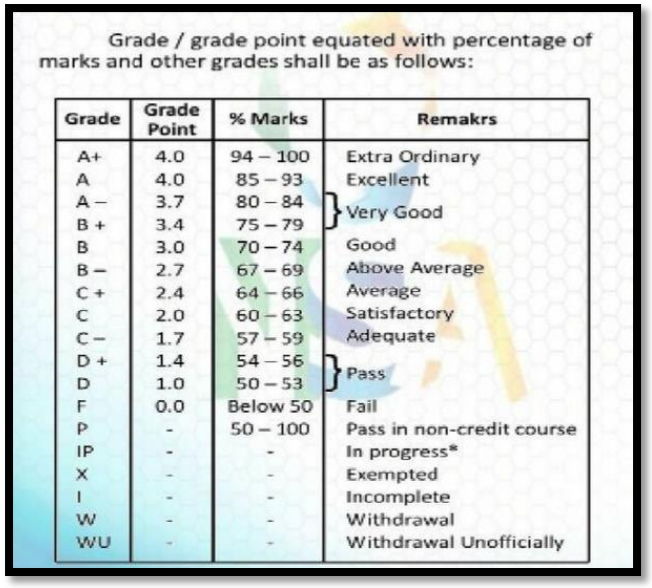
OBJECTIVE:

The main objective of this project is to develop a machine learning system to predict final CGPA of a student on the basis of GPs of previous years.

DATA PRE-PROCESSING STEPS:

- The shape of Grade Dataset is (43,571) which implies that dataset is comprises of 43 features which incorporate Seat No, courses of first year (FE) to final year (BE) and a target CGPA. In total, the dataset has 571 training samples, but it also contains missing, noisy, and inconsistent values. Therefore, preprocess the data first.
- First, apply a nominal encoding (as shown in image below) on courses. As each grade has a weight in the calculation used for CGPA prediction, therefore it is needed to convert the courses from object type to float type.



Grade / grade point equated with percentage of marks and other grades shall be as follows:

Grade	Grade Point	% Marks	Remakrs
A+	4.0	94 – 100	Extra Ordinary
A	4.0	85 – 93	Excellent
A –	3.7	80 – 84	Very Good
B +	3.4	75 – 79	
B	3.0	70 – 74	Good
B –	2.7	67 – 69	Above Average
C +	2.4	64 – 66	Average
C	2.0	60 – 63	Satisfactory
C –	1.7	57 – 59	Adequate
D +	1.4	54 – 56	Pass
D	1.0	50 – 53	
F	0.0	Below 50	Fail
P	-	50 – 100	Pass in non-credit course
IP	-	-	In progress*
X	-	-	Exempted
I	-	-	Incomplete
W	-	-	Withdrawal
WU	-	-	Withdrawal Unofficially

- There are some training samples comprises of grades F, W, WU, and I.
- Students having **grade F** are considered as fail in the course and are encode to zero.
- Students having **grade W or WU** are considered dropped out and are no longer considered part of the institute. Therefore, they are also encoded to zero.
- Students having **grade I** are considered as Incomplete, indicating that some of the required course work was not completed and evaluated within the prescribed time period for an unexpected but fully justified reason. A final grade is assigned when the agreed-upon work has been completed and evaluated. Since the dataset has only one Grade I student, we also encode it to zero.

Now split the dataset into three models.

Model 1: Includes only 1st year GP and predicts final CGPA under this assumption.

- Remove seat numbers and courses from SE, TE and BE. Currently, this model has only 11 courses. The shape of the model 1 is (571,12).
- After checking if model 1 has a null value using the.isnull.sum () method, delete the null row. Only 6 rows are decremented, which is very low, so it is smarter to delete some rows instead of substituting it with mean and median.
- The final shape of model 1 after preprocessing is (566,12).

Model 2: Includes GP for the first two years and predicts the final CGPA under this assumption.

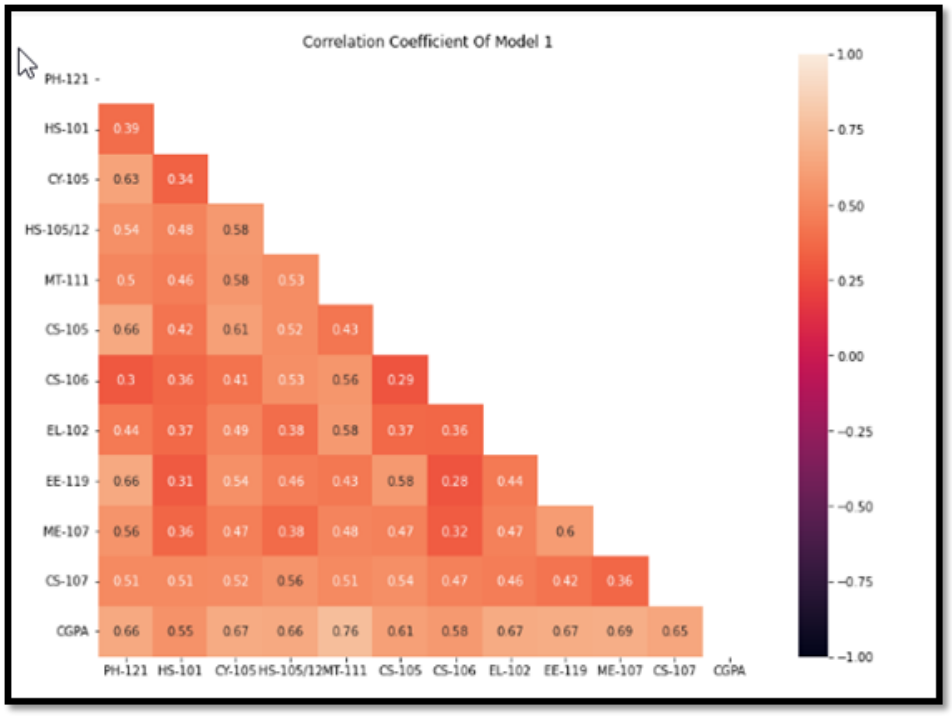
- Remove the seat number and course from TE and BE. Currently, this model has only 22 courses. The shape of the model 2 is (571,23).
- Use the dropna () method to remove the null row. Nine rows are deleted.
- The final shape of model 2 after preprocessing is (562,23).

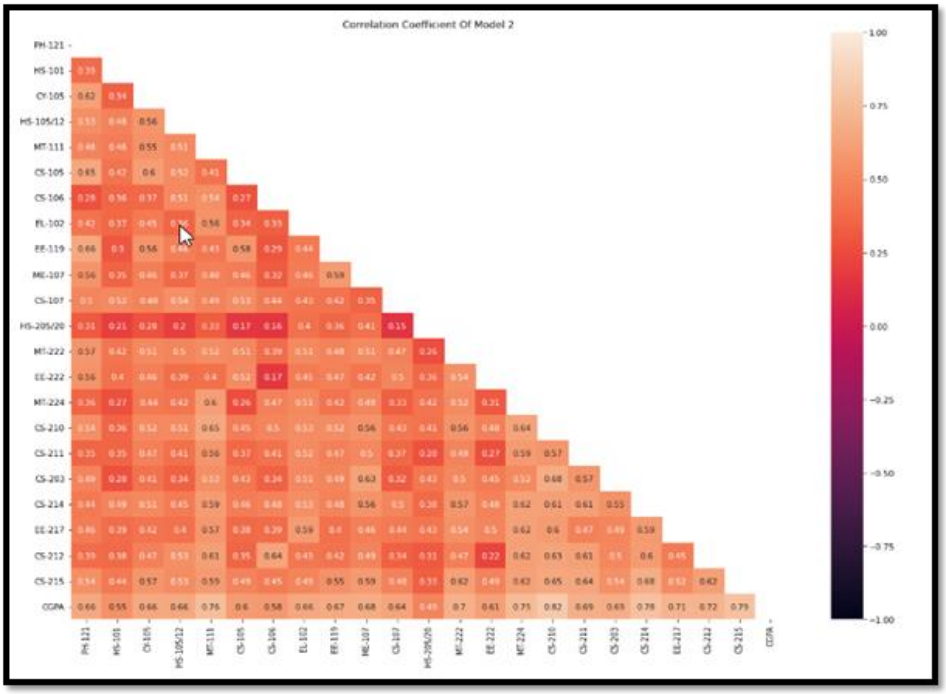
Model 3: Includes GP for the first 3 years and predicts final CGPA under this assumption.

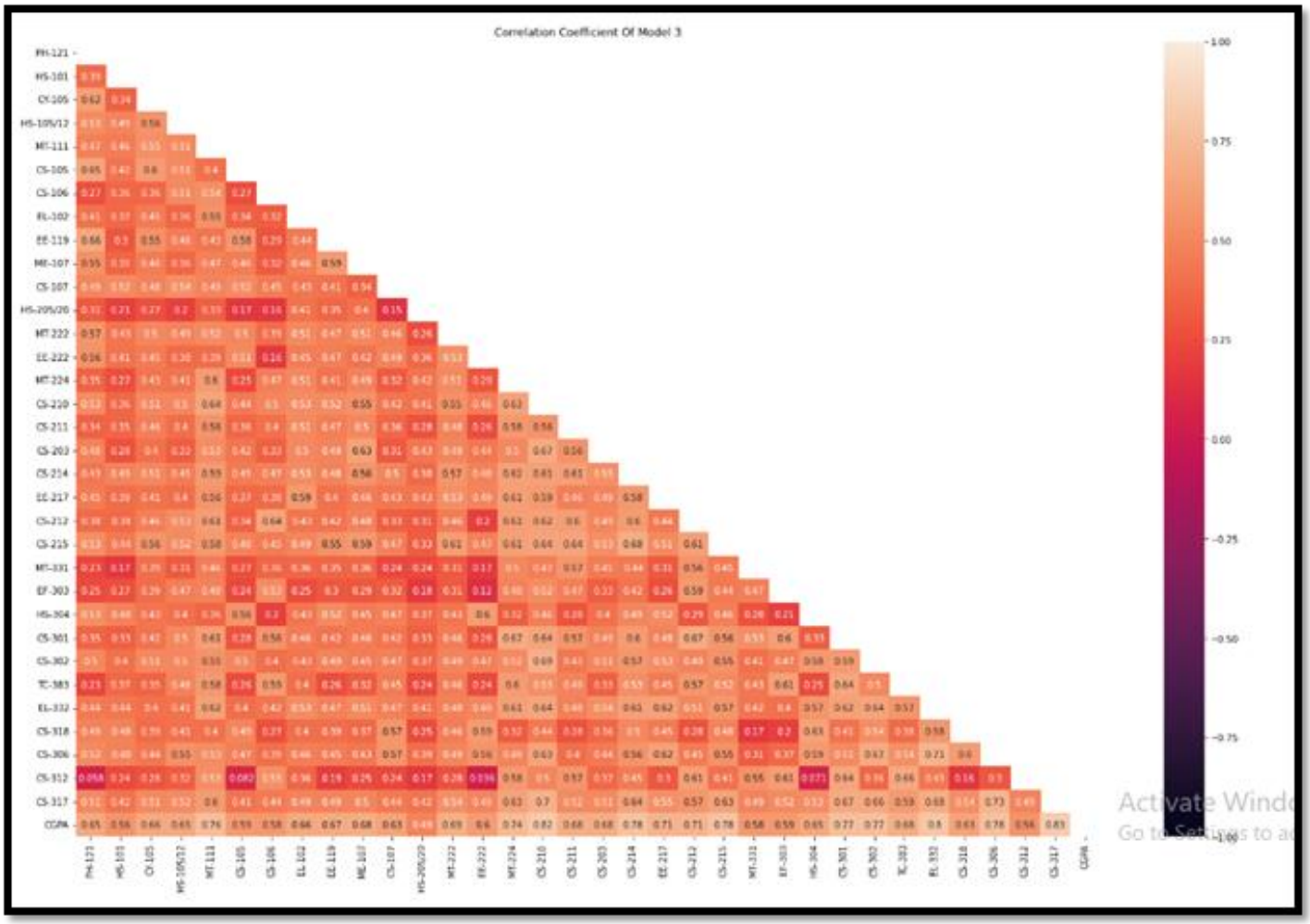
- Remove seat number and course from BE. Currently, this model has only 33 courses. The shape of the model 3 is (571,34).
- Use the dropna () method to remove the null row. The number of deleted rows is 12.
- The final shape of model 3 after preprocessing is (559,34).

Finding Correlation:

Correlation for each model is found to understood which variables were related and how strong this relationship was. Looking at this correlation heatmap for each model, we can see that all variables are positively correlated with each other. Therefore, there is no need to delete any feature.






Correlation Coefficient Of Model 3
PH-121
HS-101
CY-105
HS-105/12
MT-111
CS-105
CS-106
EL-102
EE-119
ME-107
CS-107
HS-205/20
MT-222
EE-222
MT-224
CS-210
CS-211
CS-203
CS-214
EE-217
CS-212
CS-215
MT-331
EF-303
HS-304
CS-301
CS-302
TC-383
EL-332
CS-318
CS-306
CS-312
CS-317
CGPA
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

DETAILS OF MODEL AND MACHINE LEARNING ALGORITHM CHOOSEN FOR IMPLEMENTATION:

Model 1

Model 1 includes only 1st year GP and predicts final CGPA under this assumption. The shape of Model 1 is (566, 12) which means there are 566 training samples and 12 features (11 courses and 1 target variable CGPA). The algorithms applied on Model 1 are:

- **Linear Regression:** finds out a linear relationship between x (input) and y(output).

$$\hat{Y} = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_p X_p$$

Where, $\hat{Y}$ is the predicted or expected value of the dependent variable, X_1 through X_p are p distinct independent or predictor variables, b_0 is the value of Y when all of the independent variables (X_1 through X_p) are equal to zero, and b_1 through b_p are the estimated regression coefficients.

Therefore, Model 1 has more predictors because it only has grades for the first-year course. This means that the prediction rate is high. This is why linear regression on model 1 does not work that well as in models 2 and 3.

Also, we know our dataset has no outliers or multicollinearity therefore, it is recommended to use linear regression.

- **Polynomial Regression:** This is a form of linear regression in which the relationship between the independent variable x and the dependent variable y is modeled as an nth degree polynomial. Here we use the degree 2 of the polynomial that best fits the model. As the model degree increases, the risk of data overfitting and underfitting increases. We also apply polynomial regression to the data because our data have correlations, but the relationships do not look linear.
- **Bayesian Ridge:** Estimate the probabilistic model of the regression problem. This can be a better option in some scenarios if:
    - Data is limited.
    - Uncertainty is important
    - The model (data generation process) is hierarchical.

The Model 1 is based only on the 1st year course (limited data) and is hierarchical, therefore Bayesian Ridge is recommended to use.

Model 2

Model 2 includes 1st year and 2nd year GP and predicts final CGPA under this assumption. The shape of Model 2 is (562, 23) which means there are 562 training samples and 23 features (22 courses and 1 target variable CGPA). The algorithms applied on Model 2 are:

- **Linear Regression:** Since Model 2 is based on the 1st and 2nd year courses, it has fewer predictors than Model 1, so Model 2's linear regression performs better than Model 1.
- **KNN Regressor:** The k-nearest neighbor (k-NN) algorithm is a nonparametric supervised learning method that returns the mean of the k-nearest neighbors.
    - We first create a KNN classification instance, then prepare the range of values for the hyperparameter K from {2,3,9,8,7} that GridSearchCV uses to find the optimal value for K.
    - In GridSearchCV, set the cross-validation batch size cv = 5. Once the model is fitted, we find the best parameters for K and the highest score obtained by GridSearchCV.
    - We see that the best value for K for our model is 9, and the corresponding accuracy is 91.51%.

Model 3

Model 3 includes 1st year, 2nd year and 3rd year GP and predicts final CGPA under this assumption. The shape of Model 3 is (559, 34) which means there are 559 training samples and 34 features (33 courses and 1 target variable CGPA). The algorithms applied on Model 3 are:

- **Linear Regression:** Linear regression works best with Model 3 as the prediction ratio here is least as compare to Model 1 and Model 2.
- **Gradient Boosting:** Gradient Boost is a powerful boosting technique. The accuracy of the model is improved by combining the weak trees in order to form a strong tree. We adjust the hyperparameters to get the best output from the algorithm.
    - ***n_estimator:*** The estimator number indicates the number of trees in the forest. The larger the number of trees, the better it will help you learn the data. The optimum value for this saved hyperparameter kept is 500.
    - ***max_depth:*** This is the estimated depth of the decision tree. The default value is 3, which is an optional parameter. The optimum value for this saved hyperparameter kept is 4.
    - ***learning_rate:*** The learning rate is a hyperparameter of the gradient boosting regressor algorithm and is limited to 0.1.

TABULAR AND GRAPHICAL REPRESENTATION OF MODEL

This table shows a comparison of each model based on MSE, training, and test accuracy. From Table, we can conclude that:

- There is a slight difference in MSE and accuracy of algorithms applied in Model 1, therefore we can say all algorithms perform better on Model 1.
- Linear regression works better in Model 2. This is because the MSE value is low and the test accuracy is higher than the KNN regressor (95.37).
- Linear regression works better in Model 3 as it has lower MSE values and higher test accuracy than gradient boosting (99.03).

Therefore, linear regression works best with models with larger datasets (Model 3)

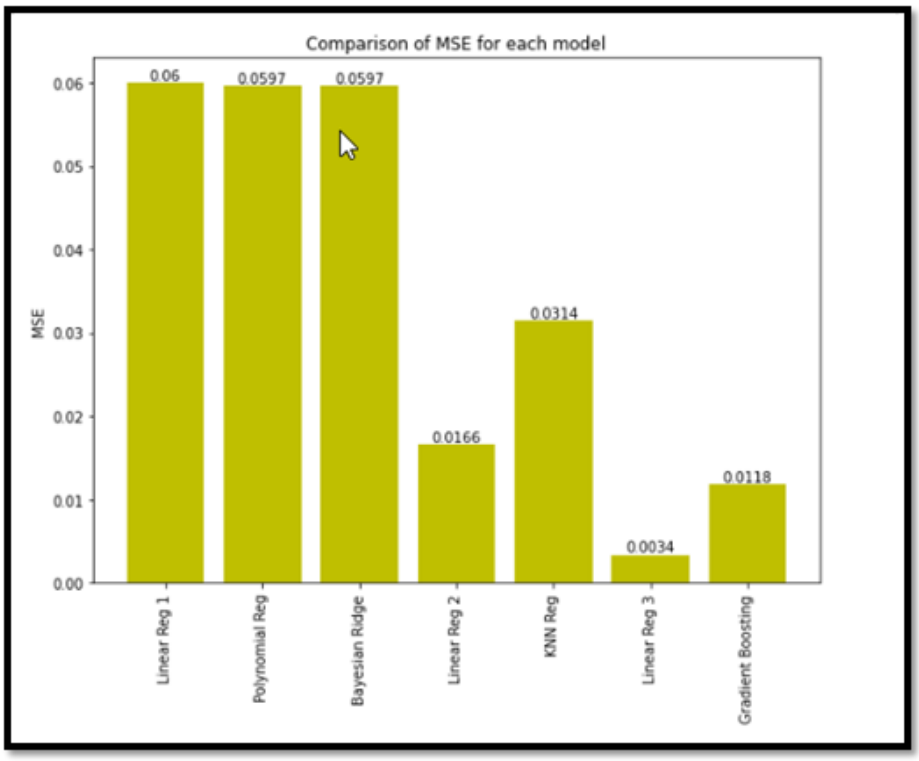
Tabular Representation of Model

Model	Machine Learning Algorithm Used	Mean Square Error	Training Accuracy	Testing Accuracy
Model 1	Linear Regression	0.06	85.9	82.62
	Polynomial Regression	0.0597	90.81	82.69
	Bayesian Ridge	0.0597	85.9	82.69
Model 2	Linear Regression	0.0166	94.67	95.37
	KNN Regressor	0.0314	94.16	91.24
Model 3	Linear Regression	0.0034	99.09	99.03
	Gradient Boosting Regressor	0.0118	99.99	96.67

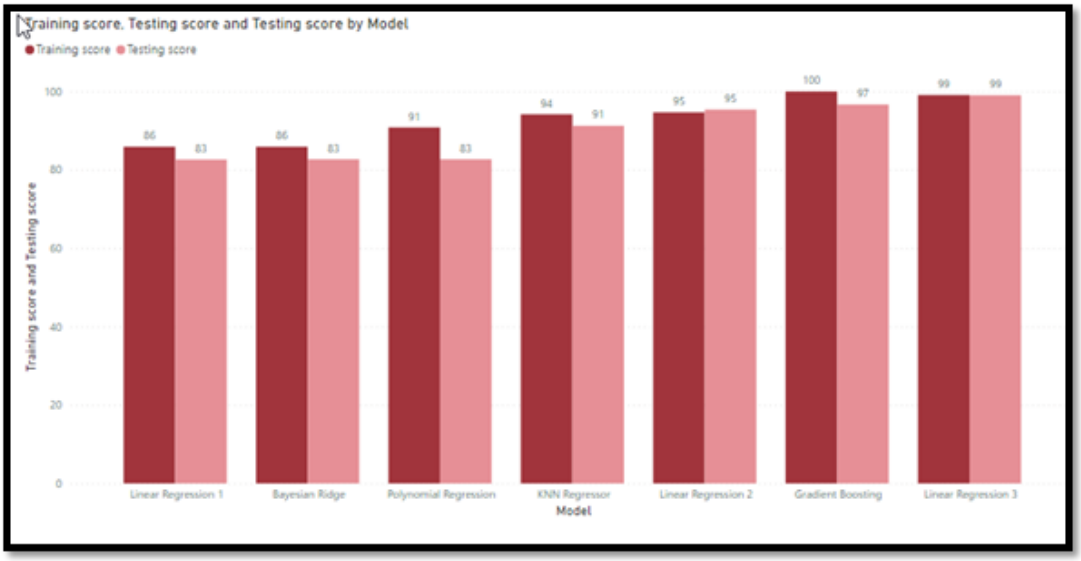
The table below shows the prediction of CGPA by the algorithm of each model.

Prediction by each Model			
Model	**Machine Learning Algorithm Used**	**Sample given for prediction**	**Predicted CGPA**
Model	Linear Regression	[1,1,4,3.7,3.4,1,1,1,1,1,3]	**2.14208**
		[4,4,4,4,4,4,4,4,4,4,4]	**3.8405**
	Polynomial Regression	[1,1,4,3.7,3.4,1,1,1,1,1,3]	**2.3432**
		[4,4,4,4,4,4,4,4,4,4,4]	**3.938**
	Bayesian Ridge	[1,1,4,3.7,3.4,1,1,1,1,1,3]	**2.1418**
		[4,4,4,4,4,4,4,4,4,4,4]	**3.8366**
Model 2	Linear Regression	[4,3.6,4,2.6,2.3,4,2.6,4,1.9,4,4,2.2,4,3.4,4,2.4,3.0,4,3.7,4,3.7,1.9]	**3.1262**
		[4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4]	**3.9193**
	KNN Regressor	[4,3.6,4,2.6,2.3,4,2.6,4,1.9,4,4,2.2,4,3.4,4,2.4,3.0,4,3.7,4,3.7,1.9]	**3.4682**
		[4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4]	**3.9126**
Model 3	Linear Regression	[4,2.7,1,3,1,2,1,3.7,4,4,1,1,1,1,4,3,2,2,2.3,2.6,1,1,1,4,4,4,4,0,0,0,2.5,3.5,4]	**2.5904**
		[4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4]	**3.9865**
	Gradient Boosting Regressor	[4,2.7,1,3,1,2,1,3.7,4,4,1,1,1,1,4,3,2,2,2.3,2.6,1,1,1,4,4,4,4,0,0,0,2.5,3.5,4]	**2.7546**
		[4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4]	**3.9717**

The following graph shows a comparison of mean square errors by the algorithms applied to each model. From the graph below, we can also conclude that the performance of the Model 3 linear regression MSE values is the best.



The clustered bar chart below provides a visual representation of the training and test accuracy of each model. It is very easy to conclude that model 3 linear regression, gives the best training and test results.



COMMENTS ABOUT PERFORMANCE OF IMPLEMENTED MACHINE LEARNING ALGORITHM

The machine learning algorithms implemented are linear regression, polynomial regression, baysian ridge, KNN regression, and gradient boosting regression.

> Linear regression works best with Model 3 compared to the other two models.

> Polynomial regression of degree 2 is implemented on model 1 which provides the best training and test accuracy. But when we use degree 3 or 4 it provides the best training accuracy, but the test accuracy begins to decline and overfits the model.

Techniques for avoiding overfitting:

- *Train more data*-Training more data helps the model identify trends in the data and make more accurate predictions.
- *Cross validation* – Divide the data sample into subsets, perform the analysis in the training set, and validate the analysis in the test set.
- *Regularization* – This is a form of regression that limits the coefficient estimates of the model to zero. This technique discourages more complex models to avoid the risk of overfitting. Two common forms of regularization are ridge regression and lasso regression.
- *Ensembling* - Ensemble is a machine learning technique for combining predictions from multiple individual models. There are several different ways to ensemble, but the two most common are:
    - Bagging attempts to reduce the possibility of overfitting complex models.
    - Boosting seeks to increase the flexibility of predicting simple models.

Therefore, to prevent model overfitting, we use powerful ensemble machine learning algorithms, the Bayesian Ridge Regression and Gradient Boosting algorithms.