MIE 1624 Introduction to Data Science and Analytics – Winter 2023

Assignment 2

Anbumanivel Mohan Suganthi - 1008696653

March 12, 2023

The aim of this assignment is to train, validate and tune multi-class ordinal classification models that can predict a survey respondent's current yearly compensation bucket, based on a set of survey responses by a data scientist.

1. Data Cleaning

First, the feature 'Duration (in seconds)' is dropped because the survey duration is unrelated to the target. Also, the features 'Q29' and 'Q29_buckets' are removed because 'Q29_Encoded' represents them as it is already encoded based on categories. To improve the accuracy of target prediction, irrelevant features are eliminated based on the questions. The questions that were dropped from consideration are Q5, Q6, Q7, Q9, Q14, Q26, Q32, Q35, and Q44. Additionally, the "Other" option in multiple-choice questions is removed as it does not provide any valuable information. Furthermore, upon examining the data, it was found that several columns have a considerable number of missing values. This is primarily because of the nature of multiple-choice questions, where missing values are the options that respondents did not choose. And for some questions, the participant did not choose to answer. The percentage of missing values was calculated and analyzed. And it was determined that the columns with 80% or more missing values are eliminated as they are a minority and do not contribute much information to the output prediction. The first row containing the question descriptions is dropped.

To encode the categorical data using order, an ordinal encoding approach was utilized. The following columns were selected for encoding: Q8, Q11, Q16, Q25, Q30, and Q43. Before encoding, missing values in these columns were imputed with the most common value (mode). The selected features were then encoded from low to high. The remaining categorical data: Q2, Q3, Q4, Q23, Q24, and Q27 did not have any missing values and were encoded using label encoding. For the remaining features, which were multiple-option questions, the existing values were replaced with 1 to indicate that the respondent selected that particular option. On the other hand, missing values were filled with 0 to indicate that the respondent did not select that option. This approach effectively converts the multiple-choice questions into binary variables, making them easier to handle for further analysis. After completing the data cleaning process, 52 features are remaining in the dataset.

2. Exploratory data analysis and feature selection

Feature Engineering is a process that involves extracting and organizing the most relevant features from raw data using domain knowledge. Its primary objective is to improve the accuracy and effectiveness of machine learning models by utilizing these extracted features during training. The extracted features are carefully selected to align with the purpose of the machine learning model, and this approach helps to enhance the efficiency of the model, making it easier to detect patterns in the data and providing greater flexibility in feature selection.

For this assignment, feature engineering can help us identify the most important features with high feature importance and correlation with yearly compensation. By selecting these essential features, we can build better models and improve prediction accuracy. First, a correlation graph is displayed to visualize relationships between the features. The correlation plot is shown as a heatmap with a range from -1 to 1. Since there are many features, the correlation plot is difficult to analyze. Therefore, the feature importance is plotted as a bar chart based on the correlation of all features with 'Q29_Encoded' in descending order of importance. The top 5 important features are Q4, Q11, Q16, Q2, and Q30.

Having too many features can increase training time, model complexity, and the risk of overfitting. Therefore, feature engineering is applied to the dataset to reduce the number of features. Lasso regression is performed for the feature selection process with an alpha value of 0.05. The features that have a coefficient of 0 are removed, resulting in a dataset with 27 features, including the target variable. This approach helps to reduce the complexity of the model and improve its performance by focusing on the most important features.

3. Model implementation

When the target variable has more than two classes and the classes are ordered, an ordinal logistic regression model is used. For this assignment, the target variable has 15 classes and is ordinal, so the ordinal logistic regression is implemented. To implement the model, an algorithm is designed to perform multiple binary classifications with orders. For each iteration, the algorithm divides the 15 classes into two classes and labels them as 0 and 1. The first class, class 0, is treated as label 0 and the remaining classes are treated as label 1. Then, the algorithm makes binary predictions for each sample. In the next iteration, the algorithm uses the probability of the group of classes from the last iteration to subtract the group probability from the current iteration. This step is repeated until the last class is reached. For class 14, the probability is obtained directly by extracting it from label 1 in the binary classifier. Finally, an array of probabilities for each class is obtained for all the samples. The class with the highest probability is chosen as the predicted class for each sample.

The entire dataset is split into training (70%) and test (30%) data. To calculate accuracy and perform 10-fold cross-validation, the test dataset is set aside, and the training set is split into new training and validation sets. It is necessary to avoid some features dominating others in magnitude during the training process. Since there are ordinal data in some columns, standardization should be performed separately on the training and validation sets. Accuracy shows consistency across all folds in the training set, while there is some variability in the validation set. The average accuracy and variance for the training set are 41.007% and 0.046, respectively. On the other hand, the average accuracy and variance for the validation set are 39.526% and 3.741, respectively.

Bias is the difference between the average prediction of a model and the actual value, while variance refers to the variability of the model's prediction for a given data point. To develop models for this task, the hyperparameter C, which is the inverse of the regularization strength, is chosen. The model is run with different values of the hyperparameter C, such as 0.0001, 0.0005, 0.001, 0.01, 0.05, 0.1, 0.5, and 1, and the bias and variance are computed for each value. A graph is plotted to illustrate the bias-variance trade-off. The results demonstrate that the bias decreases while the variance increases as the value of C increases. The optimal model is obtained when the value of C is set to 0.0001, which corresponds to the lowest mean-squared error with average accuracy and variance for the training set are 37.471% and

0.06, respectively. On the other hand, the average accuracy and variance for the validation set are 37.472% and 4.873, respectively.

4. Model tuning

The logistic regression function has the following parameters: penalty, dual, tol, C, fit_intercept, solver, intercept_scaling, class_weight, random_state, max_iter, multi_class, verbose, warm_start, n_jobs, and l1_ratio. For this task, two hyperparameters have been chosen:

- C: It is the inverse of regularization strength. C will be tuned using a list of values: [0.0001, 0.0005, 0.001, 0.01, 0.05, 0.1, 0.5, 1].
- Solver: It is the algorithm used to solve the optimization problem in logistic regression. Solver will be tuned using a list of values: ['lbfgs', 'liblinear', 'newton-cg', 'newton-cholesky', 'sag', 'saga']

Grid search is performed by iterating over the two sets of hyperparameters and evaluating the performance of the model on cross-validation data. The dataset is imbalanced as most of the respondent's yearly compensation belongs to class 0. Therefore, accuracy is not a good metric for evaluating model performance. Instead, it is recommended to use precision and recall to get a comprehensive evaluation. The F1 score is a useful metric that takes both precision and recall into account. After performing a grid search, the optimal model is found to have C=1 and a ' liblinear' solver, and it achieves a validation F1 score of 0.276 and an accuracy of 39.544%.

A graph depicting the feature importance of the best model is generated using the coefficients of the features. In section 2, the feature 'Q4' exhibited the highest correlation, while 'Q21_8' had the most significant negative correlation. However, after tuning the model, 'Q21_8' was found to be more important, and 'Q28_3' had the highest negative importance.

5. Testing & Discussion

The optimal model, which uses hyperparameters C=1 and Solver='liblinear', exhibits an average training accuracy of 40.597% and a test set accuracy of 40.188%. However, it is observed that the training set produces slightly higher accuracy than the test set. The fact that the training set and test set have low accuracy and f1 score indicates that the model may be underfitting the data. This is further supported by the distribution plots, which show that the model applied to both training and test data results in similar predictions. Moreover, most of the predictions made by the model are for the salary buckets of 0 (0-9,999), 10 (100,000-124,999), and 12 (150,000-199,999), indicating that the model's predictions are overly simplistic and do not capture the complexity of the dataset. These findings suggest that the model may result in poor performance when applied to new, unseen data.

The insights gained from the dataset and the model suggest that the dataset is highly imbalanced, with a significant majority of respondents falling into the first salary bucket. This imbalance in the dataset can lead to an unbalanced model, which affects the model's accuracy. Therefore, to improve the model's performance, it is essential to balance the data and assign appropriate weights to each class. The feature importance analysis revealed that certain questions were highly correlated with respondents' annual compensation, indicating that these questions could be critical in predicting an individual's salary range. Overall, the insights gained from the dataset and model highlight the need for careful consideration of dataset balance and feature selection to develop an accurate and reliable model.

Appendix

Figure 1: Correlation graph

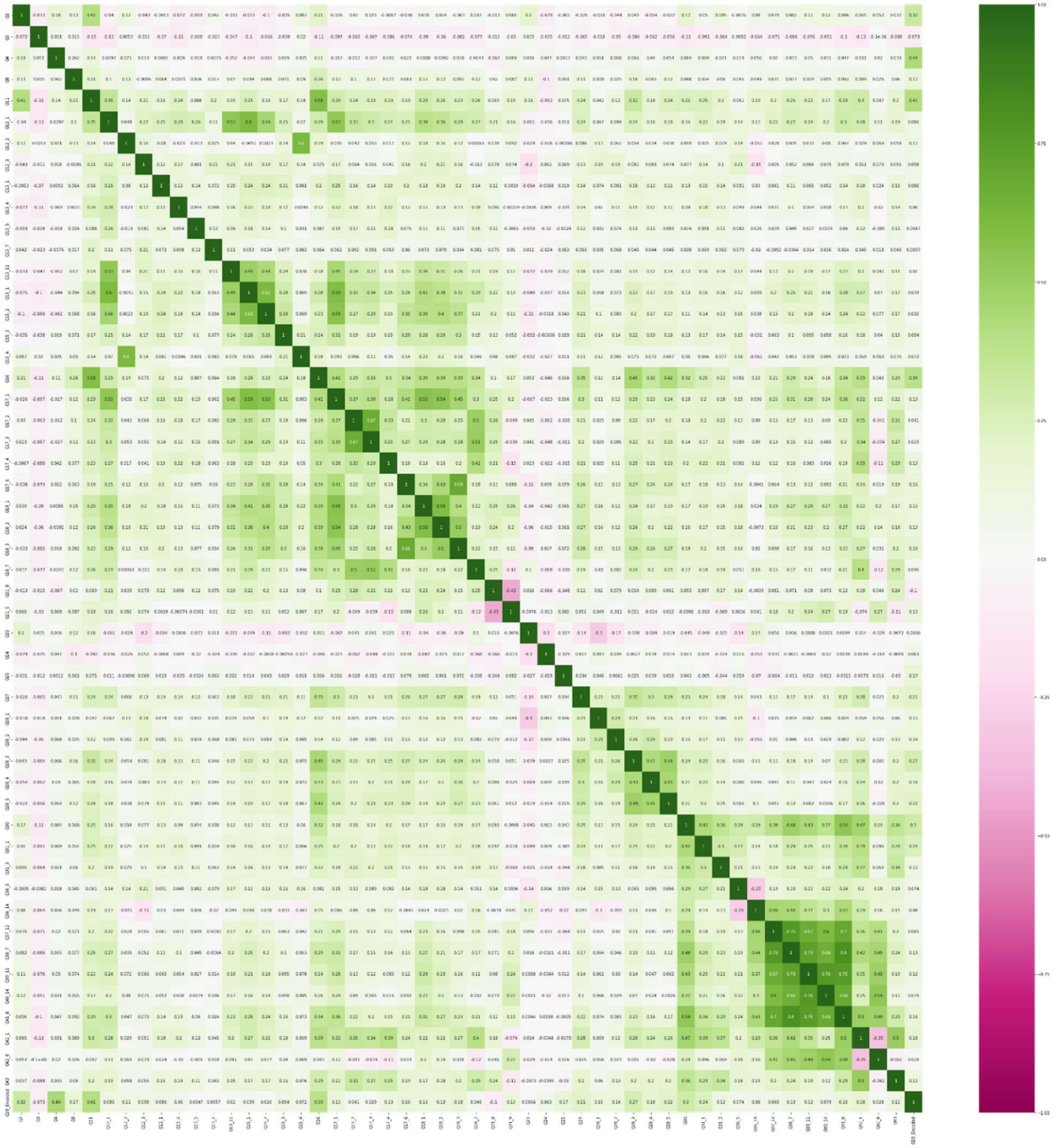

Figure 2: Order of feature importance

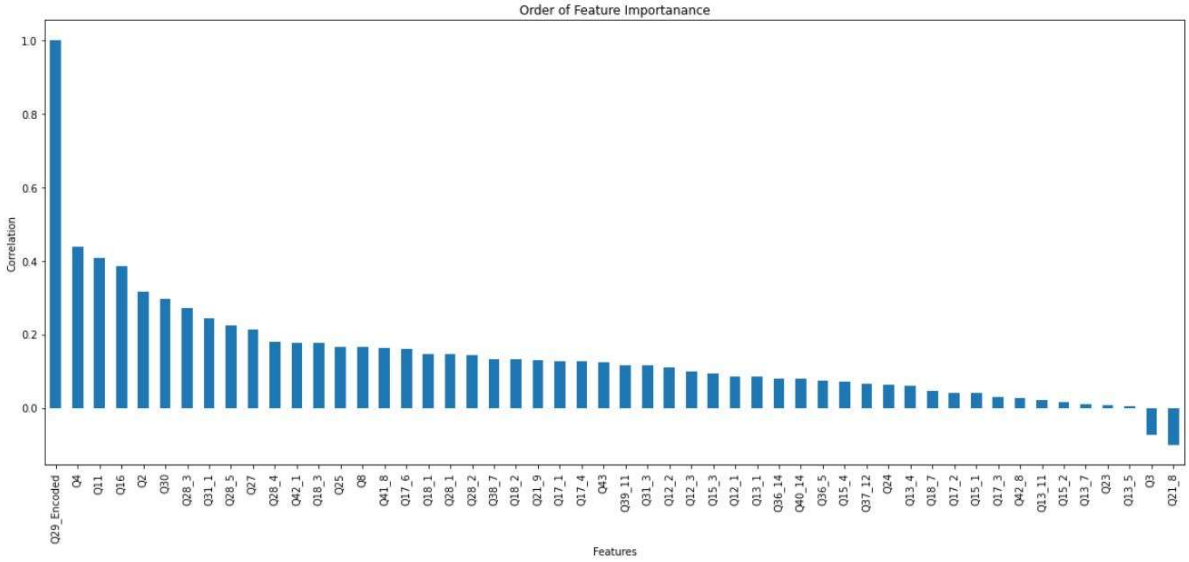

Figure 3: Mean-squared error graph

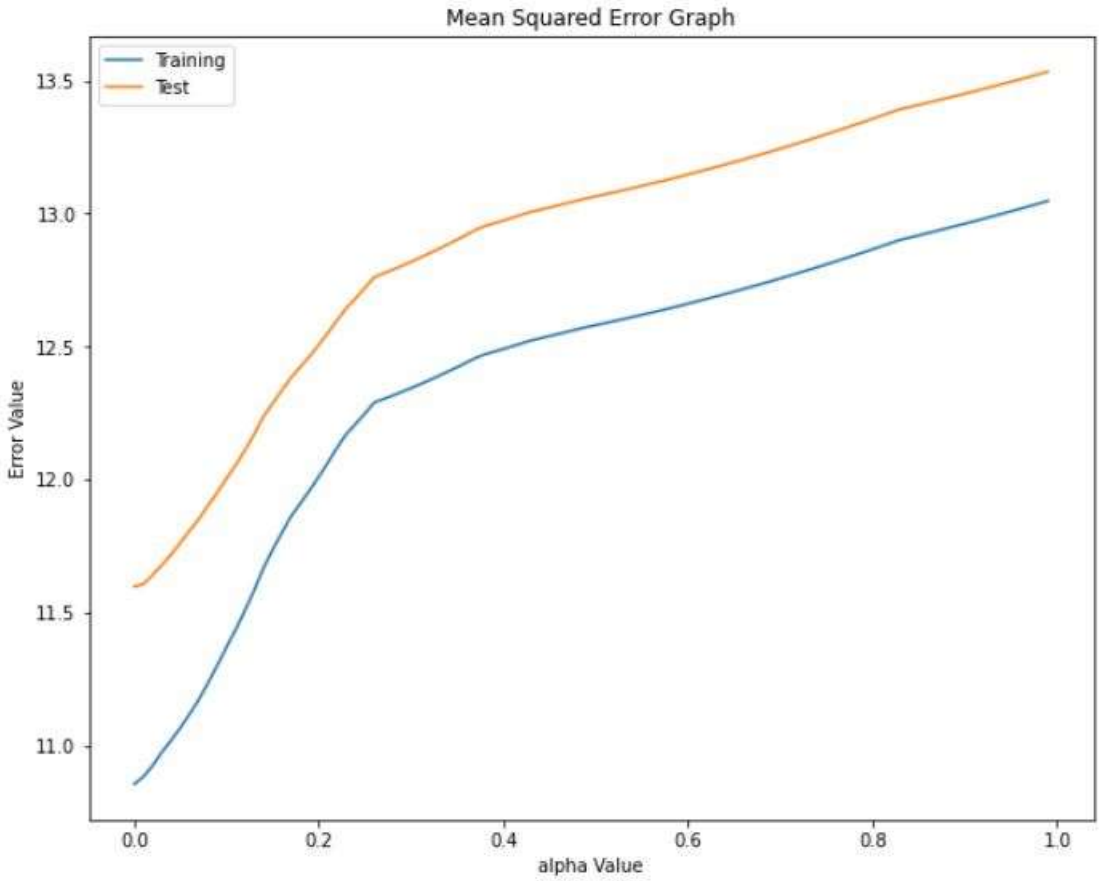

Figure 4: R-squared error graph

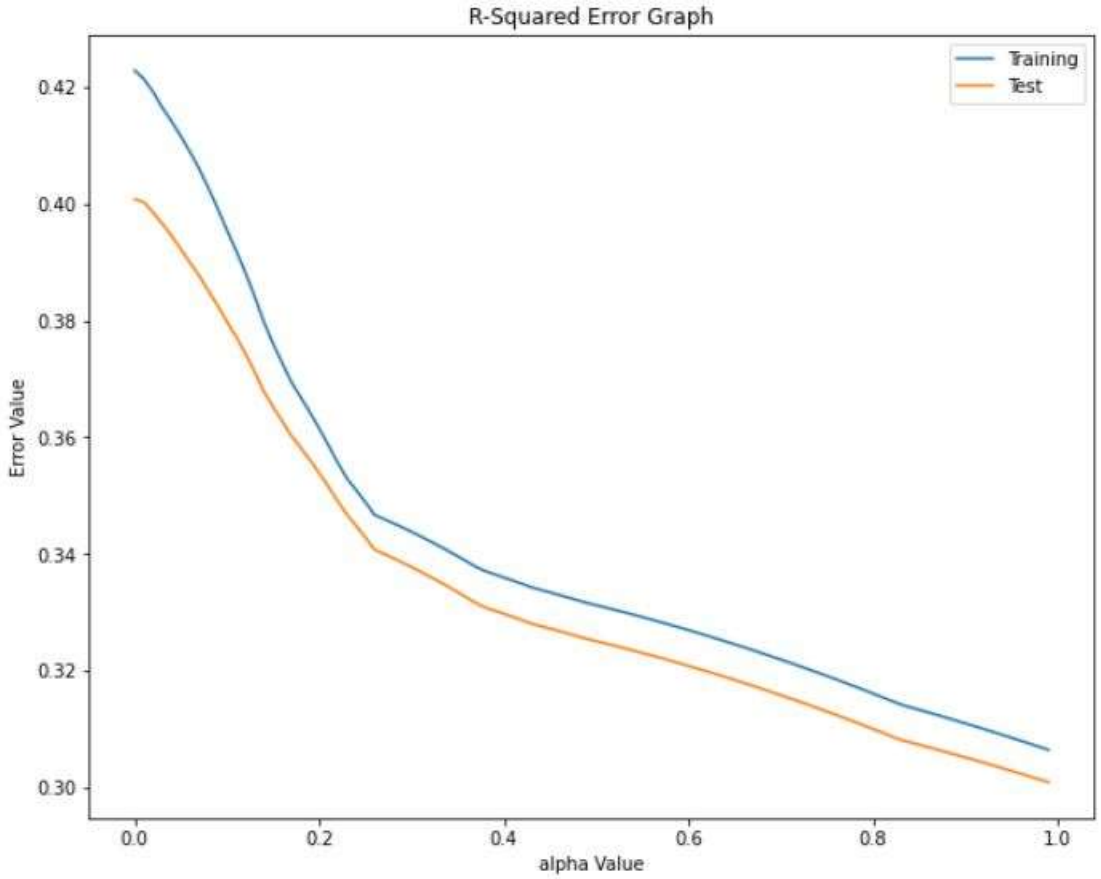

Table 1: Dataset after feature selection

	Q2	Q3	Q4	Q8	Q11	Q12_1	Q13_7	Q13_11	Q15_1	Q15_2	Q16	Q17_2	Q17_3	Q18_3	Q18_7	Q21_8	Q23	Q24	Q25	Q27	Q28_3	Q28_5	Q30	Q31_1	Q37_12	Q40_14	Q29_Encoded
1	8	0	15	2.0	5.0	1.0	0.0	1.0	1.0	1.0	2.0	0.0	0.0	0.0	0.0	0.0	4	11	0.0	4	1.0	1.0	3.0	0.0	1.0	0.0	2.0
2	3	0	16	3.0	5.0	1.0	0.0	1.0	1.0	0.0	5.0	1.0	1.0	0.0	0.0	0.0	11	6	2.0	3	1.0	0.0	0.0	0.0	0.0	0.0	10.0
3	10	0	2	5.0	6.0	1.0	0.0	1.0	0.0	0.0	6.0	0.0	1.0	0.0	0.0	0.0	10	5	3.0	4	1.0	0.0	2.0	1.0	1.0	1.0	10.0
4	5	0	55	5.0	5.0	1.0	0.0	1.0	0.0	1.0	6.0	1.0	1.0	1.0	1.0	0.0	5	3	3.0	3	1.0	0.0	2.0	1.0	0.0	0.0	13.0
5	5	0	55	4.0	5.0	1.0	0.0	0.0	1.0	1.0	6.0	1.0	1.0	1.0	0.0	0.0	4	3	3.0	3	1.0	1.0	5.0	1.0	0.0	0.0	13.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8132	5	0	55	4.0	5.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	2	1	4.0	3	1.0	1.0	0.0	0.0	0.0	0.0	10.0
8133	4	0	20	4.0	2.0	1.0	0.0	1.0	1.0	1.0	2.0	0.0	0.0	1.0	0.0	0.0	4	2	4.0	3	0.0	1.0	2.0	0.0	1.0	0.0	0.0
8134	5	0	20	3.0	2.0	1.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	3	3	0.0	1	0.0	0.0	1.0	1.0	0.0	0.0	0.0
8135	2	0	51	4.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0	6	4	4.0	1	0.0	0.0	2.0	0.0	1.0	1.0	2.0
8136	4	4	24	5.0	3.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	10	0	0.0	4	0.0	0.0	0.0	0.0	0.0	0.0	1.0

8136 rows × 27 columns

Figure 5: Training and validation accuracy across each fold, average and variance of accuracy

```
Fold 1 training accuracy: 40.917%
Fold 2 training accuracy: 41.054%
Fold 3 training accuracy: 41.385%
Fold 4 training accuracy: 41.288%
Fold 5 training accuracy: 40.663%
Fold 6 training accuracy: 40.909%
Fold 7 training accuracy: 41.065%
Fold 8 training accuracy: 41.104%
Fold 9 training accuracy: 40.968%
Fold 10 training accuracy: 40.714%
Average training accuracy: 41.007%
Variance of training accuracy: 0.046

Fold 1 validation accuracy: 40.702%
Fold 2 validation accuracy: 38.421%
Fold 3 validation accuracy: 38.246%
Fold 4 validation accuracy: 37.018%
Fold 5 validation accuracy: 41.404%
Fold 6 validation accuracy: 39.895%
Fold 7 validation accuracy: 38.313%
Fold 8 validation accuracy: 37.083%
Fold 9 validation accuracy: 40.949%
Fold 10 validation accuracy: 43.234%
Average validation accuracy: 39.526%
Variance of validation accuracy: 3.741
```

Figure 6: Bias-Variance trade-off

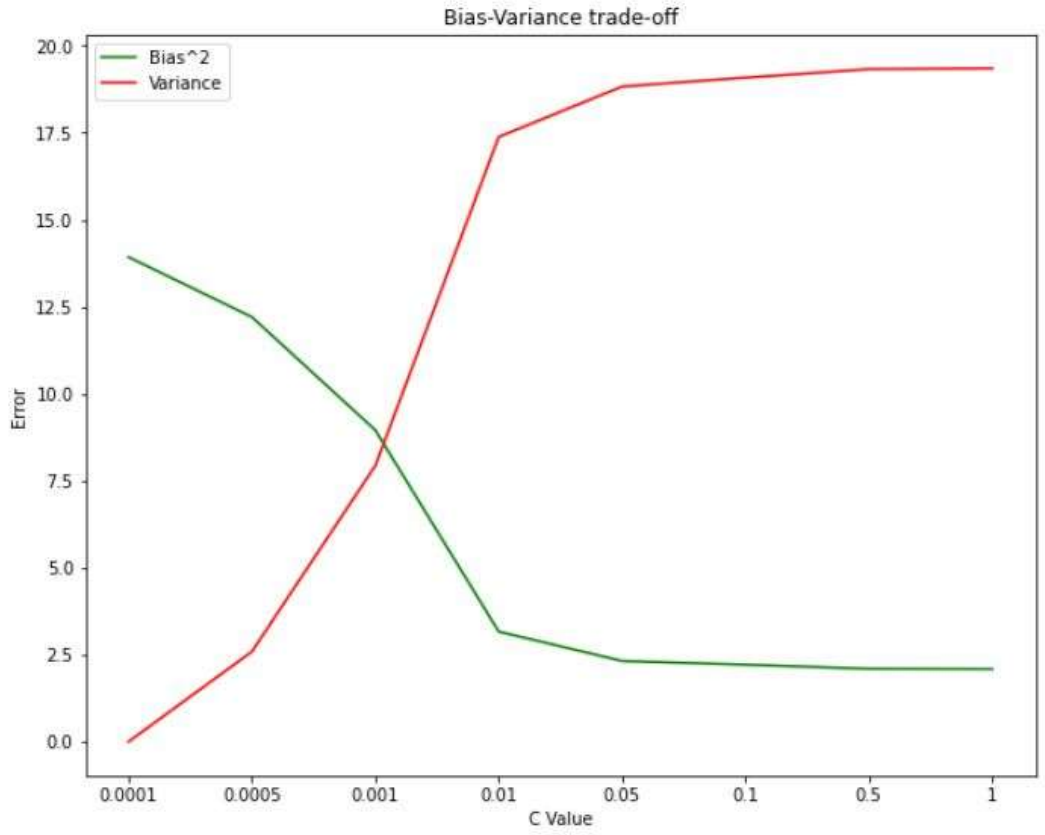

Figure 7: Accuracy for the best performed model in model implementation across hyperparameter C

```
Fold 1 training accuracy: 37.366%
Fold 2 training accuracy: 37.522%
Fold 3 training accuracy: 37.62%
Fold 4 training accuracy: 37.834%
Fold 5 training accuracy: 37.171%
Fold 6 training accuracy: 37.378%
Fold 7 training accuracy: 37.671%
Fold 8 training accuracy: 37.807%
Fold 9 training accuracy: 37.222%
Fold 10 training accuracy: 37.124%
Average training accuracy: 37.471%
Variance of training accuracy: 0.06

Fold 1 validation accuracy: 38.421%
Fold 2 validation accuracy: 37.018%
Fold 3 validation accuracy: 36.14%
Fold 4 validation accuracy: 34.211%
Fold 5 validation accuracy: 40.175%
Fold 6 validation accuracy: 38.313%
Fold 7 validation accuracy: 35.677%
Fold 8 validation accuracy: 34.446%
Fold 9 validation accuracy: 39.719%
Fold 10 validation accuracy: 40.598%
Average validation accuracy: 37.472%
Variance of validation accuracy: 4.873
```

Figure 8: Feature importance graph after model tuning

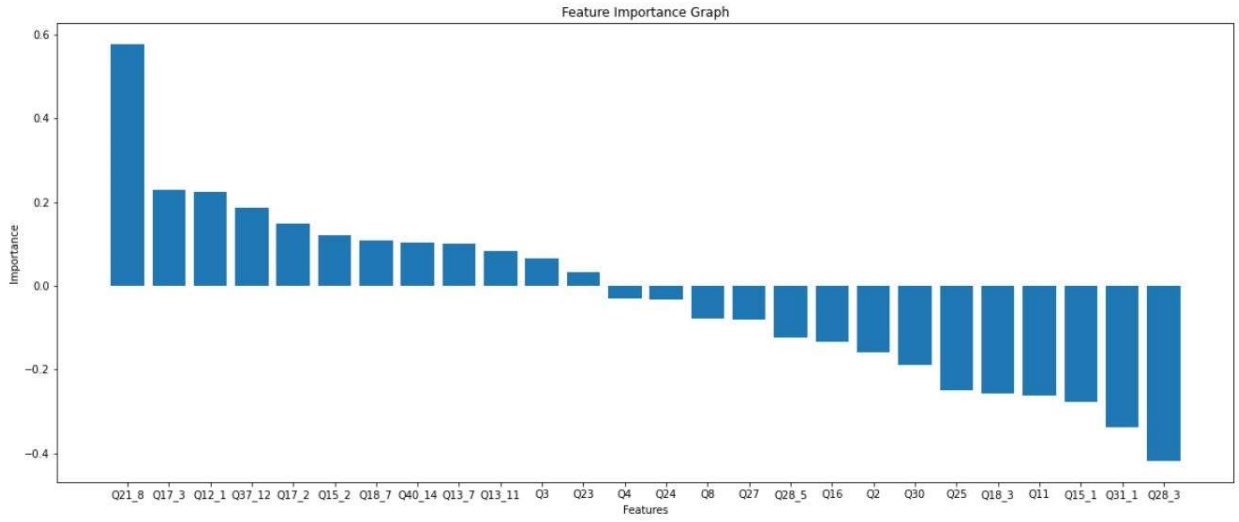

Figure 9: Accuracy and variance of accuracy for the best model

```
Fold 1 training accuracy: 40.917%
Fold 2 training accuracy: 41.093%
Fold 3 training accuracy: 41.346%
Fold 4 training accuracy: 41.307%
Fold 5 training accuracy: 40.624%
Fold 6 training accuracy: 40.968%
Fold 7 training accuracy: 41.007%
Fold 8 training accuracy: 41.046%
Fold 9 training accuracy: 40.929%
Fold 10 training accuracy: 40.714%
Average training accuracy: 40.995%
Variance of training accuracy: 0.046

Fold 1 validation accuracy: 40.877%
Fold 2 validation accuracy: 38.421%
Fold 3 validation accuracy: 38.421%
Fold 4 validation accuracy: 36.842%
Fold 5 validation accuracy: 41.404%
Fold 6 validation accuracy: 39.895%
Fold 7 validation accuracy: 38.313%
Fold 8 validation accuracy: 37.083%
Fold 9 validation accuracy: 40.949%
Fold 10 validation accuracy: 43.234%
Average validation accuracy: 39.544%
Variance of validation accuracy: 3.834
```

Figure 10: F1 for Training and Test Data across 10 Folds for Best Model

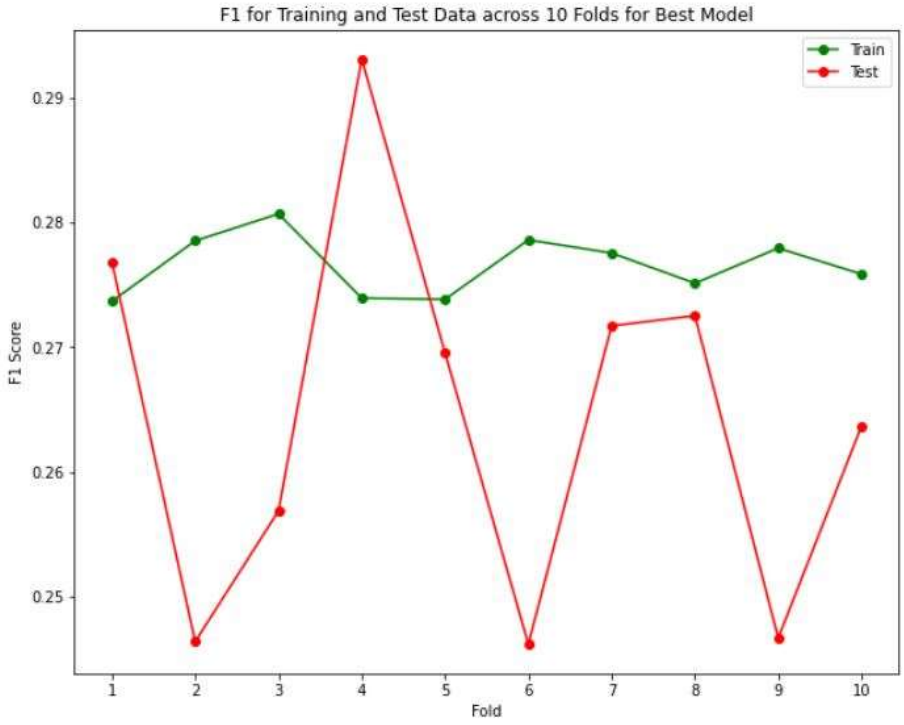

Figure 11: Mean Squared Error for Training and Test Data across 10 Folds for Best Model

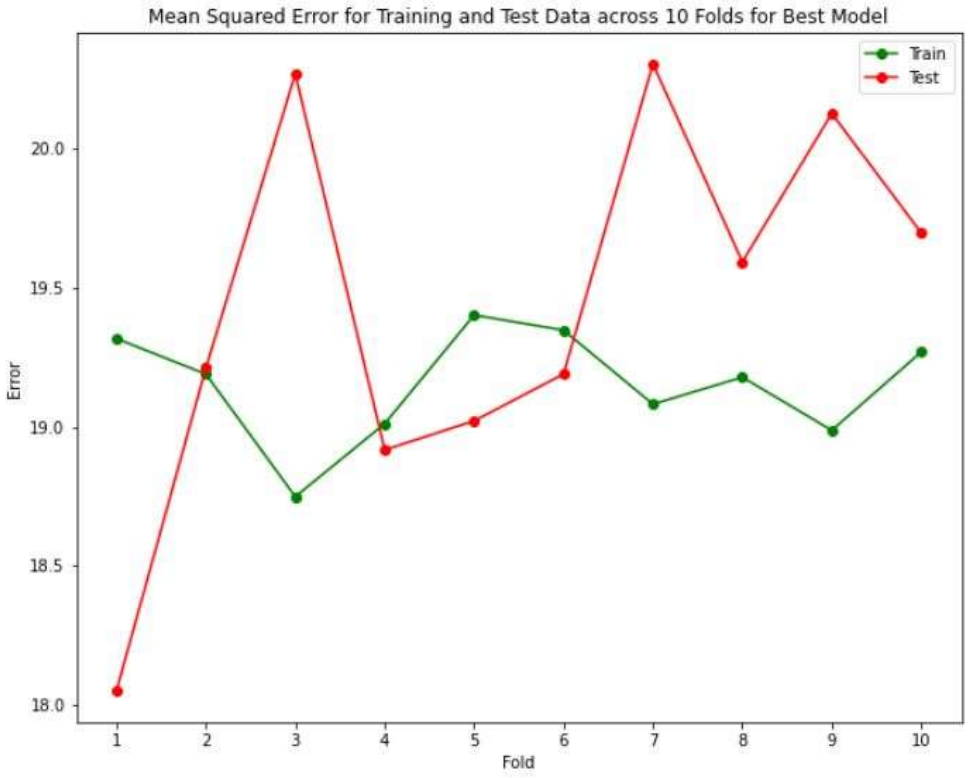

Figure 12: Bias-Variance Trade-off for Training Set across 10 Folds for Best Model

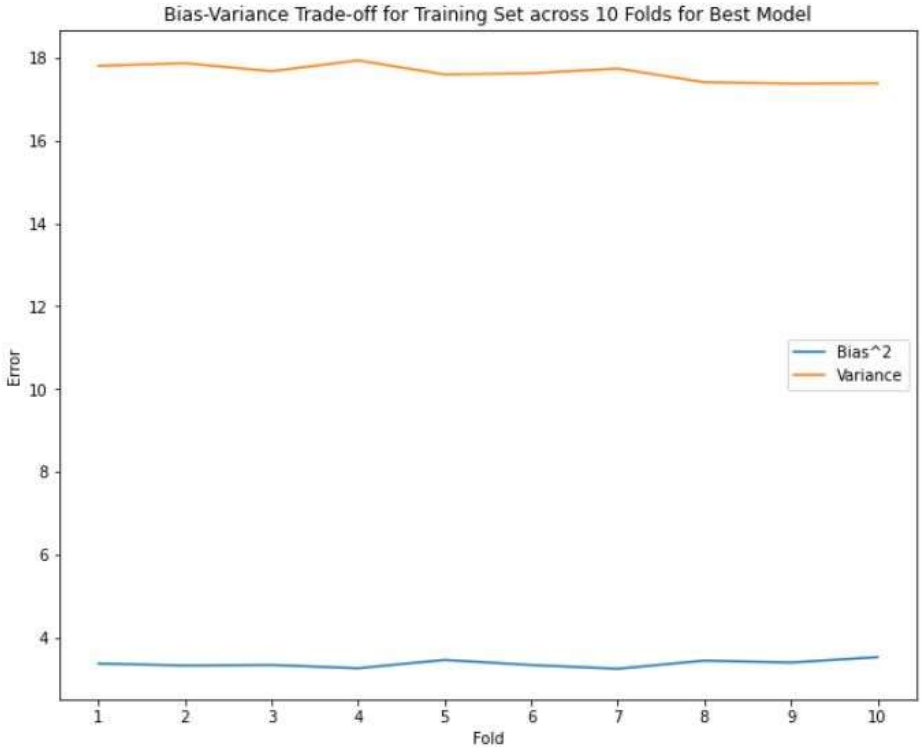

Figure 13: Bias-Variance Trade-off for Test Set across 10 Folds for Best Model

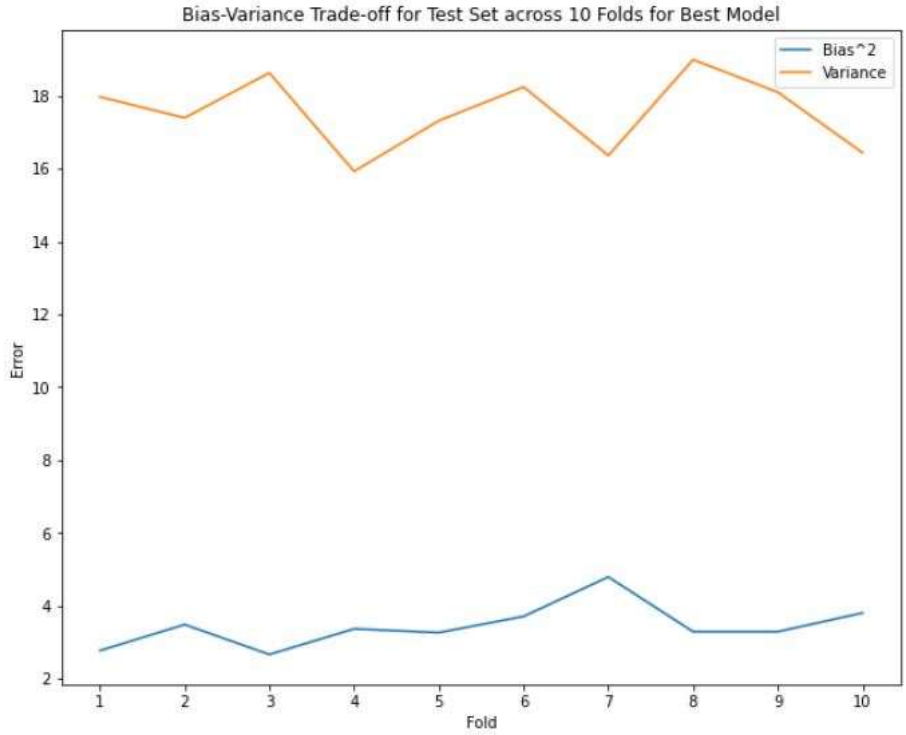

Figure 14: Distribution of True Target Values and their Prediction on Training Set

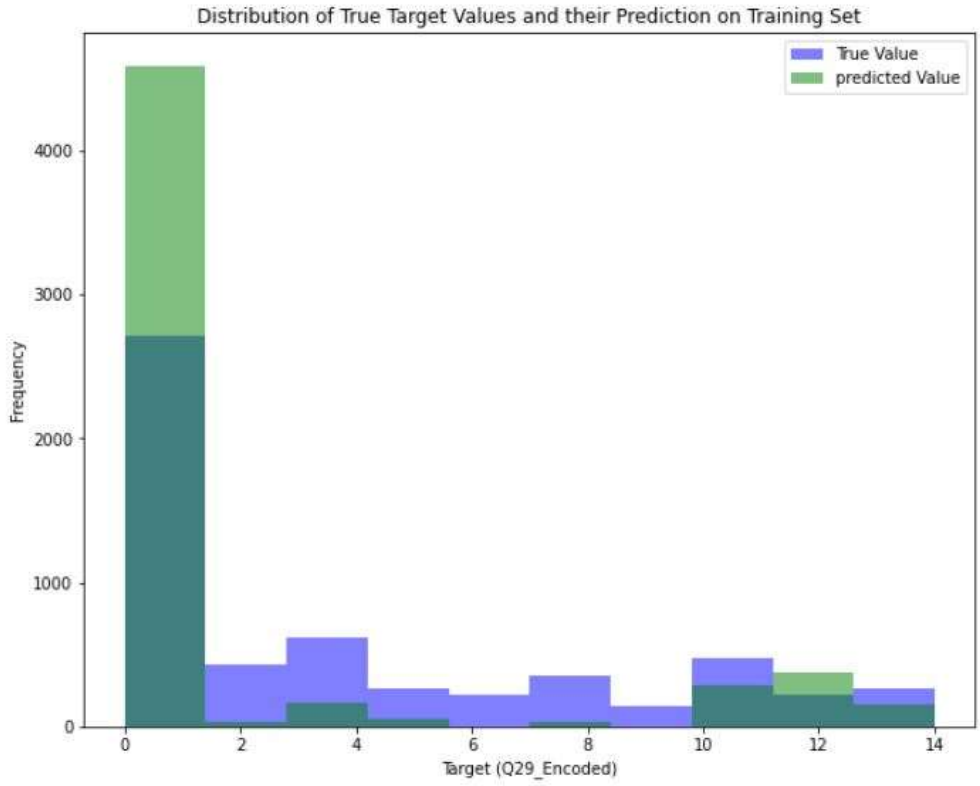

Figure 15: Distribution of True Target Values and their Prediction on Test Set