Regression Analysis of Portuguese Student Performance

Dataset 2: Student performance (in Portuguese schools)

Aditi Bodhankar, bodhanka@usc.edu

May, 4th 2022

1 Abstract

In this project an end-to-end Regression based Machine Learning model is built on the Portugese Student Performance Dataset. On whole a totla of three Missions were performed:

- MISSION 1: Predict first-period academic performance without any prior academic performance data.
- MISSION 2: Predict final-period academic performance without any prior academic performance data.
- MISSION 3: Predict final academic performance using all available prior academic performance data.

Firstly, a detailed Exploratory data analysis is conducted along with Feature selection A brief, informative description of your project. Include the problem, dataset(s) you used, approach (naming the machine learning methods you used and how you compared them), and key results (be sure to include at least your best result and the method with which you obtained it, for each dataset).

2 Introduction

2.1 Problem Assessment and Goals

To begin with, the dataset consists of 30 features plus the three grades 'G1', 'G2', 'G3'. The following are the input features:

- school -student's school ('GP' -Gabriel Pereira or 'MS' -Mousinho da Silveira)
- sex -student's sex ('F' -female or 'M' -male)
- age -student's age (15-22)
- address -student's home address type ('U' -urban or 'R' -rural)
- famsize -family size ('LE3' -less or equal to 3 or 'GT3' -greater than 3)
- Pstatus -parent's cohabitation status ('T' -living together or 'A' -apart)
- Medu -mother's education (0 -none, 1 -primary education (4th grade), 2 -5th to 9th grade, 3 secondary education or 4 higher education)
- Fedu -father's education (0 -none, 1 -primary education (4th grade), 2 -5th to 9th grade, 3 -secondary education or 4 -higher education)
- Mjob -mother's job ('teacher', 'health' care related, civil 'services', 'at home' or 'other')

- Fjob -father's job ('teacher', 'health' care related, civil 'services', 'at home' or 'other')
- reason -reason to choose this school (close to 'home', school 'reputation', 'course' preference or 'other')
- guardian -student's guardian ('mother', 'father' or 'other')
- traveltime -home to school travel time (1 -¡15 min., 2 -15 to 30 min.,3 -30 min. to 1 hour, or 4 -¿1 hour)
- studytime -weekly study time (1 -¡2 hours, 2 -2 to 5 hours, 3 -5 to 10 hours, or 4 -¿10 hours)
- failures -number of past class failures (n if 1¡=n¡3, else 4)
- schoolsup -extra educational support (yes or no)
- famsup -family educational support (yes or no)
- paid -extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
- activities -extra-curricular activities (yes or no)
- nursery -attended nursery school (yes or no)
- higher -wants to take higher education (yes or no)
- internet -Internet access at home (yes or no)
- romantic -with a romantic relationship (yes or no)
- famrel -quality of family relationships (from 1 -very bad to 5 -excellent)
- freetime -free time after school (from 1 -very low to 5 -very high)
- goout -going out with friends (from 1 -very low to 5 -very high)
- Dalc -workday alcohol consumption (from 1 -very low to 5 -very high)
- Walc -weekend alcohol consumption (from 1 -very low to 5 -very high)
- health -current health status (from 1 -very bad to 5 -very good)
- absences -number of school absences (from 0 to 93)

After loading the data into pandas dataframes, train and test sets are created which are later used for feature selection. We first begin with the visualizations of the features. For this, the Seaborn and MAtplotlib library was used to compute histograms, KDE plots and bar plots to visualize the behaviour of the different features. Additionally, In the case of feature Selection, five different techniques were thoroughly studied using the scikit-learn library.

- Pearson Correlation Technique
- Univariate Feature Selection using F-score
- Recursive Feature Elimination
- Recursive Feature Elimination, Cross Validated Feature Selection
- Mutual Information Feature Selection

Then for the model selection, an initial baseline model were implemented which consisted of Linear Regression, 1-Nearest Neighbor, Trivial Regression. Additionally, a Support Vector Regression, k-Nearest Neighbor Regression, an Artificial Neural Network and Ridge Regression were also implemented.

For the Performance Metrics, r2 score, Root Mean Square Error, Mean Absolute Error were computed.

3 Approach and Implementation

First of all the dataset was loaded using **pandas** into dataframes. Then a thorough data preprocessing was conducted. This was followed by Data visualizations and some exploratory data analysis. In the section, histograms for the three grades outputs were plotted to visualize their distribution. Again a set of KDE distribution and histograms for the actual Grades were plotted. additionally, to visualize the features, a simple pie plot and barplots were plotted. All these plots are presented in the **Preprocessing** section. For the plots, both Matplotlib and Seaborn were used.

Three baseline models were implemented. The Trivial Regressor, One-Nearest Neighbor were implemented without using sklearn (by self) and for the linear Regression, sklearn's Linear Regression was used.

Secondly, once the data was loaded and preprocessed, extensive feature selection was conducted to select certain important features that contributed to the training without overfitting the training data. Four different feature selection techniques were implemented.

- **Pearson Correlation Technique**: Correlation between each feature and the output target is computed. and the ones with smallest correlations are dropped.
- **Univariate Feature Selection using F-score**: We take 'k' features at a time and run a regression to compute the f-statistic, where k-ranges from 1 to D (total number of features) and choose the features based on ranking
- **Recursive Feature Elimination**: This technique takes a subset of the features and fits it with the ML algorithm inbuilt in the core of the function and discards the features which do not contribute to best fit of the model.
- **Recursive Feature Elimination, Cross Validated Feature Selection**: This technique calculates the cross validation score for each of the subset of features.
- **Mutual Information Feature Selection**: Here, we measure the uncertainty in one variable keeping the other variable constant. The mutual information of any feature, if too low, is removed.

To conduct this feature selection, necessary sklearn libraries and functions were used as mentioned above.

The technique Recursive Feature Selection resulted in better set of selected features and for the model selection, this reduced feature set was used.

After the feature selection, for the model selection, 4 different models were cross validated and the distributions for predictions for each of the missions was plotted. The four models are as follows:

- Support Vector Regression - SVR
- MLP Regression - MLPRegressor
- Ridge Regression - Ridge
- KNN - KNeighborsRegressor

For the cross validation, the sklearn's function - **cross_validate** was used which evaluates the model and gives, validation score ($'test\_score'$), along with the train_score and the corresponding estimator.

For this, a pipeline was created using the sklearn's 'make_pipeline' function, which scaled the features using StandardScalar() necessarily before training the model. The model selection was done using cross validation and the optimal estimator/model was then used to test the data.

3.1 Dataset Usage

The given testing set was used for testing and for the validation and the training sets, the original training set was used. For the visualization, the train dataset is divided into features and targets (grades). When conducting Feature Selection, the train data is imputed with the target to compute the Pearson correlations as mentioned above. Later for the purpose of training the model and testing it, three new train sets and targets were created for each mission.

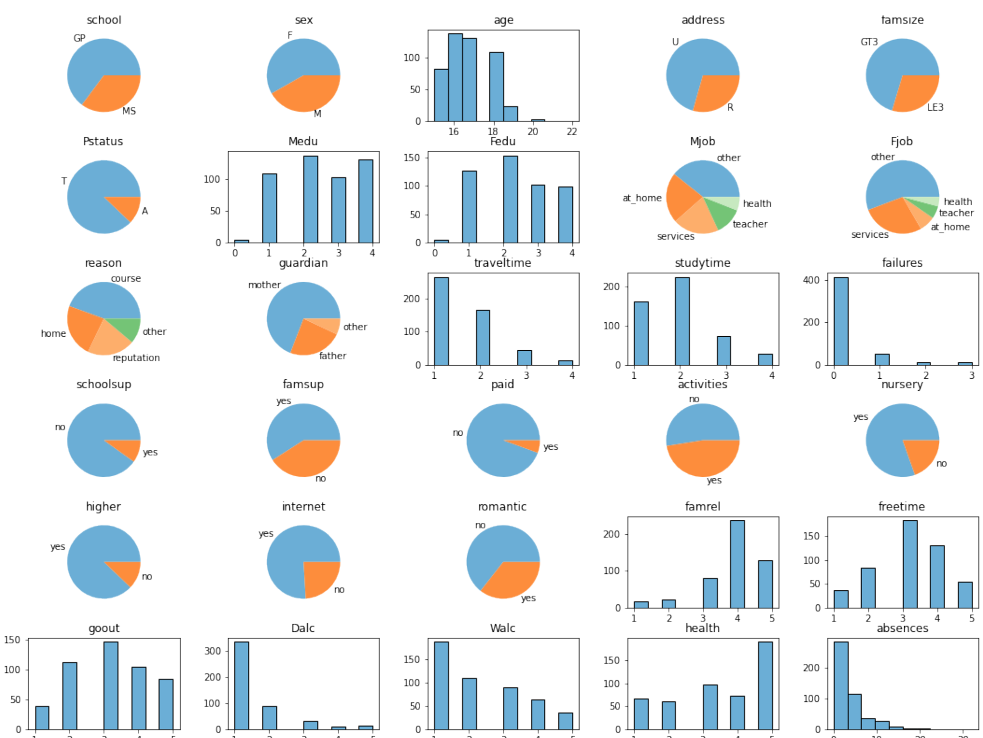
For cross validation, k-fold CV approach is implemented while using the 'cross_validate' method of the sklearn library. In this the training set is divided into 'k' parts and one of those parts is treated as the validation set and a validation score is computed for this split in training set. Such splits are run until each part serves as a validation set. Finally, the average of all the 'k' validation scores is finally computed to estimate the performance of the model. Now this cross validation is computed for a few number of runs corresponding to the different choices of model hyperparameters.The high validation score corresponding to a particular hyperparameter of the model, gives the most optimized model which is later used on the testing dataset.

Finally, the test set was used three different times. First during baseline model evaluations. Second, during feature selection and third while testing the model after model selection.

3.2 Preprocessing

In the datapreprocessing, initially the data, both train and test sets were loaded into dataframes using pandas library. Next, the binary features were converted into numeric by assigning values 1's and 0's and the categorical/nominal features were one hot encoded using the sklearn function, 'preprocessing.OneHotEncoder'. This can be seen in the method 'categorical_to_numeric' from the 'Data_Preprocessing' class. Once the nominal features are one-hot encoded, the original features are dropped from the dataframes.

The features are visualized as follows;



Additionally, the dataset is also split into features and labels for ease of computation in the later use.

As a part of Data Visualization, the following histograms and distribution plots were computed for the three grade targets - G1 (first period grade), G2 (second period grade), G3 (final grade) respectively. Additionally, I have calculated the KDE plots for the three grades target as well. Which are plotted below after the histograms.

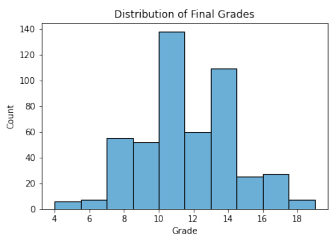


Figure 1: G1

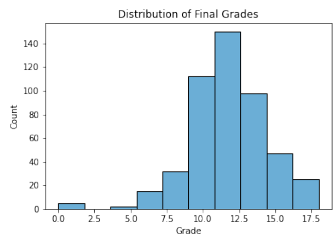


Figure 2: G2

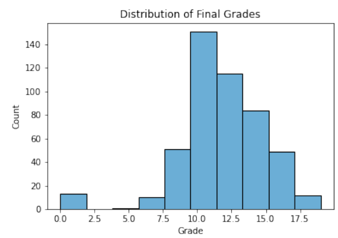
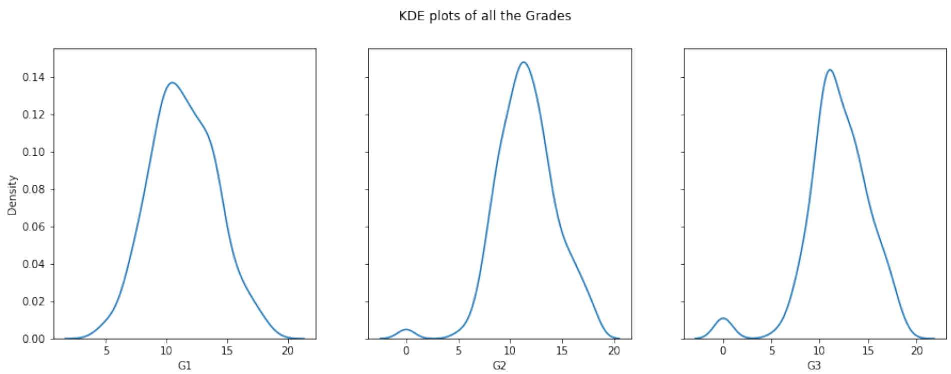


Figure 3: G3



3.3 Feature dimensionality adjustment (Feature Selection)

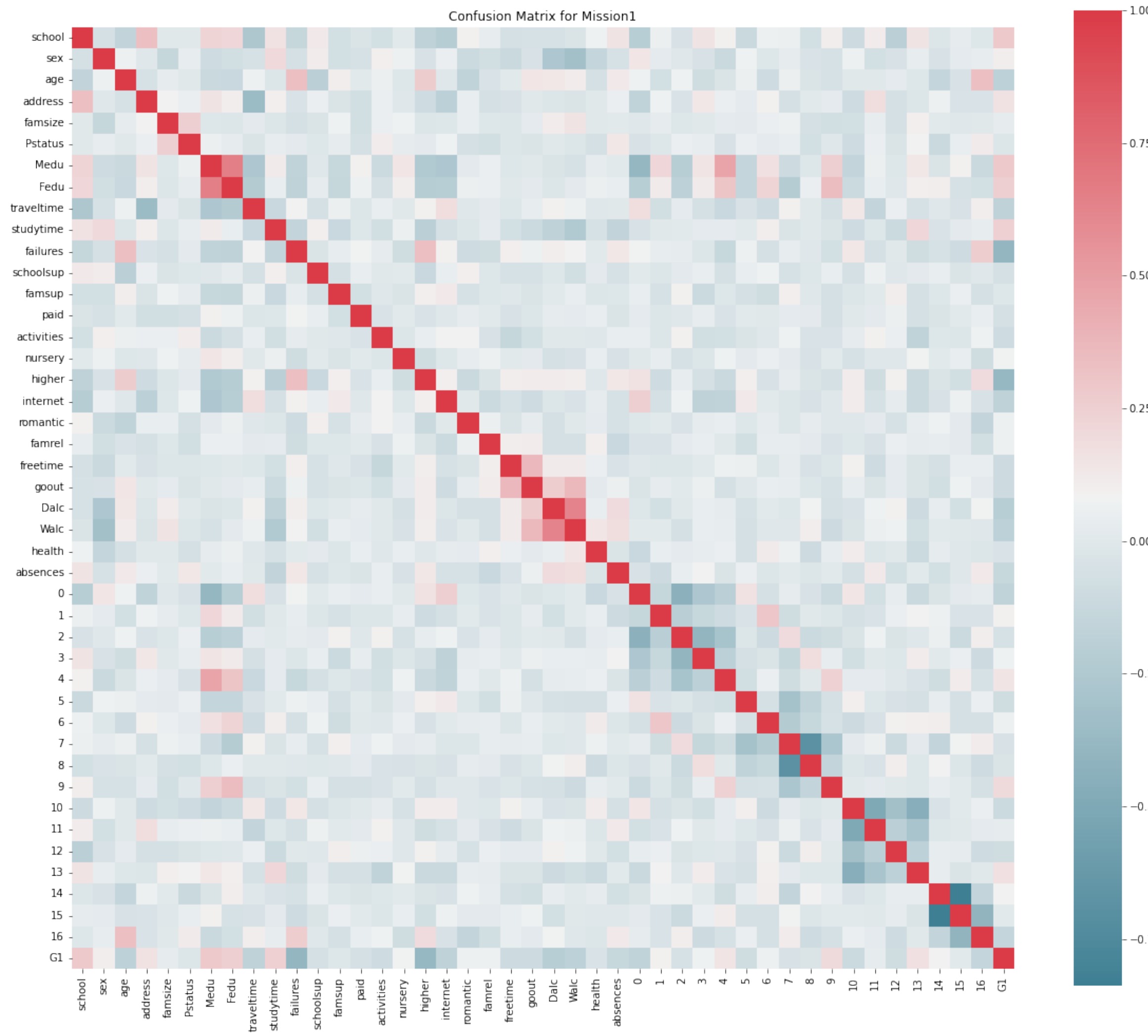
3.3.1 Pearson Correlation Feature Selection

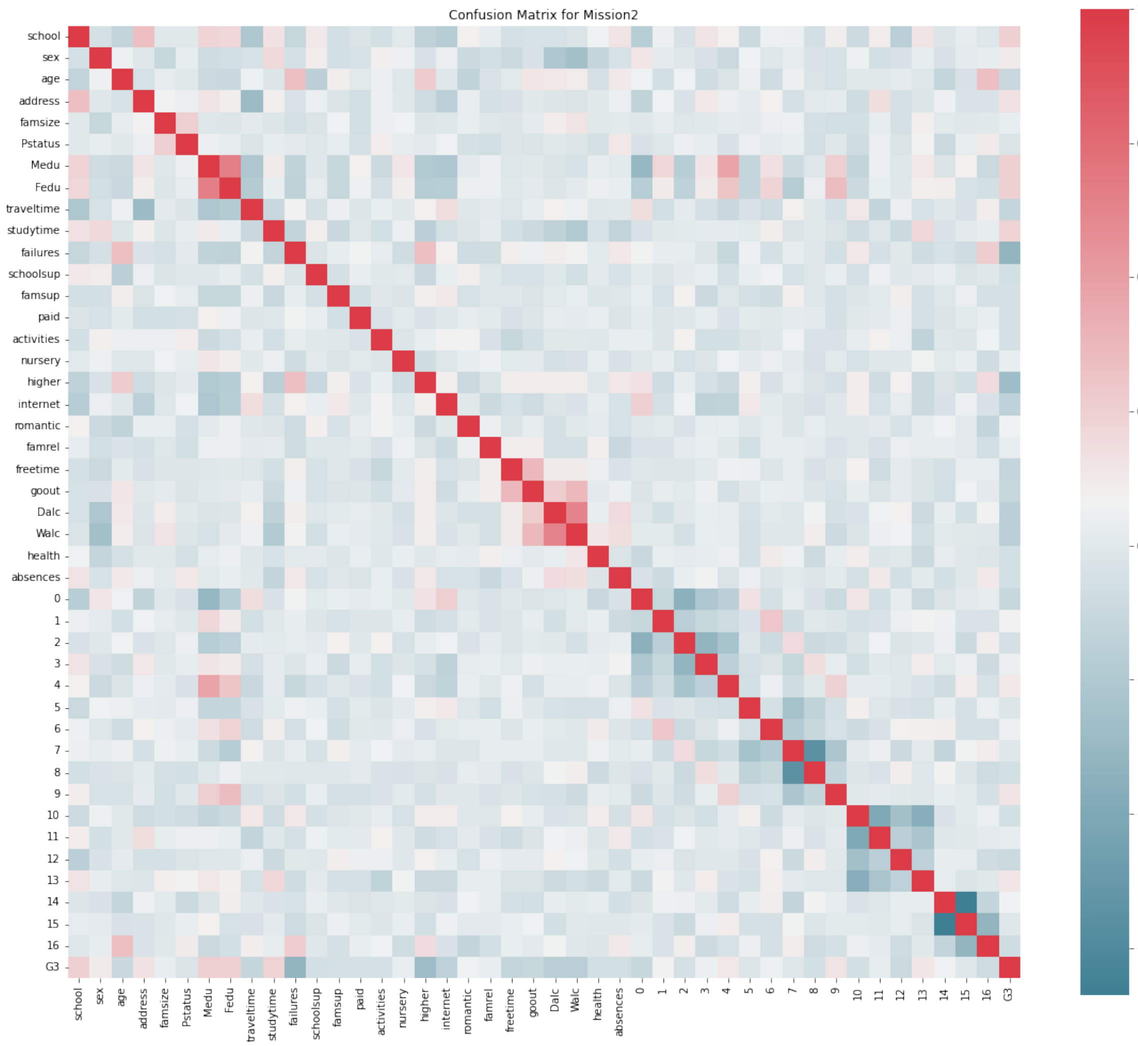
For the feature selection as said, I have implemented 5 different techniques. For the first technique - Pearson Correlation, I have computed correlation matrix for the three missions which are as follows. From the correlation matrices, features that are less correlated to the target variable are dropped. For this, the function - 'correlation_matrix(df, m)' and 'get_correlated_columns(df, c)' calculates the correlated matrix and plots it and the latter function, finds features which have correlations less than the threshold 'c' (0.1) and drops them. The resultant features are as follows. The threshold was chosen to be this low, because of the already lesser correlation of the existing features and also inorder to avoid overfitting the training model.

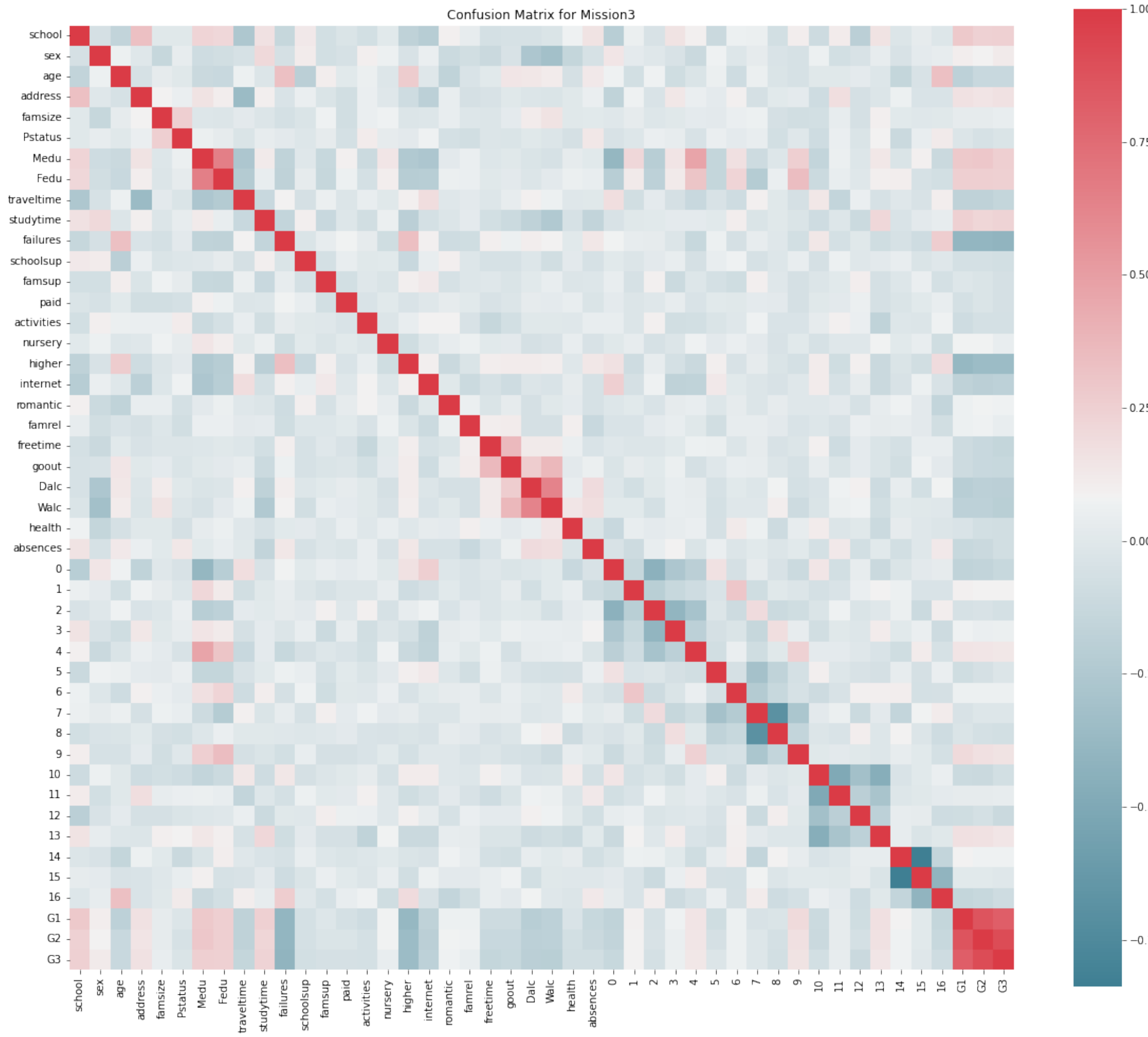
Selected Features for the three Missions
Number of correlated features_mission1: 22
correlated features for X1: 'schoolsup', 1, 2, 3, 5, 6, 7, 8, 11, 'nursery', 12, 14, 'famsize', 15, 'paid', 'famrel', 'activities', 'romantic', 'goout', 'health', 'famsup', 'Pstatus' Number of correlated features_mission2: 23
correlated features for X2: 'schoolsup', 1, 2, 3, 5, 6, 7, 8, 10, 11, 'nursery', 14, 'famsize', 15, 16, 'paid', 'absences', 'famrel', 'activities', 'romantic', 'health', 'famsup', 'Pstatus' Number of correlated features_mission3: 23
correlated features for X3: 'schoolsup', 1, 2, 3, 5, 6, 7, 8, 10, 11, 'nursery', 14, 'famsize', 15, 16,

'paid', 'absences', 'famrel', 'activities', 'romantic', 'health', 'famsup', 'Pstatus'

The integer indices here corresponds to the on-hot encoding of the four nominal features. Through this, we can see that only a certain categories of the nominal features were selected.




Confusion Matrix for Mission2
school
sex
age
address
famsize
Pstatus
Medu
Fedu
traveltime
studytime
failures
schoolsup
famsup
paid
activities
nursery
higher
internet
romantic
famrel
freetime
goout
Dalc
Walc
health
absences
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
G3
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75



3.3.2 Univariate Feature Selection-Fscore

As presented before, here I use the 'f_regressor' method from the sklearn library to compute the f_score based selected features. Here are the expected plots;

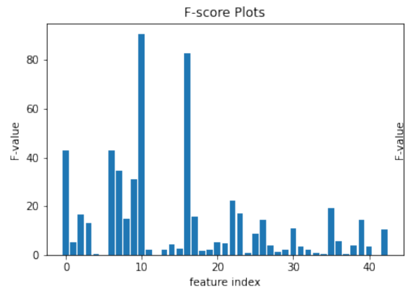


Figure 4: Mission1

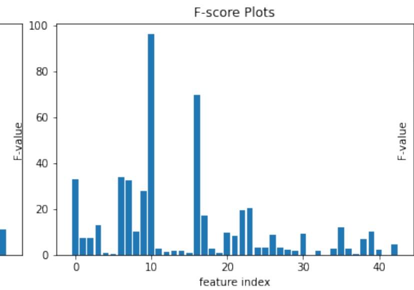


Figure 5: Mission2

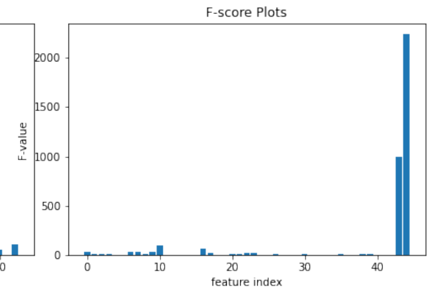


Figure 6: Mission3

The features with 25 best F-scores are selected after evaluating this technique.The selected features are as follows:

MISSION 1:
'failures', 'higher', 'Medu', 'school', 'Fedu', 'studytime','Dalc',9, 'Walc', 'age', 'internet', 'traveltime', 13, 0, 'address',4, 16, 'absences', 10, 'sex', 'freetime', 'goout', 'activities', 1, 12

MISSION 2:
'failures', 'higher', 'Medu', 'school', 'Fedu', 'studytime','Dalc', 'Walc', 'internet', 'address', 9, 13, 'traveltime', 'freetime', 4, 0, 'goout', 'age', 'sex', 12, 16, 1, 'health', 'absences', 'schoolsup'

MISSION 3:
'G2', 'G1', 'failures', 'higher', 'Medu', 'school', 'Fedu', 'studytime', 'Walc', 'Dalc', 'internet', 'address', 9, 13 'traveltime', 'freetime', 4, 0, 'goout','age','sex',12,16,1,'health'

3.3.3 Recursive Feature Elimination

As presented above, for the recursive feature Elimination, I have used the RFE method from the sklearn library. The following are the selected features:

MISSION 1:
'school', '1', '2', '3', '4', '5', '6', '0', '7', '9', '10', '11', '12', '13', '14', '8', '15', '16', 'schoolsup', 'higher', 'failures', 'sex', 'studytime', 'internet', 'famsup', 'Dalc', 'address', 'activities', 'paid', 'Medu', 'Pstatus', 'famsize', 'romantic', 'Fedu', 'famrel', 'health', 'traveltime', 'freetime', 'Walc', 'age', 'absences', 'goout', 'nursery'

MISSION 2:
'school', '1', '2', '3', '4', '5', '6', '0', '7', '9', '10', '11', '12', '13', '14', '8', '15', '16', 'schoolsup', 'higher', 'failures', 'sex', 'internet', 'studytime', 'address', 'Pstatus', 'paid', 'Fedu', 'famsize', 'Walc', 'freetime', 'nursery', 'romantic', 'activities', 'famsup', 'health', 'Dalc', 'goout', 'age', 'famrel', 'Medu', 'traveltime', 'absences'

MISSION 3:
'school', '16', '15', '12', '11', '10', '9', '8', '7', '5', '2', '1', 'higher', 'nursery', 'G2', 'paid', 'sex', 'address', 'Pstatus', 'traveltime', 'failures', 'schoolsup', '3', '4', '14', '0', '13', '6', 'Medu', 'Fedu', 'G1', 'famsup', 'freetime', 'activities', 'famsize', 'Walc', 'internet', 'studytime', 'famrel', 'health', 'age', 'romantic', 'absences', 'goout', 'Dalc'

3.3.4 Recursive Feature Selection Cross Validated

Here we use the RFECV method from the sklearn's 'feature_selection' module, the cross correlation scores for all the subset of features is computed and plotted as below:

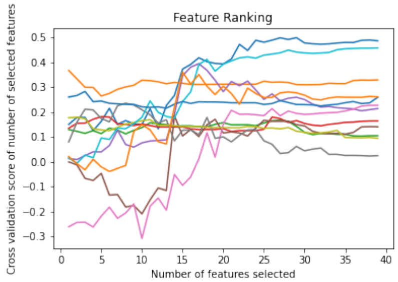


Figure 7: Mission1

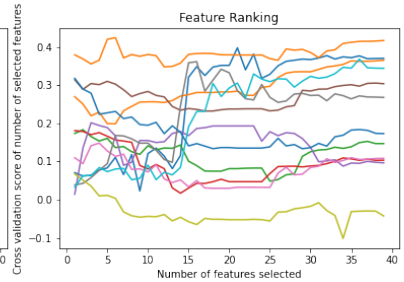


Figure 8: Mission2

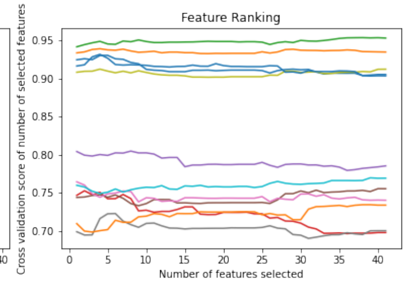


Figure 9: Mission3

From these plots, we see that the subset of features corresponding to the highest cross validation score is the set of selected features obtained from this technique. For the first mission, the cross validation plots are more clear and consistent. With increasing features, the score reaches a maximum saturation. This trend is seen even in the case of Mission 2. But in the case of Mission 3, since, 'G1', 'G2' correspond the most to the target variable, which can be clearly seen from the f-score plots previously, the cross validation score does not seem to change much even after adding more features than necessary.

Therefore, the final selected features are as follows;

MISSION 1:
'school', 'sex', 'address', 'Medu', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'higher', 'internet', 'Dalc', 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

MISSION 2:
'school', 'sex', 'age', 'address', 'famsize', 'Pstatus', 'Fedu', 'studytime', 'failures', 'schoolsup', 'famsup', 'paid', 'activities', 'nursery', 'higher', 'internet', 'romantic', 'famrel', 'freetime', 'goout', 'Dalc', 'Walc', 'health', 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

MISSION 3:
'failures', 'schoolsup', 'paid', 'higher', 5, 12, 16, 'G2'

3.3.5 Mutual Information Feature Selection

The mutual information of one feature by keeping the others constant, gives a good information about which features are predominantly contributing to the target variable. The following are the plots. In the Mission 3, it is clear that the prior grades contributed the most to the target variable.

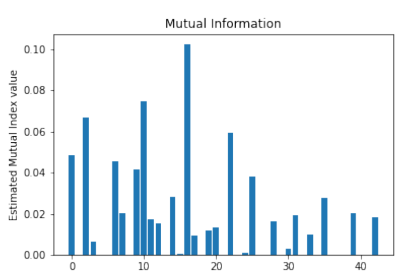


Figure 10: Mission1

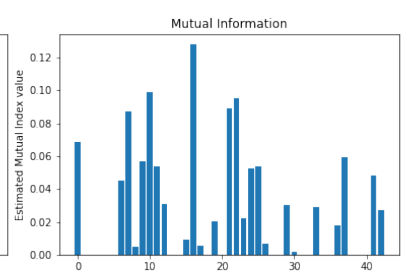


Figure 11: Mission2

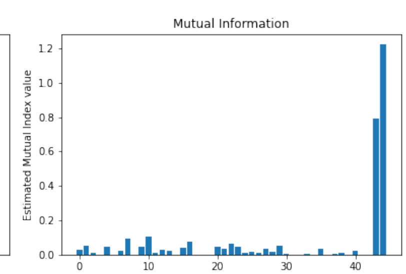


Figure 12: Mission3

As seen from the plot 3, for the mission 3, the feature 'G2' contributes the highest and which is also justified by the selected features obtained from the previous techniques.

The following are the selected features obtained from this particular technique;

MISSION 1:
'higher', 'failures', 'age', 'Dalc', 'school', 'Medu', 'studytime', 'absences', 'activities', '9', 'Fedu', '13', '5', '16', 'schoolsup', '2', 'famsup', 'freetime', 'famrel', '7', 'internet', 'address', '4', 'health', 'nursery'

MISSION 2:
'higher', 'failures', 'Dalc', 'goout', 'Fedu', 'school', '11', 'studytime', 'absences', 'schoolsup', 'health', '15', 'Medu', 'famsup', '3', '7', '16', 'Walc', 'famrel', '10', 'nursery', '0', 'internet', 'traveltime', '4'

MISSION 3:
'G2', 'G1', 'failures', 'Fedu', 'higher', 'Dalc', '3', 'sex', 'Walc', 'freetime', 'studytime', 'famsize', 'nursery', 'goout', '1', '9', 'school', 'famsup', '14', 'paid', 'Medu', 'absences', '2', '12', 'schoolsup'

In the next stage, I have selected features from Recursive Feature Elimination as it resulted in better results.

3.4 Training, Classification or Regression, and Model Selection

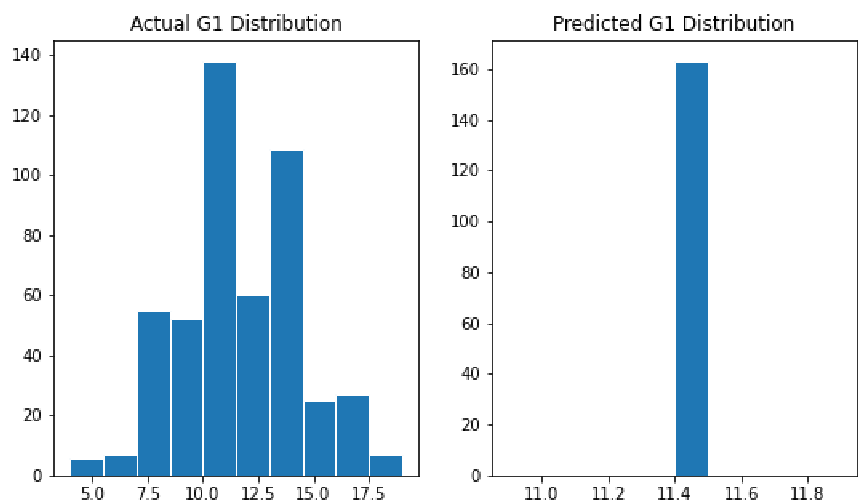
3.4.1 Baseline Models

Three Baseline models are evaluated. The Trivial Regression function was written without the help of sklearn and is evaluated over all the 3 missions as stated above.

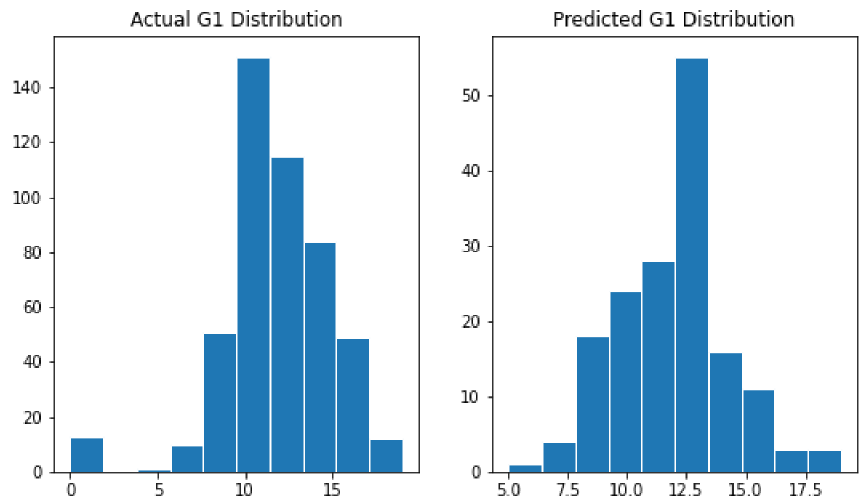
The following are the results for **MISSION 1**

Model	RMSE	MAE	R2
Trivial Regression	2.8404	2.2441	-2.0225e-08
One-Nearest-Neighbor	3.3907	2.4908	-0.4250
Linear Regression	2.4208	1.8632	0.2736

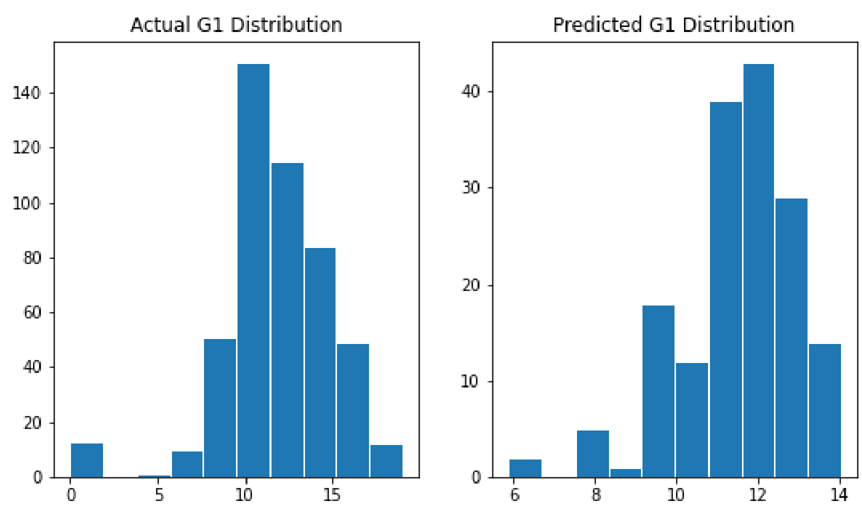
The Histogram plots for the actual and predicted grades for the Trivial Regression is as follows:



For the 1-NN regressor:



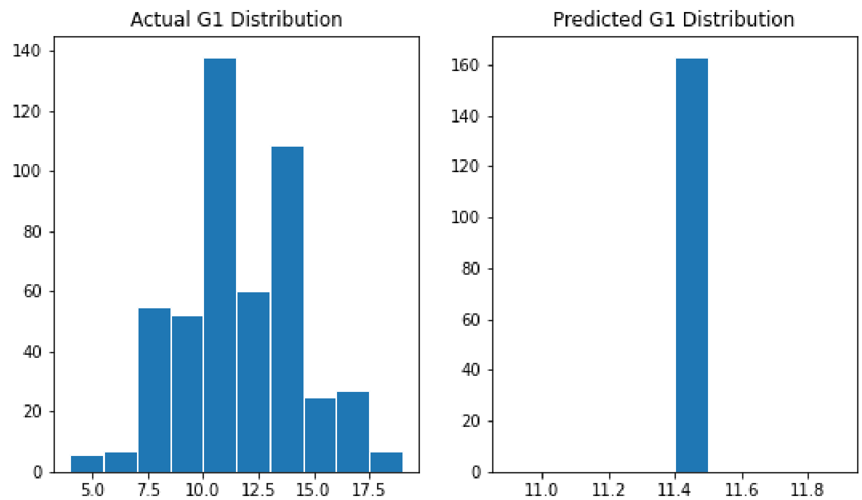
For Linear Regressor:



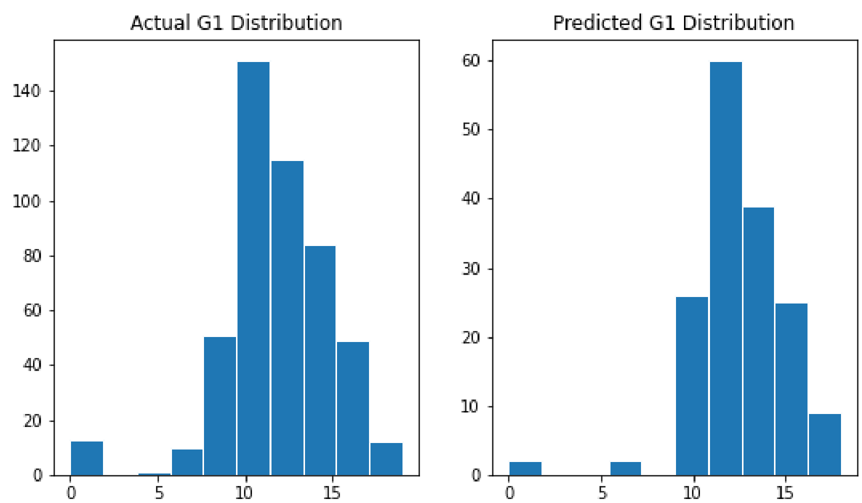
The following are the results for **MISSION 2**

Model	RMSE	MAE	R2
Trivial Regression	3.2209	2.5080	-0.0295
One-Nearest-Neighbor	3.8507	2.7546	-0.4715
Linear Regression	2.7124	2.0416	0.2699

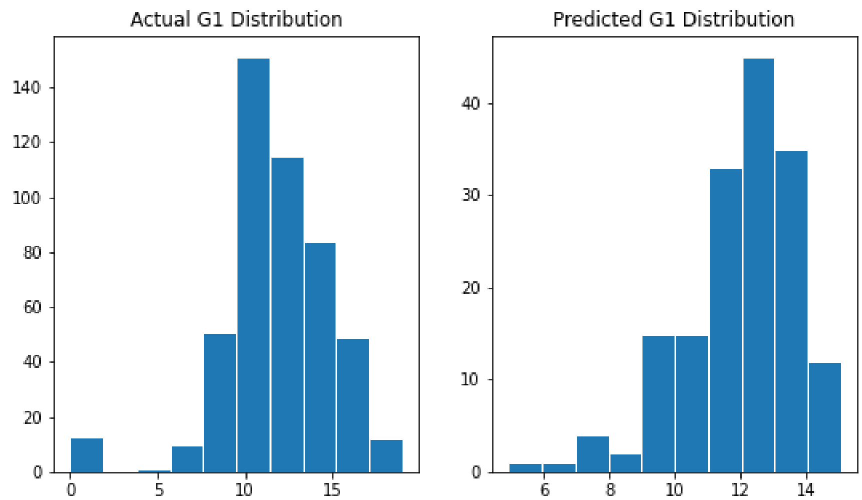
The Histogram plots for the actual and predicted grades for the Trivial Regression is as follows:



For the 1-NN regressor:



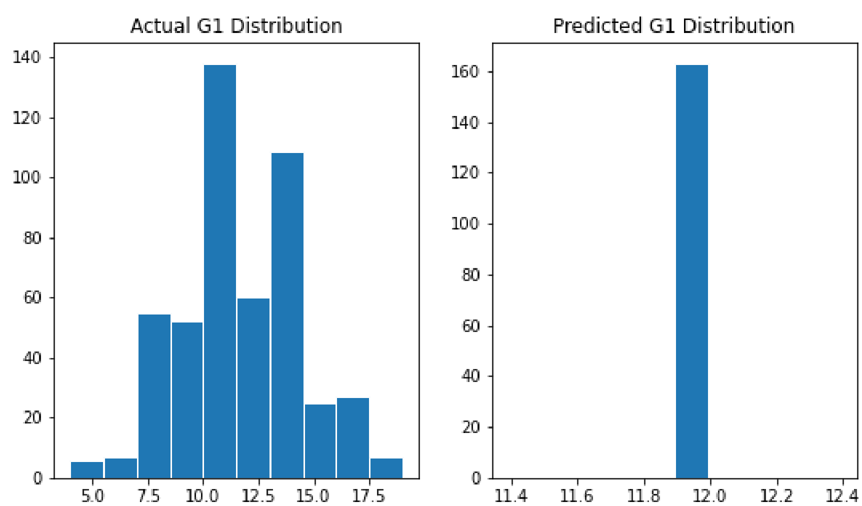
For Linear Regressor:



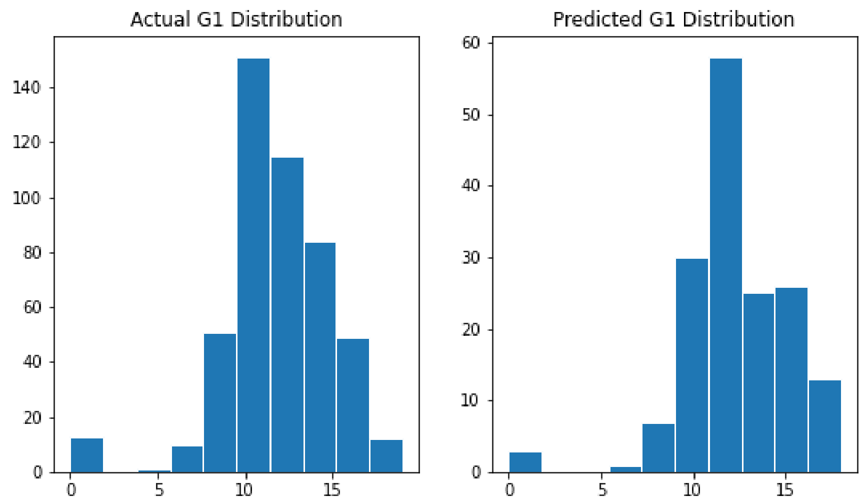
The following are the results for **MISSION 3**

Model	RMSE	MAE	R2
Trivial Regression	3.1748	2.4686	-0.0003
One-Nearest-Neighbor	1.6166	1.1043	0.7407
Linear Regression	1.0136	0.7168	0.8980

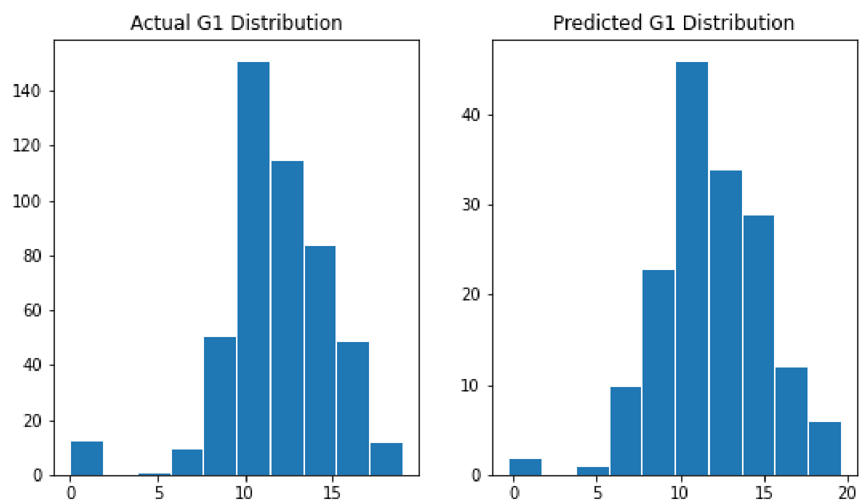
The Histogram plots for the actual and predicted grades for the Trivial Regression is as follows:



For the 1-NN regressor:



For Linear Regressor:



3.4.2 Model Selection - k-Nearest Neighbor Regression

The results obtained for the KNN regressor are as follows. The number of neighbors 'k' parameter was tuned for different values before finding the optimized estimator.

The train and the validation scores as as follows:

Figure 13: Mission1

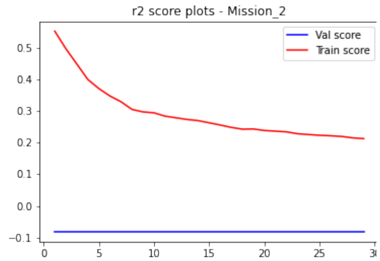


Figure 14: Mission2

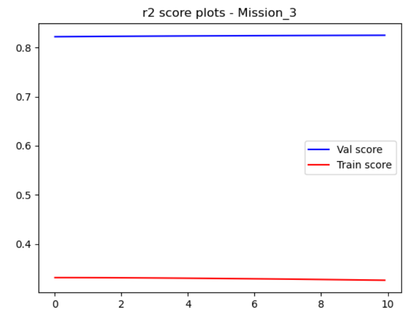
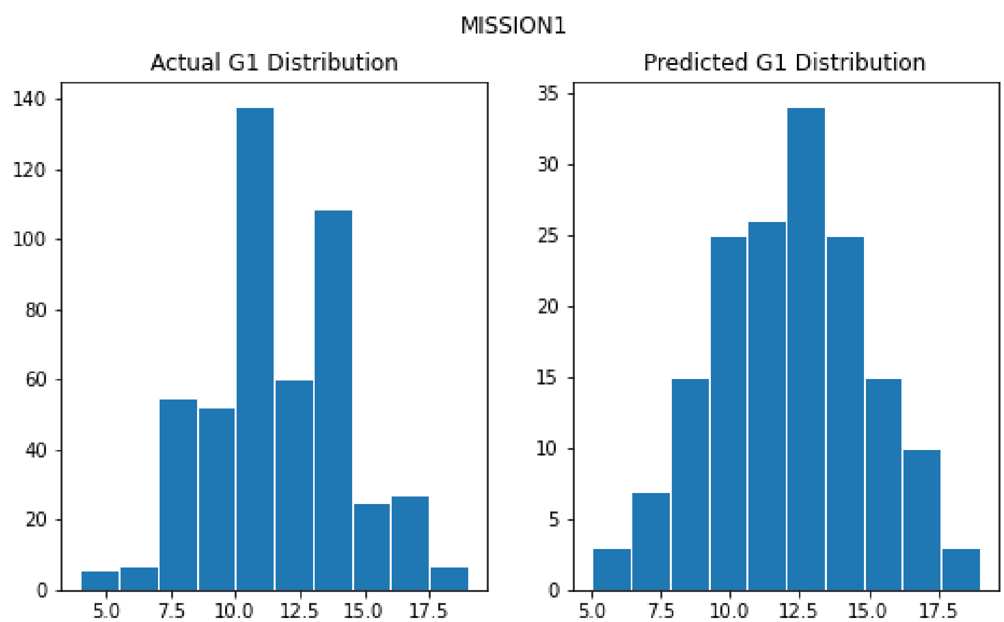
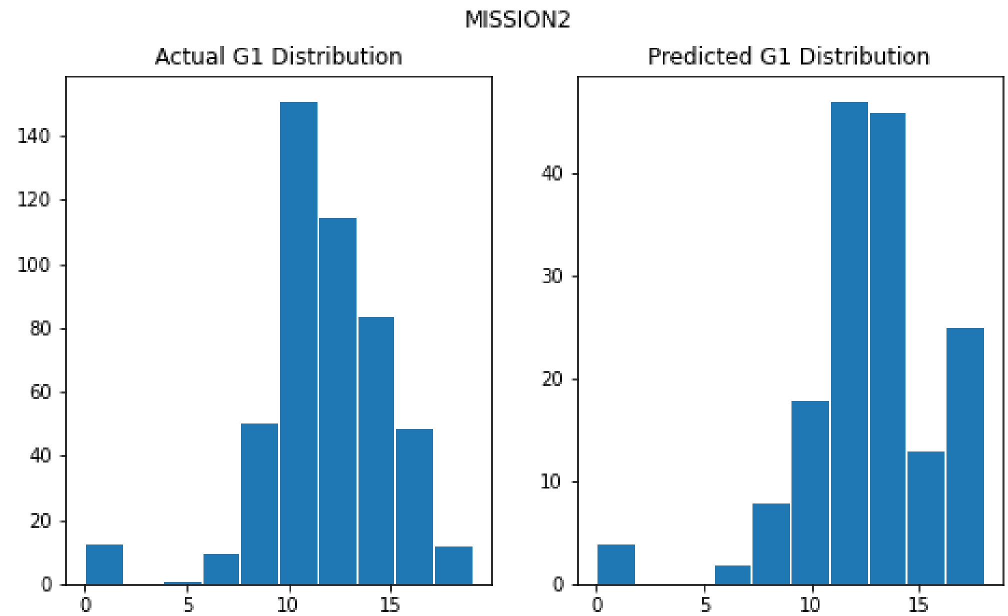


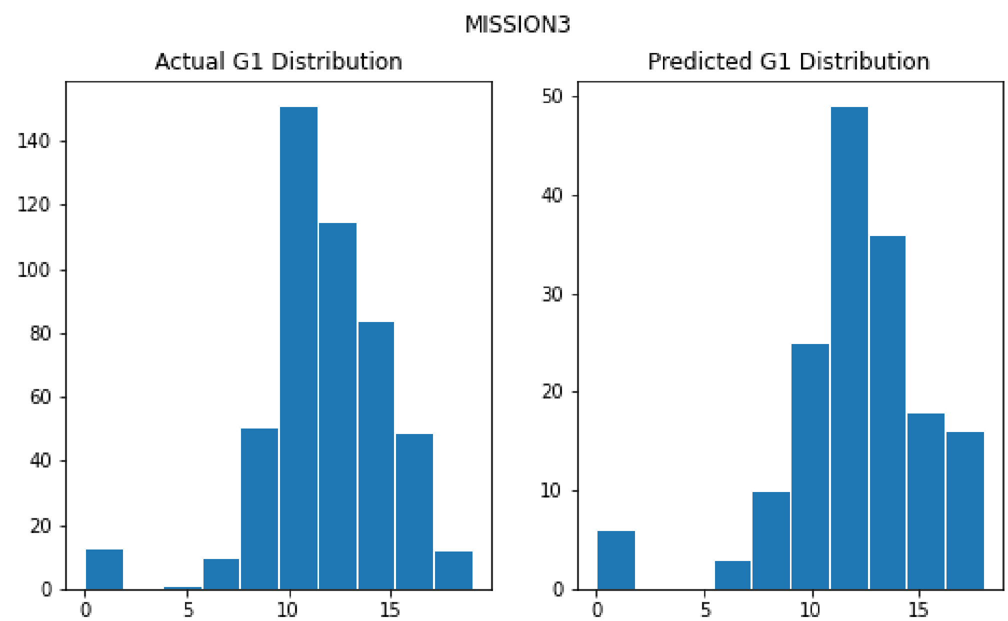
Figure 15: Mission3

The final results for this regressor is as follows:

Model	RMSE	MAE	R2
Mission1	3.5713	2.7546	-0.5809
Mission2	4.0069	3.0368	-0.5933
Mission3	1.5646	0.9509	0.7571







The weights used in this case are 'uniform'. For the metric, a more generalized euclidean distance called the 'Minkowski's' distance metric was used in the kernel computations. The number of neighbors was the hyperparameter which was tuned using cross validation approach to find the best model. As seen even with the best features selected from the feature selection, there has been no significant improvement on the test set compared to the best baseline model.

3.4.3 Model Selection - Support Vector Regression

The results obtained for the Support Vector Regressor are as follows. The number of neighbors 'k' parameter was tuned for different values before finding the optimized estimator.

The train and the validation scores as as follows:

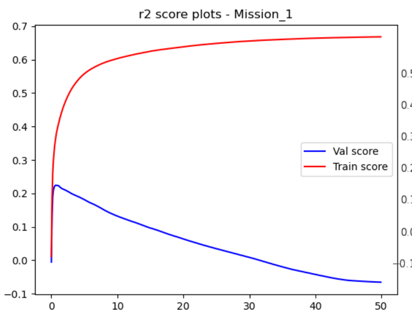


Figure 16: Mission1

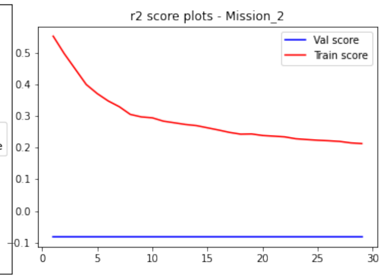


Figure 17: Mission2

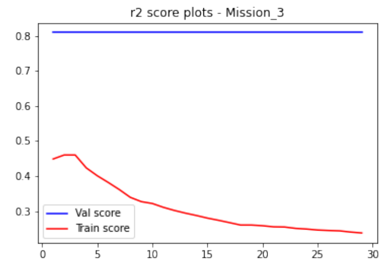
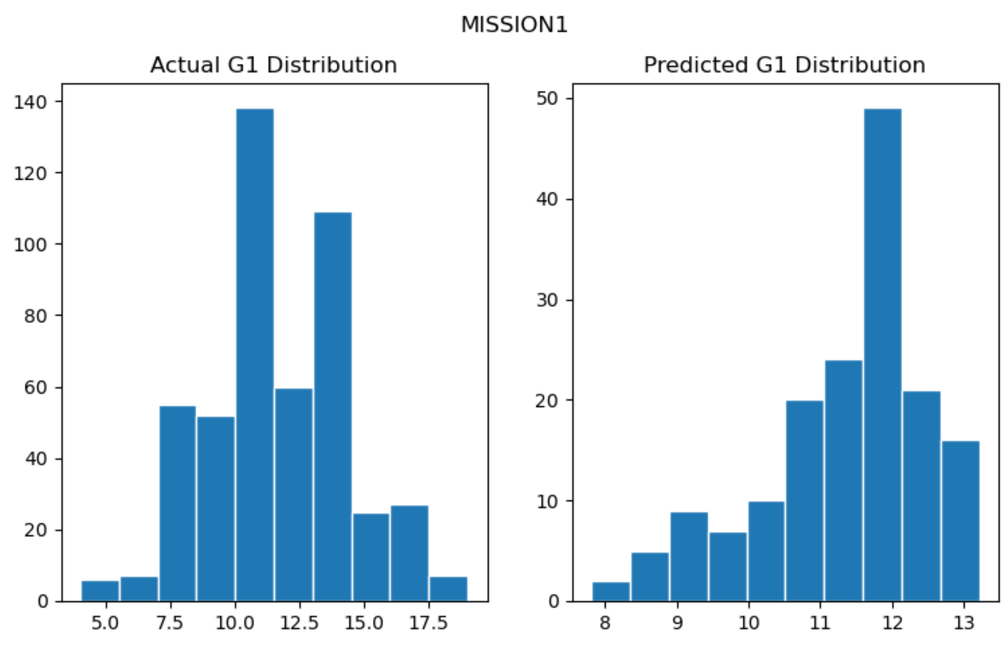
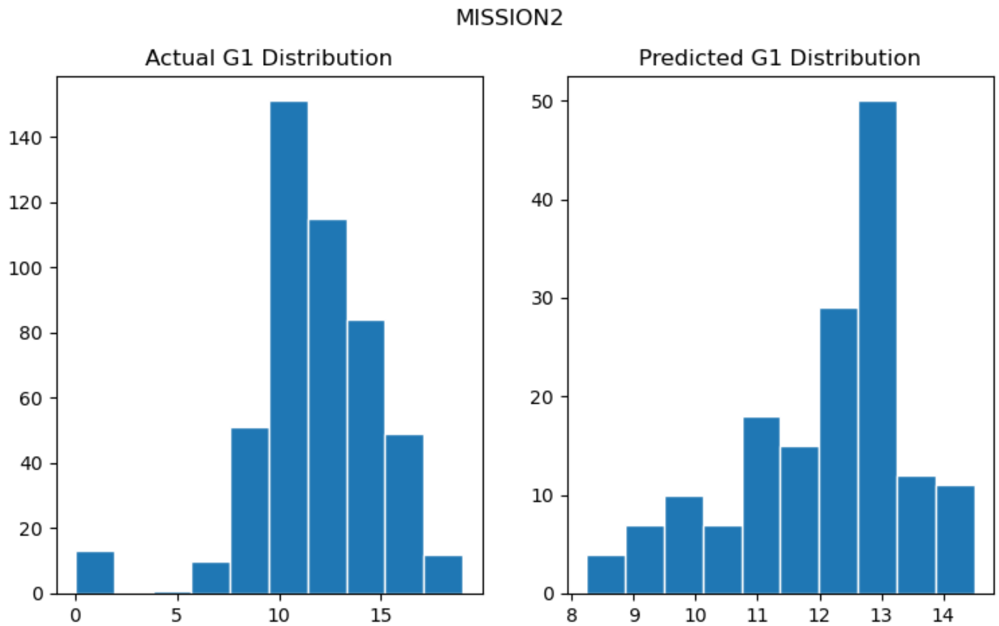


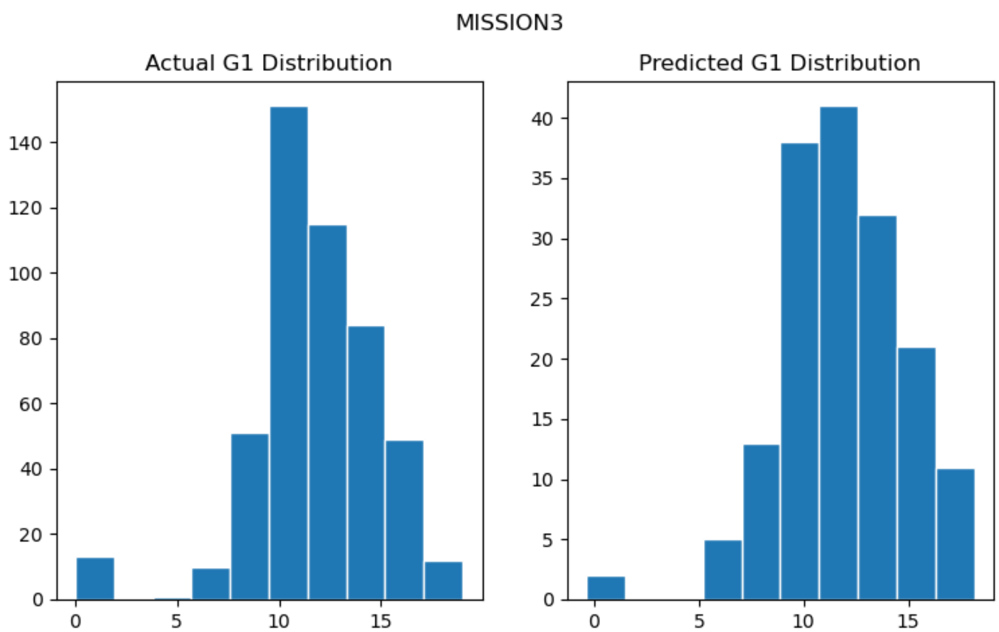
Figure 18: Mission3

The final results for this regressor is as follows:

Model	RMSE	MAE	R2
Mission1	2.4230	1.8368	0.2681
Mission2	2.6993	2.0171	0.2769
Mission3	1.0058	0.7225	0.8996







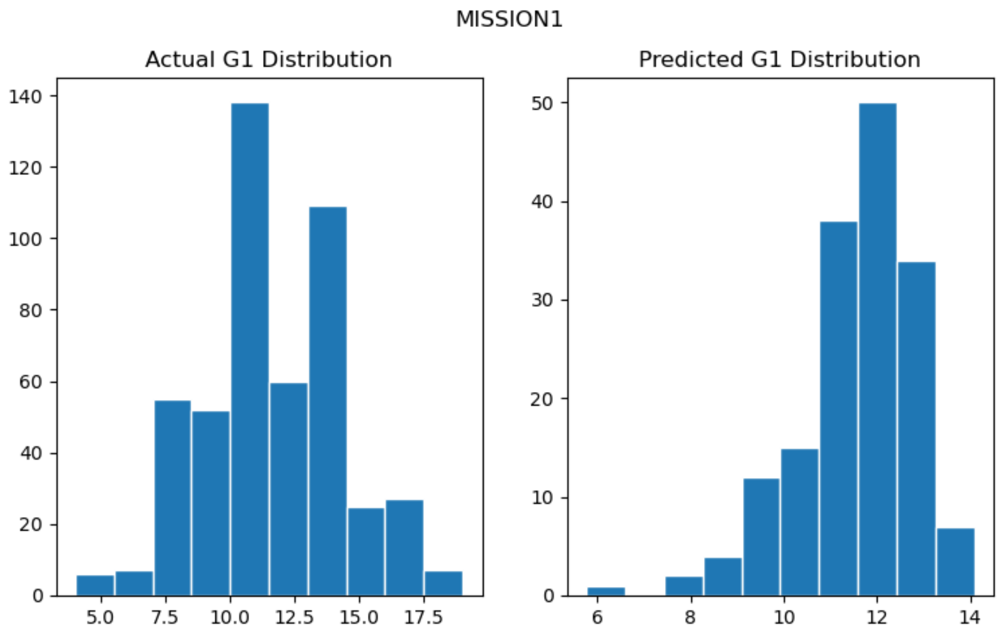
In this case the regularizer term, 'C' in the SVR model of sjlearn was tuned to find the best model. The kernel function used in this case was the Radial Basis Function.The optimized C values for Mission 1 was 0.71, Mission 2 was 1.11 and for Mission 3 was 7.41

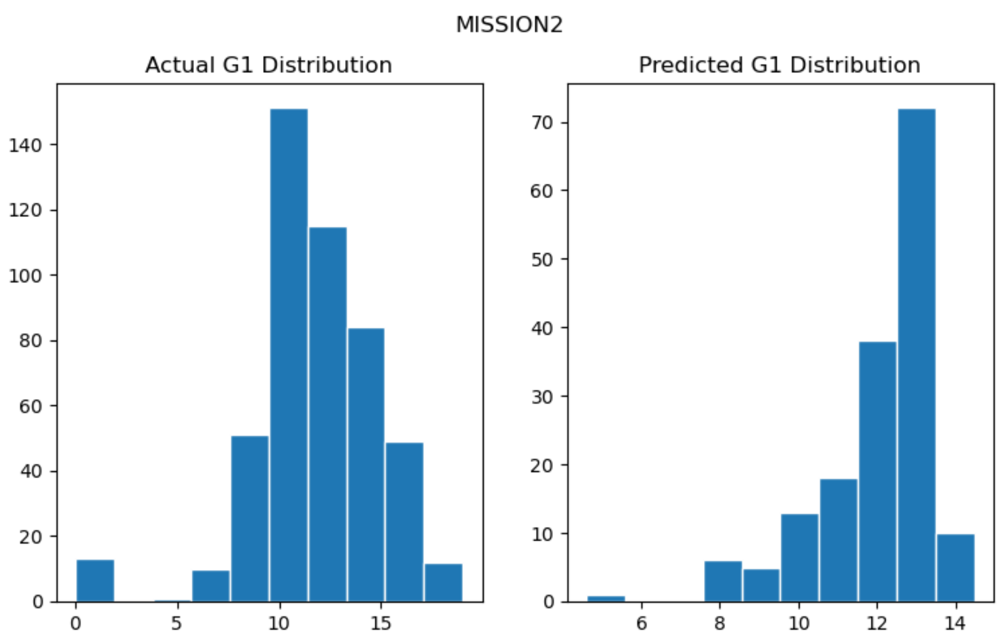
3.4.4 Model Selection - Ridge Regression

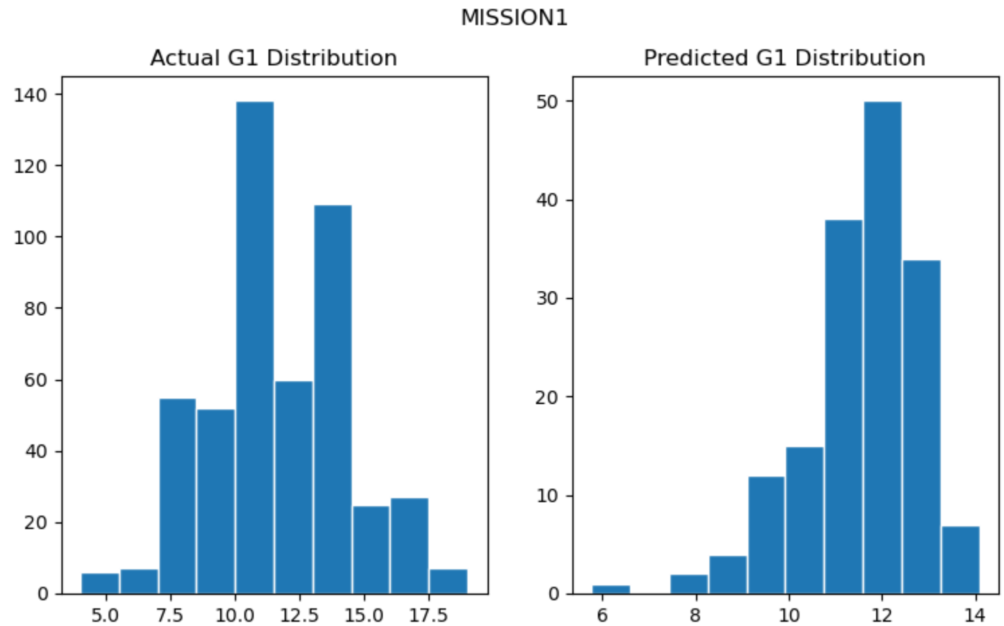
The results obtained for the Ridge Regressor are as follows. The number of neighbors 'k' parameter was tuned for different values before finding the optimized estimator.

The final results for this regressor is as follows:

Model	RMSE	MAE	R2
Mission1	2.490	1.8876	0.2315
Mission2	2.7584	2.0602	0.2449
Mission3	1.003	0.7344	0.9001







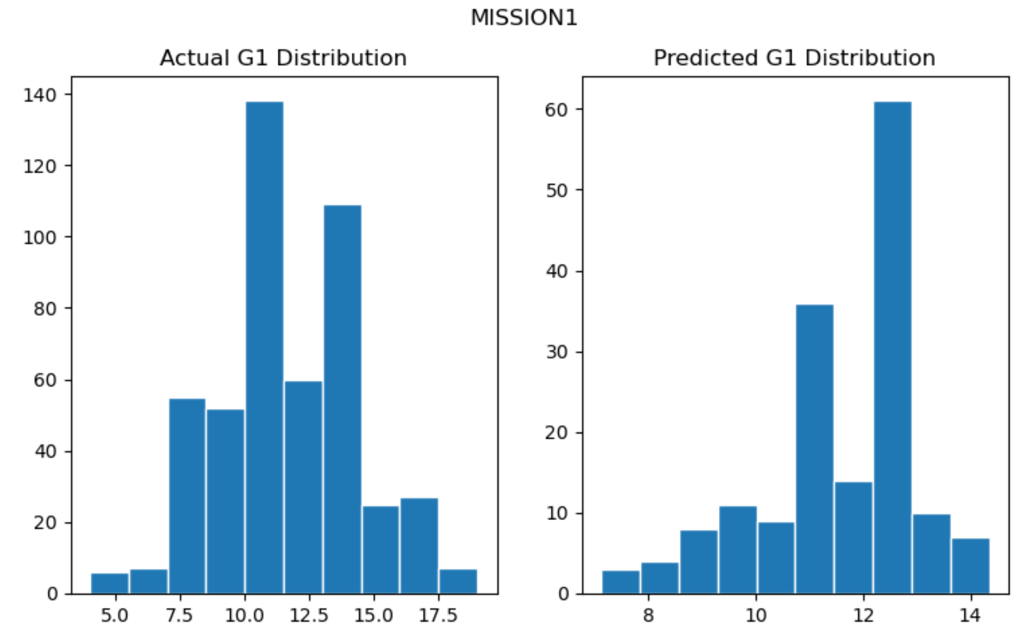
The last set of plots above correspond to Mission 3 (reconsider the typo)

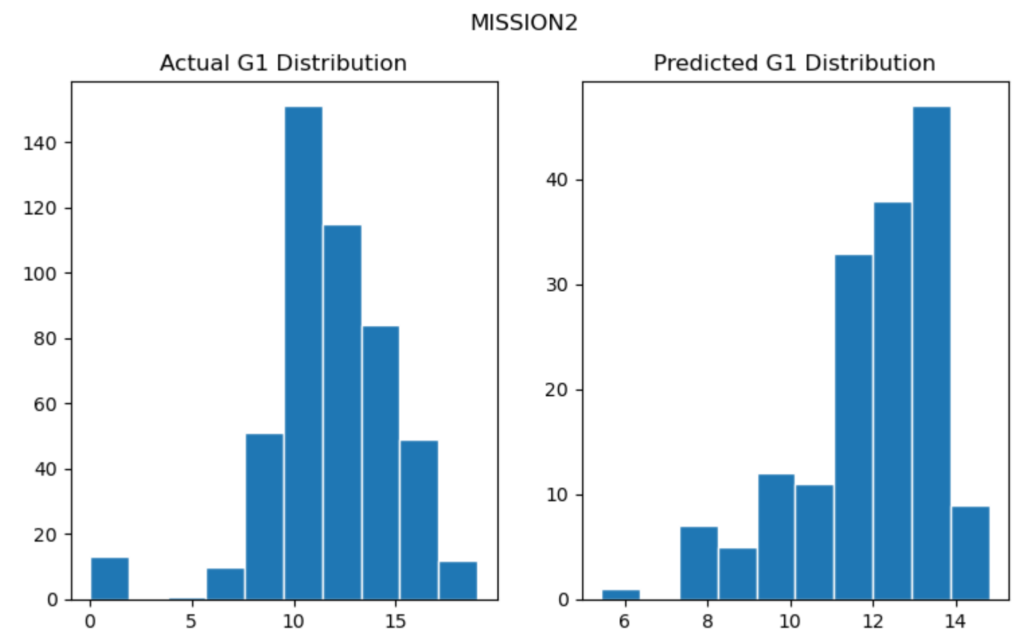
3.4.5 Model Selection - MLP Regression

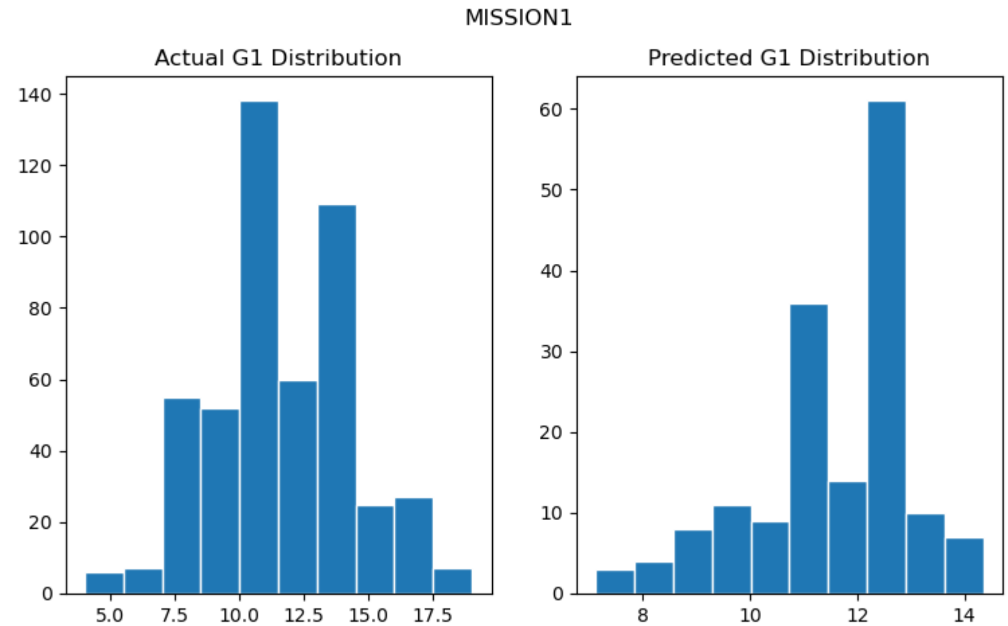
The results obtained for the MLP Regressor are as follows. The number of neighbors 'k' parameter was tuned for different values before finding the optimized estimator.The hidden layer size in the case of mission 1 = 119, for mission 2 = 118 for mission 3 = 116. These values were slightly varying when cross validation was done over and over again.

The final results for this regressor is as follows:

Model	RMSE	MAE	R2
Mission1	2.4758	1.8772	0.2510
Mission2	2.705	2.016	0.2739
Mission3	1.0119	0.76552	0.8984


MISSION1
Actual G1 Distribution
0
20
40
60
80
100
120
140
5.0
7.5
10.0
12.5
15.0
17.5
Predicted G1 Distribution
0
10
20
30
40
50
60
8
10
12
14


MISSION2
Actual G1 Distribution
0
20
40
60
80
100
120
140
0
5
10
15
Predicted G1 Distribution
0
10
20
30
40
6
8
10
12
14



The last set of plots above correspond to Mission 3 (reconsider the typo)

Among all the above models, the paarmeters for SVR, Ridge Regression and KNN Regression were calculated via model selection and cross validation. In the case of MLP Regression, The hidden layer size was more a heuristic parameter and is slightly approximate value. For reference, all the sklearn documentation is added in the References section.

Because the number of constraints (data points) were not significantly comparable to the total number of 30-32 features, adding non-linear features, did not make a lot of sense. So, no Feature Engineering was done. Because the number of features was too high, inorder to avoid overfitting, feature selection was thoroughly done.

4 Analysis: Comparison of Results, Interpretation

Here the score plots for all the 4 models is plotted below:

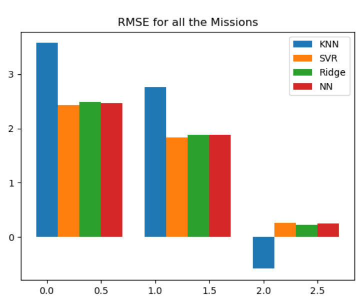


Figure 19: RMS error

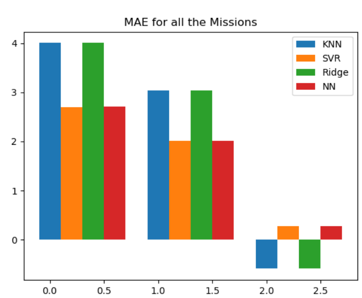


Figure 20: Mean Absolute Error

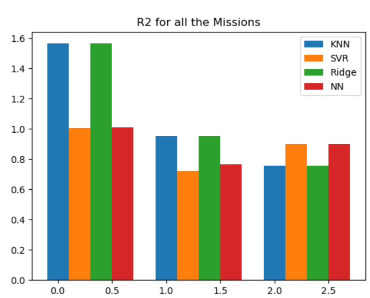


Figure 21: R-squared

In the case of Mission1, SVR and MLP Regressor gave the least errors but a relatively low R2 scores. In the case of Mission1, Ridge Regressor and KNN gave better results. In the case of Mission 2, again KNN and Ridge Regressor gave better results. In the case of Mission3, MLPRegressor and SVR gave better results and also the R2 score is slightly close to and higher than the baseline models.

The selected features corresponded to that obtained from Recursive Feature Elimination technique.

MISSION1: Best model: MLP Regression No Normalization
MISSION2: Best model: MLP Regression No Normalization
MISSION3: Best model: Ridge Regression No Normalization

Overall, the optimization was not upto the mark because of the lower correlation of the features with the target variable. Also, dimensionality reduction conducted here was also more approximate as the exact number of features needed was not known. The mutual information between the features also suggest the extreme correlation of only certain small number of features (like prior grades had more effect on the target, nullyfying the importance of other features). Also, since the numeric features were not highly continuous, resulting in higher uncertainties in the distribution of the datasets, making the dataset more challenging. Also, normalization did not result in satisfactory results and hence, I omitted normalization all together.

5 Libraries used and what you coded yourself

5.1 Python Libraries

I have mentioned the list of libraries that were used in this project -

- **Numpy**
- **Pandas**
- **Matplotlib**
- **sklearn**-preprocessing
- **sklearn**.linear_model-LinearRegression, Ridge
- **sklearn**.metrics-mean_squared_error, mean_absolute_error, r2_score, accuracy_score
- **sklearn**.feature_selection-SelectKBest, f_regression, RFE, RFECV, mutual_info_regression
- **sklearn**.svm-SVR
- **sklearn**.pipeline-make_pipeline
- **sklearn**.preprocessing-StandardScaler
- **sklearn**.neural_network-MLPRegressor
- **sklearn**.neighbors-KNeighborsRegressor
- **sklearn**.model_selection-cross_validate
- **seaborn**

A clear description of the Feature Selection techniques and the parameters were presented in the previous sections.

As per the models, for KNN Regression, I used the sklearn's K-NeighborsRegressor was used with the k value optimized during cross validation. The range of the k-value was [1, 30] only intergers. Here the prediction is done based on the prediction of the 'k' nearest neighbors. The target prediction is a simple regression problem.The points in the neighbors are equally weighted, meaning the weights here is uniform. the algorithm to compute the nearest neighbors was automaticaaly chosen depending on the data to be fit. The available algotithms include - 'Brute force', 'ball_tree', 'kd_tree'

For Support Vector Regression, I used the sklearn's SVR with the C (regularizer) value optimized during cross validation. The range of th C-value was [0.1, 50] in increments of 0.01. The regularizer here is l2 regularization technique. The coefficients for the kernels are floats here. The epsilon term here is chosen to be default that is there is no penalty for the training loss is recorded within this deviation.

For Ridge Regression, I used the sklearn's Ridge() with the alpha (regularizer) value optimized during cross validation. The range of the aplha-value was [0.01, 10] in increments of 0.1.The alpha values for chosen to be small, to reduce the variances in teh estimates.

For MLP Regression, I used the sklearn's MLPRegressor() with a single hidden layer and size (c) value optimized during cross validation. The range of the c-value was [100, 120] in increments of 1 The activation function was ReLU and for the optimizer, the default 'Adam' optimizer was used. The learning rate was maintained constant throught at 0.001 to avoid underfitting. the network was run for a total of 200 iterations with Nesterov momentum with a friction value of 0.9. There was no early stopping required in this case.

Most of the concepts studied and implemented in this project were a part of the course EE559. In the case of feature selection, Recursive Feature Elimination and Recursive Feature Elimination cross validated were additionally studied. In the case of Neural Networks used here, the concepts like momentum and ways to avoid overfitting (early stopping) were all additionally added in this project.

6 Contributions of each team member

There was individual contribution on this project.

7 Summary and conclusions

In this project, a detailed preprocessing and feature selection was conducted on the Portugese Student performance dataset. In the feature selection, the prime focus was on dimensionality reduction to avoid overfitting. Recursive Feature Elimination technique corresponded in optimized features. Important concepts like correlation, mutual information, diferent performance metrics and Machine Learning and Deep Learning models were thoroughly studied and implemented. In the three missions mentioned, the MLPRegressor gave best results for Mission1, Mission2 and the RidgeRegressor gave better results from the Mission3. The results obtained so far need further optimization, feature engineering and supposedly data augmentation to get better results on the testing data using already existing ML models. Also, further investigation on Normalization techniques and different ML models can be an extrapolation of this project.

References

[1] *sklearn-SVR* `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html`

[2] *sklearn-Ridge Regression* `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html`

[3] *sklearn-MLP Regression* `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html`

[4] *sklearn- K-Nearest Neighbor for Regression* `https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html`

[5] *sklearn-cross validation* https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html

[6] *sklearn-mutual information FS* https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html#sklearn.feature_selection.mutual_info_regression

[7] *sklearn-Recursive Feature Elimination* https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

[8] *sklearn-f regression* https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.f_regression.html

[9] *sklearn-Recursive Feature Elimination-CV* https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFECV.html