1. Data Cleaning

1.1 Handle missing values: I handle missing values either by dropping whole columns or by filling the missing values with the mode (for categorical variable).

- Drop columns: I drop the whole column Q33 and Q35 since these two columns have too many null values (69%+ of null values). There are 15391 total observations whereas Q33 has only 4656 non-null values and Q35 has only 2237 non-null values. These two columns with majority of null values cannot give much information for our prediction. The reasons of having majority of null values may be the inability of answering these questions for most data scientists since they most use python, Java and rarely use SQL these databases products as well as Tableau these visualization tools in their work. Dropping these two insignificant features may help to improve latter model performance.
- Imputation: I fill the missing values with the mode for the columns Q11, Q13, Q15, Q26, Q41 since these columns all represent categorical values and only have few null values. These few null values may be due to lack of survey participants' responses of these specific questions at random or data entry error. Imputation of mode for those missing values result in the loss of variation of data.

1.2 Encoding: Machine learning models such as logistic regression require input and output variables to be numeric. Thus, I encode categorical variables into numeric numbers before fitting and evaluating the model. In general, I apply one hot encoding on categorical variables without natural ranking (nominal variables), and apply ordinal encoding on categorical variables with natural raking (ordinal variables).

- One hot encoding: For columns Q2, Q3, Q5, Q11, Q20, Q41 and all multi-columns these nominal variables, I apply one hot encoding by creating binary variable for each category value of each aforementioned categorical variables.
- Ordinal encoding**:** For columns Q1, Q4, Q6, Q13, Q15, Q21, Q22, Q23, Q26 these ordinal variables, I apply ordinal encoding by assigning the ordered integer values into the categories of each aforementioned categorical variables.

1.3: I also drop the columns 'Time from Start to Finish(seconds)', 'Q8', 'Q28' since they are not related to salary analysis. I also drop Q25 since information in Q25 has been recorded in Q25_Buckets and Q25_Encoded.

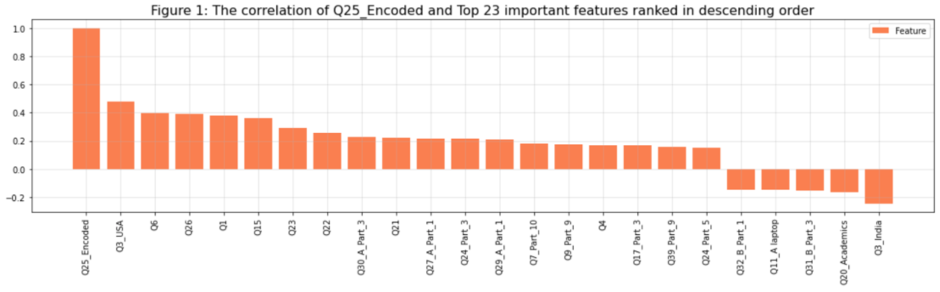
2. Exploratory data analysis and feature selection:

2.1 Usefulness of feature engineering: In this assignment, feature engineering helps to select good features to be included in the model and exclude bad features that play little roles in predictions. By selecting out the important features, it not only helps us to train a more precisive models, but also give us a less complex model, improving running velocity.

2.2 Justification of feature selection process: I apply the correlation this feature engineering technique to select the important features. By setting the correlation of target variable and features >= 0.15 as a boundary of selection, I select the 23 features in total. The reason of setting correlation boundary = 0.15 is that I want to include appropriate amounts of significant features including both of positive and negative correlations. Raising the boundary would result in insufficient features and the exclusion of negative correlations (majority of selected features have correlation from 0.15 to 0.2) whereas lowering the boundary would result in too low correlations to be worth studying. Among those 23 selected features, as seen in correlation matrix plot (refer to Figure 2 in .ipynb file), all have relatively high correlation with target variable and low correlation with other selected features to avoid multicollinearity. Thus, all 23 features look good and I decide to include all Top 23 significant features in the model later.

2.3 Feature Importance:

Figure 1: The correlation of Q25_Encoded and Top 23 important features ranked in descending order



As seen in Figure 1, Q3_USA (Residence country is US) this attribute is most related to yearly compensation.

Furthermore, Q6 (Years of programming), Q1 (Age group), Q26 (Money spent on ML), Q15 (Years of applying ML) are highly correlated with target variable.

3. Model Implementation:

3.1 Data Scaling: From my own research, the feature scaling for ordinal logistic regression is preferred but not required. In this ordinal logistic regression, I decide to scale the data since scaling data does help to make the convergence of the model faster. Scaling data helps us to find the final optimal model faster.

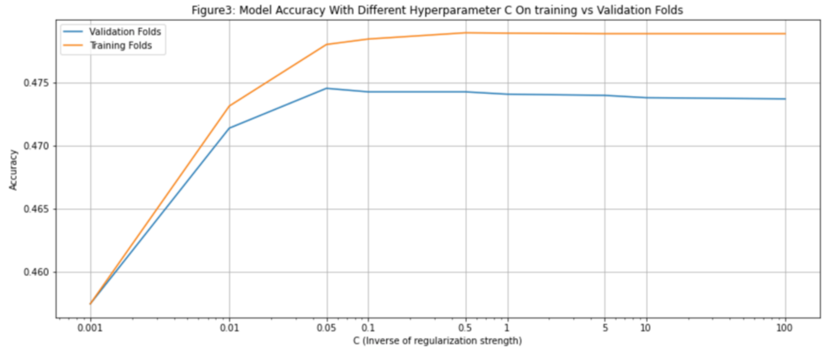
3.2 Model accuracy across folds: After implementing ordinal logistic regression on the training data using 10-fold cross-validation, the model accuracy across the folds has a slight difference but this difference is acceptable. Overall, among those models built, the average of accuracy is approximately 0.47 and the variance of accuracy for folds is 0.0001.

3.3 Loop values in one hyperparameter: Given the solver = 'liblinear', penalty = 'l1', I perform multiple models by changing the value of hyperparameter C (Inverse of regularization strength) in C_list = [0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 100]. Based on validation performance, the optimal hyperparameter C is 0.05. The optimal model is the model with C = 0.05, solver = 'liblinear' and penalty = 'l1'.

Bias-variance trade-off explanation: To explain above model is the optimal model from the bias-variance trade-off perspective, besides the validation folds performance gained above, I will also repeat the above steps and obtain the model accuracy based on training folds performance.

Below displays the model accuracies with different Cs for training folds and validation folds:

Figure3: Model Accuracy With Different Hyperparameter C On training vs Validation Folds



Trending: for training folds, with the increase of C, the model accuracy shows the increasing trend initially with slower increase rate until it is steady;
for validation folds, with the increase of C, the model accuracy increases from C = 0.001 to C = 0.05, then model accuracy starts to decrease from C = 0.05.

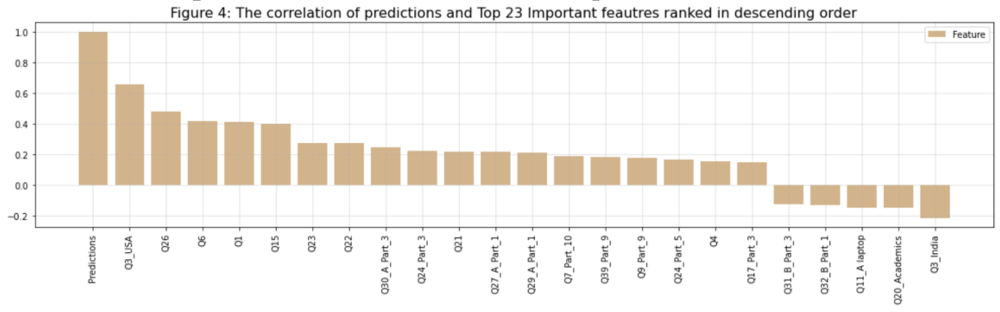
Comparing training performance and validation performance, both of model accuracies for training folds and validation folds are relatively low, which are below 50%. This indicates that the model is underfitting. That means the model experiences high bias and low variance. Considering the underfitting of the model and the increasing training folds performance, we should find the optimal model by finding the highest model accuracy (lowest error) for validation folds. At that moment, the model experiences the best bias-variance trade-off and generalizes best compared with rest ones.
In conclusion, from the bias-variance trade-off perspective, by comparing the training folds performance and validation folds performance, the model with C = 0.05 has the best bias-variance trade-off and is the optimal model.

4. Model Tuning

4.1: Hyperparameters in the model are solver, penalty and C (inverse of regularization of strength). From my own research, the type of penalty and the size of C affect the model accuracy more than the type of solver. Thus, I choose the penalty ('l1', 'l2') and C (0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 100) to do model tuning with the solver = 'liblinear'. Based on the results given in ipynb.file, the optimal hyperparameter is C = 0.05, penalty = l1. The optimal model has the average of model accuracy = 0.475 with the standard deviation = 0.01.

4.2 Performance metrics choice: I choose the accuracy score as the performance metrics since the accuracy is most suitable for the context of this study. In this study, we focus on studying whether the model could correctly predict the salary bucket to which each data point belongs (Whether we have prediction of salary bucket = actual salary bucket). What we care most is total number of (prediction of salary bucket = actual salary bucket) among all predictions. From the statistic perspective, what we care most is the true rates in this multi class classfication. By definition, the accuracy refers to the measure of all correctly identified cases. Its formula is accuracy = correct predictions / total predictions = (TP+TN) / (TP+TN+FN+FP). That is what we want.

4.3 Feature Importance: Below is the ranked importance of selected 23 features in predictions.



As seen in Figure4, the feature Q3_USA is the most determining features in predictions. Q26, Q6, Q1, Q15 also play the important role in predictions in sequence.

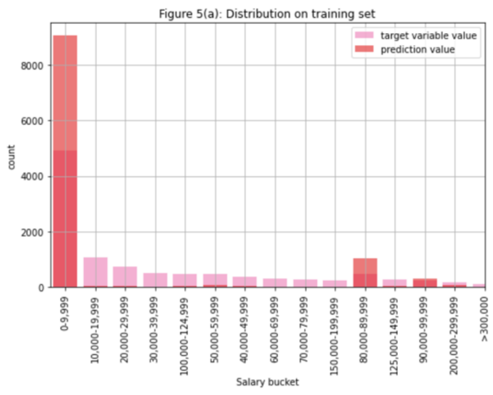
Comparing the Figure 1 (against actual values) and Figure 4 (against predictions), both have the same most determining feature, which is Q3_USA. However, the correlation of Q3_USA and the predictions is higher. Moreover, Q26, Q1, Q15, Q6 are highly correlated with both of actual values and predictions whereas they have different importance rank. There are also differences in the rank of other relatively less important features among these two correlation plots.

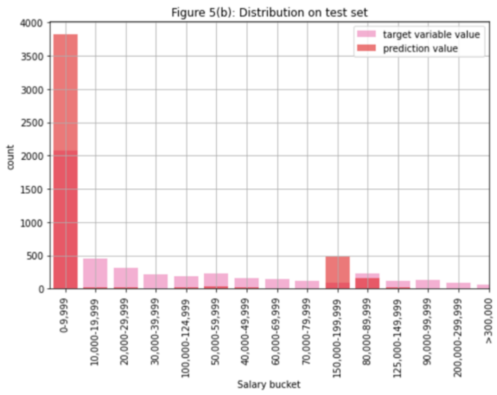
5. Testing & Discussion:

5.1 Optimal model performance: for training set, model accuracy = 0.4777; for testing set, model accuracy = 0.4647.

5.2 Fitting of the model: Given the low accuracies for both of training set and testing set, it indicates that the model is underfitting. It experiences high bias and low variance. To improve the model accuracy, we can take possible measures such as increasing training size or number of parameters. By looking into predictions vs actual values on training vs testing set, I could put forward more pertinent measures by knowing more accurate reasons leading to underfitting.

5.3 Distribution on training vs test set:





As seen in Figure 5(a), for training set, the predictions falling into the 0 - $ 9,999 and $80,000 – $ 89,999 these salary buckets are much more than the actual values falling into this range. However, the predictions are much less than the actual values in other salary buckets.

As seen in Figure 5(b), for testing set, the predictions falling into the 0 - $ 9,999 and $150,000 - $199,999 these salary buckets are also much more than the actual values falling into this range. However, the predictions are much less than the actual values in other salary buckets.

5.4 Insights gained: Based on the model accuracies and Figure 5, the model does not have high performance and has the inconsistencies between predictions vs actual values on both training and testing set. The main reason of it is the imbalances of original dataset. As seen in the distribution of original sample of salary bucket (refer to Figure 6 in .ipynb file), in original sample, there are extremely many data points falling into the $0 - $9,999 this salary bucket (major class). However, there are only few data points falling in the other salary buckets, especially high salary buckets (minor class). The imbalances of data leads to the low predictive performance since the classification algorithms are designed around the assumption of an equal number of data for each class. Another reason of low accuracy is that: even though accuracy is very suitable for the context of this study, it does not deal with the imbalanced data well. There seems like a contradiction between the context of study and properties of original dataset. It is hard to make us choose the appropriate performance metrics which can meet all suitability.

After looking into the reasons leading to underfitting, to address the problems, I may put forward below measures: 1. Try F1-score instead of accuracy (even though accuracy is more suitable for context whereas F1-score deals with imbalanced data better) 2. Apply BalancedBaggingClassifier. 3. Oversampling. When we use imbalanced dataset, we can oversample the minority class using replacement until we can get a balanced dataset for both majority and minority classes.