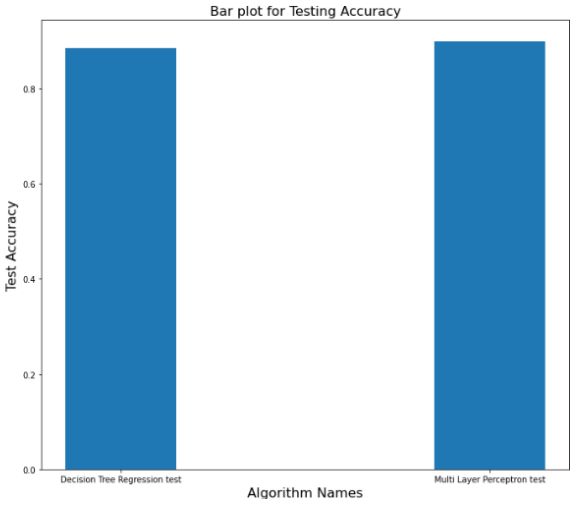
2nd year model and its implemented algorithms comparison in terms of scoring.



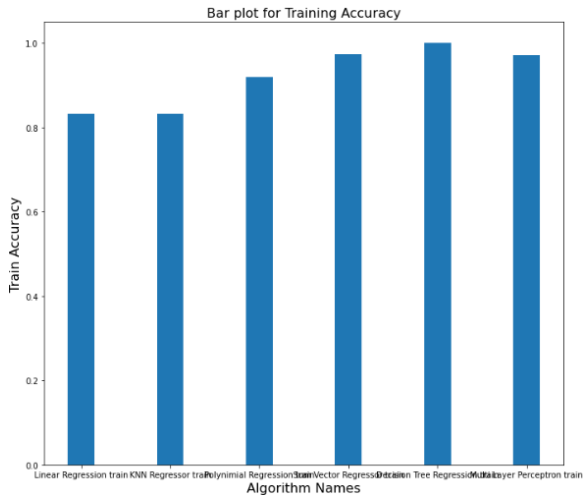


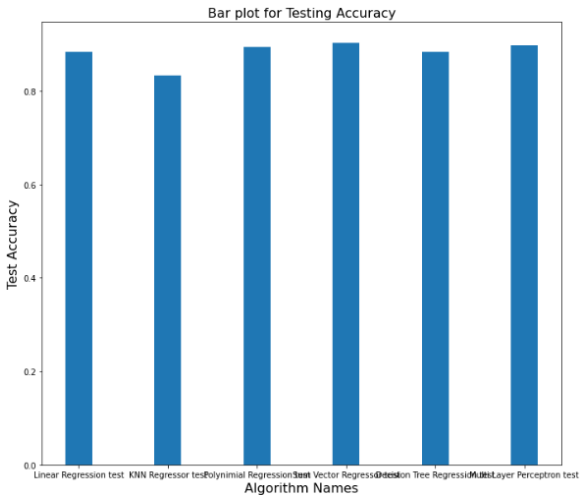
3rd year model and its implemented algorithms comparison in terms of scoring.





Overall (combine) comparison.

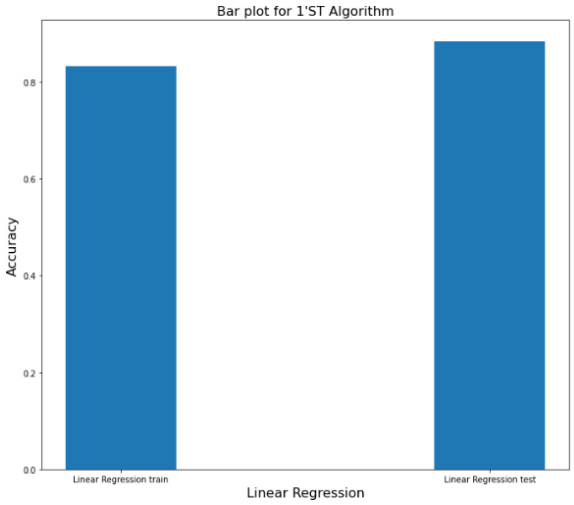


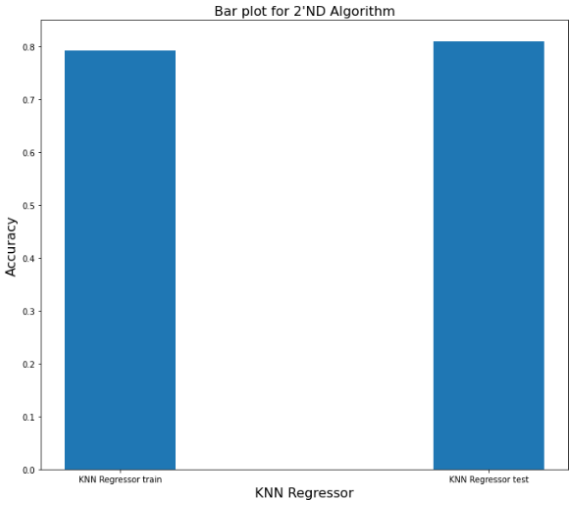


Comparing model's algorithms with each other:

In this section we are comparing algorithms training and testing accuracy to see if there is a case of overfitting or not.

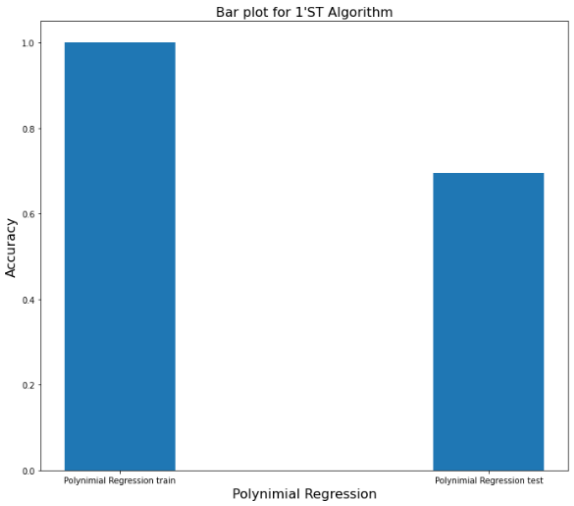
1st year model's algorithms.

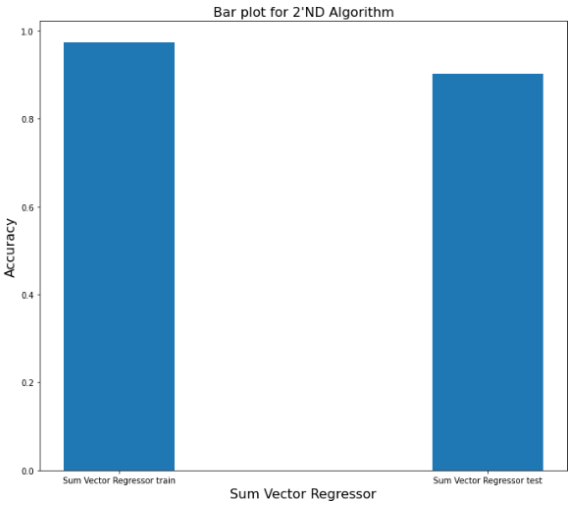




These graphs show that, training and testing accuracy of both algorithms are almost equal. Therefore, there is no presence of overfitting.

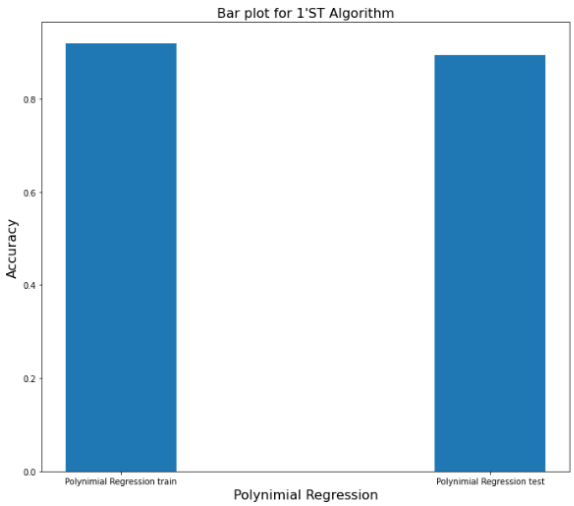
2nd year model's algorithms.

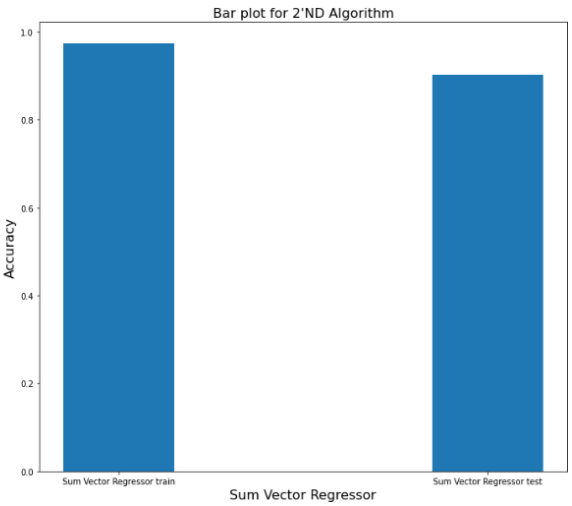




These graphs show that, training and testing score of support vector machine are almost equal or nearby but training accuracy of polynomial regression is far higher than testing accuracy. Therefore, support vector machine doesn't overfitting while polynomial regression does.

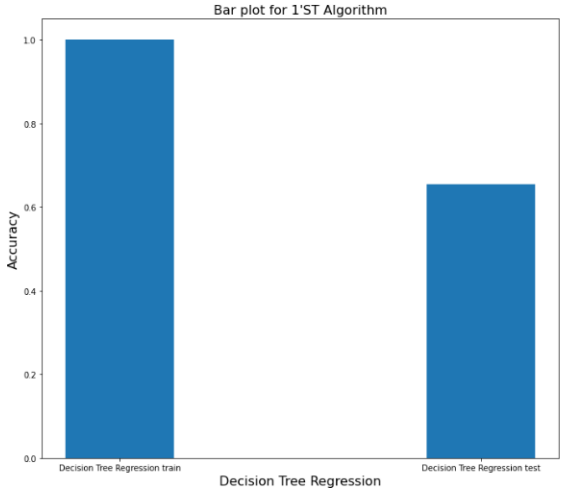
After applying appropriate techniques to avoid overfitting, we get these graphs like.

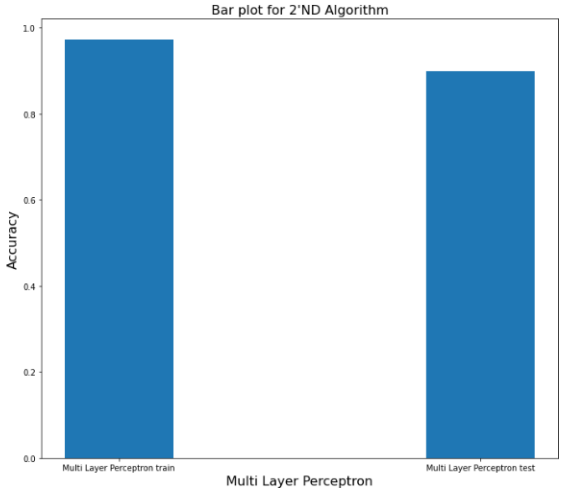




Now, from these graphs we can clearly see that overfitting has been avoided.

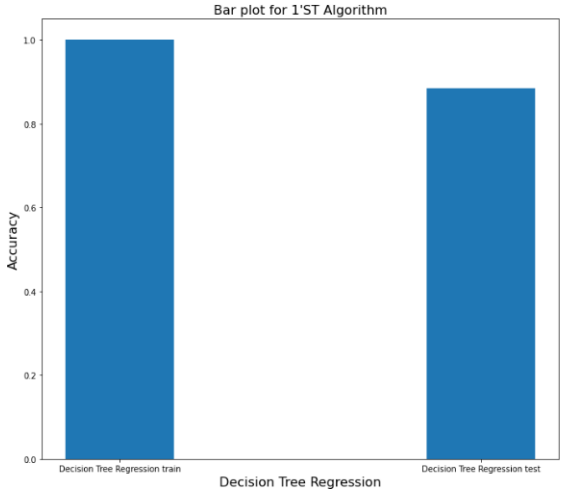
3rd year model's algorithms.

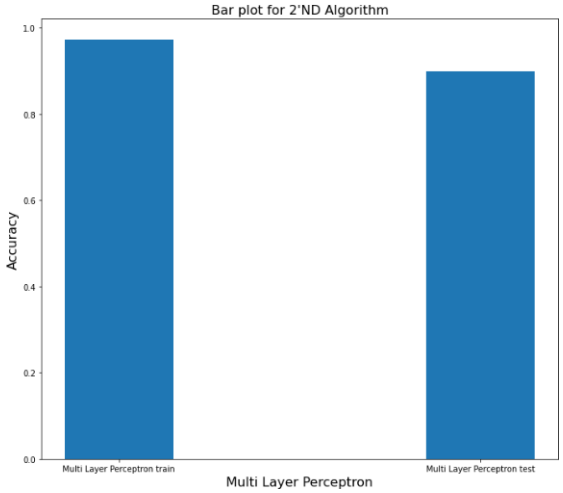




These graphs show that, training and testing score of multi-layer perceptron are almost equal or nearby but training accuracy of Decision tree is far higher than testing accuracy. Therefore, multi-layer perceptron doesn't overfitting while Decision tree does.

After applying appropriate techniques to avoid overfitting, we get these graphs like.

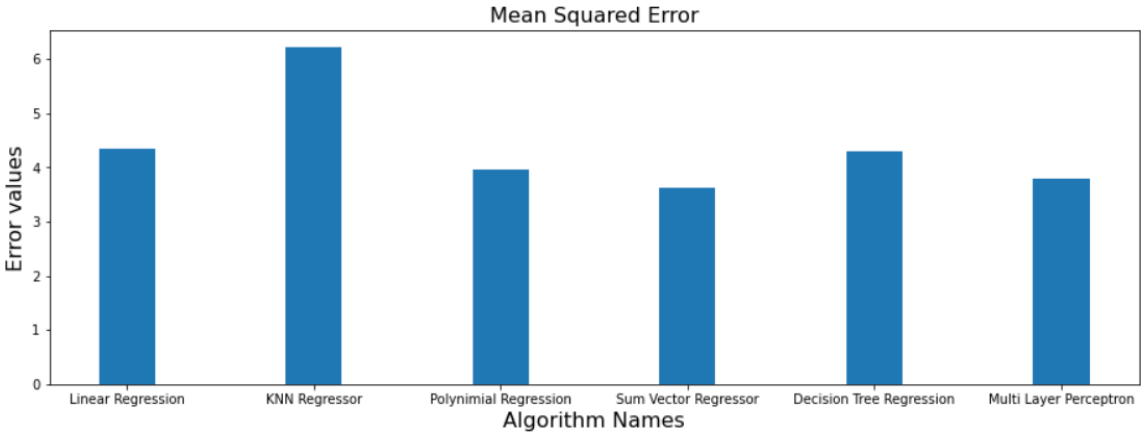




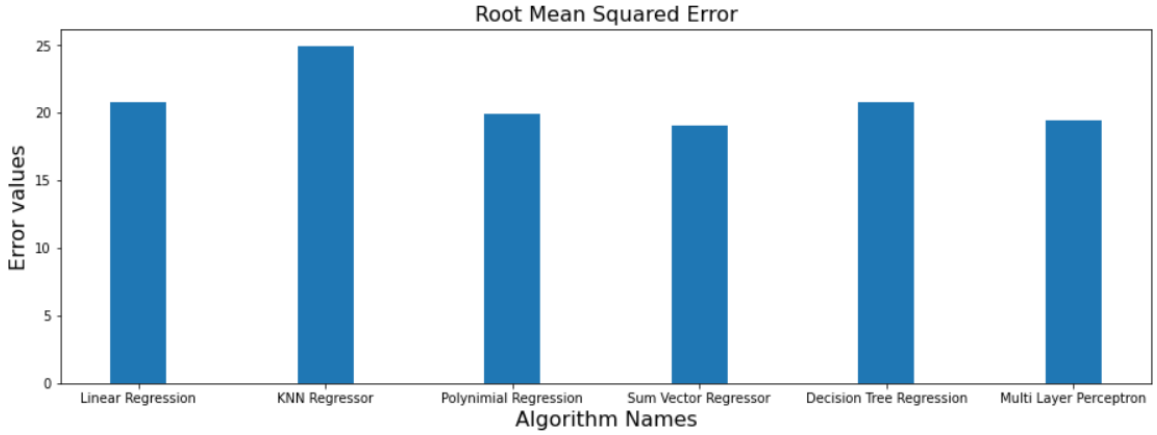
Now, from these graphs we can clearly see that overfitting has been avoided.

Error Comparison:

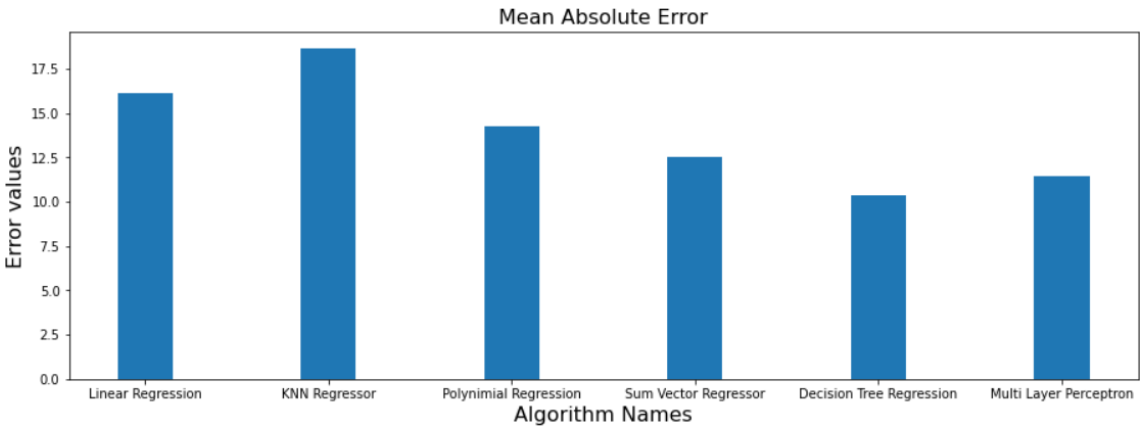
Mean Square Error:



Root Mean Square Error:

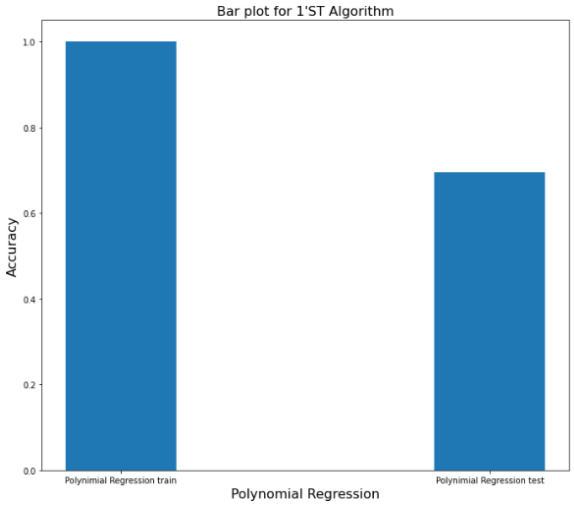


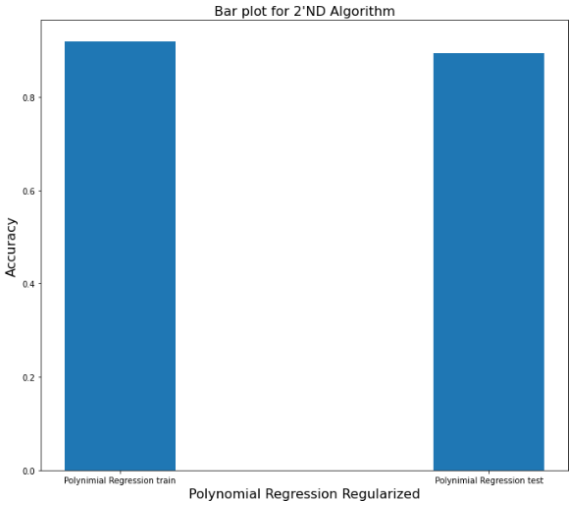
Mean Absolute Error:



Comparing algorithms before and after avoiding Overfitting.

Polynomial Regression:





Decision Tree: