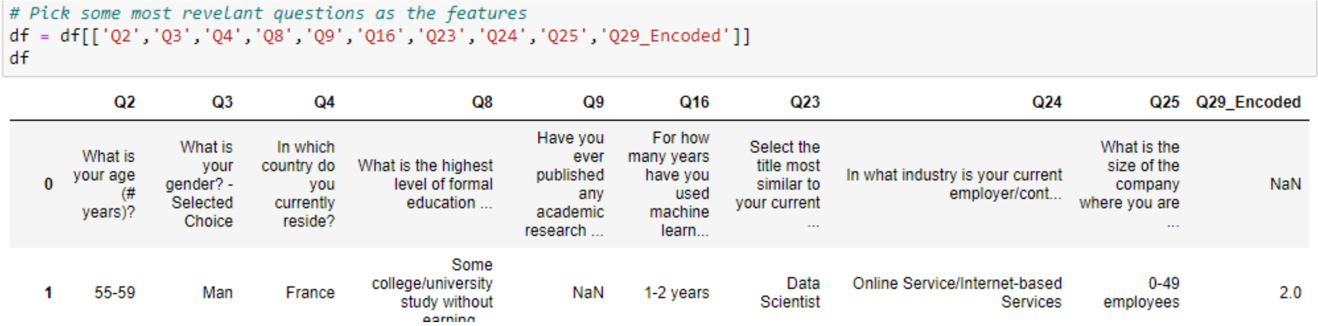
The purpose of this assignment is to train, validate, and tune multi-class ordinary classification models that can classify, given a set of survey responses by a data scientist, what a survey respondent's current yearly compensation bucket is.

1. Data Cleaning

Load the data clean_kaggle_data_2022.csv with pandas, by checking the survey questions on the first row, we can tell there are questions (ex. Time from Start to Finish (seconds), Q44 Who/what are your favorite media source that report on data science topics?) that have minor or no influence on a data scientists' income level. So firstly, we could hand pick some features/survey questions that is more likely to be relevant to the income levels of data scientists. The features picked are shown below:



```
# Pick some most revelant questions as the features
df = df[['Q2','Q3','Q4','Q8','Q9','Q16','Q23','Q24','Q25','Q29_Encoded']]
df
```

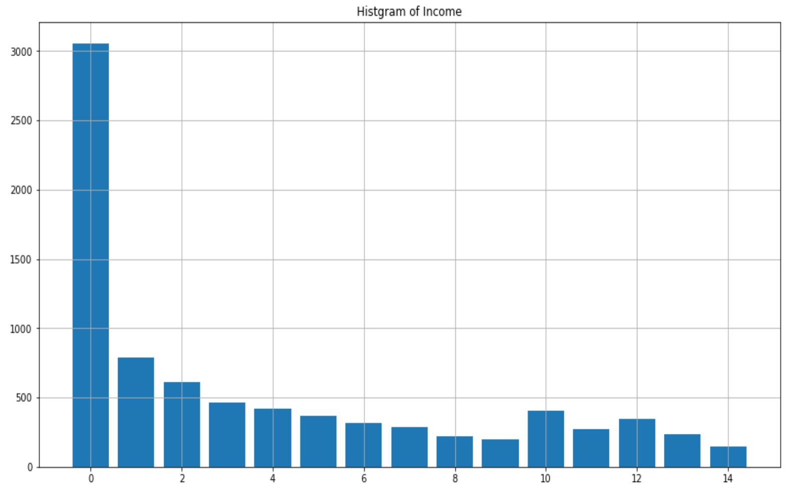
	Q2	Q3	Q4	Q8	Q9	Q16	Q23	Q24	Q25	Q29_Encoded
0	What is your age (# years)?	What is your gender? - Selected Choice	In which country do you currently reside?	What is the highest level of formal education ...	Have you ever published any academic research ...	For how many years have you used machine learn...	Select the title most similar to your current ...	In what industry is your current employer/cont...	What is the size of the company where you are ...	NaN
1	55-59	Man	France	Some college/university study without earning	NaN	1-2 years	Data Scientist	Online Service/Internet-based Services	0-49 employees	2.0

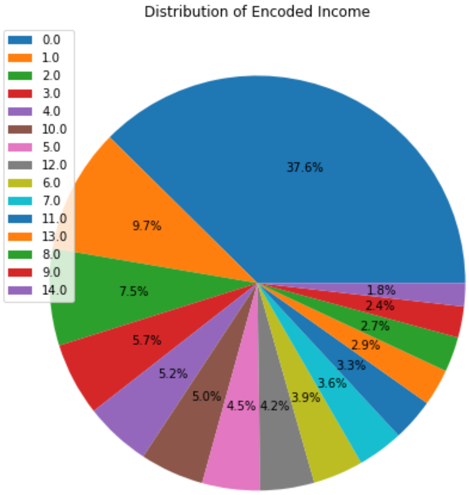
Then we drop the first row with survey questions. By using df.isnull().sum() we found that there exists missing values in Q9 and Q16. Q9 survey questions is asking if the survey respondent has published any academic research (papers, preprints, conference proceedings, etc)? Though the mode (that most responses) is yes, it makes more sense that people who leave this blank have not published any research papers, so we fill the missing values with No. For similar reasons, we fill the Q16 with 'Under 1 year'.

2. Data Exploration

The distribution of income is plotted by a pie chart and a bar chart, from the charts we can see that 37.6% (over 3000) of survey attenders has income level at 0.0, the distribution of income is highly unbalanced, with most survey attenders' income are on the lower side.

Since many all the data except the already encoded Q29 are categorical data, OneHotEncoder is used to create dummies to covert the categorical data to numerical data, this also expand the 9 features to 117 columns. Therefore, we need to cut some unimportant ones with a feature selection algorithm.





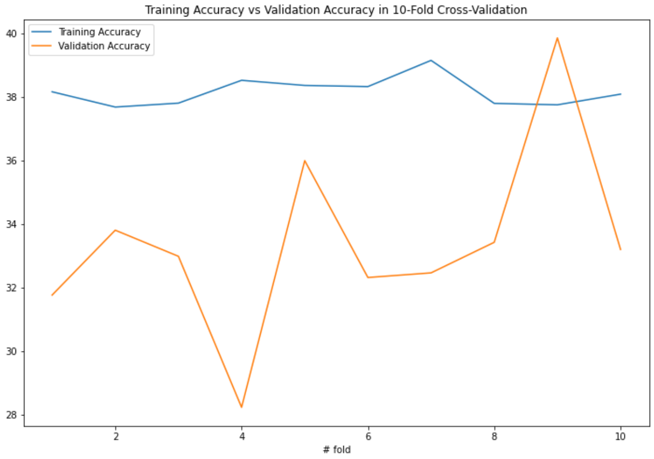
3. Feature Selection

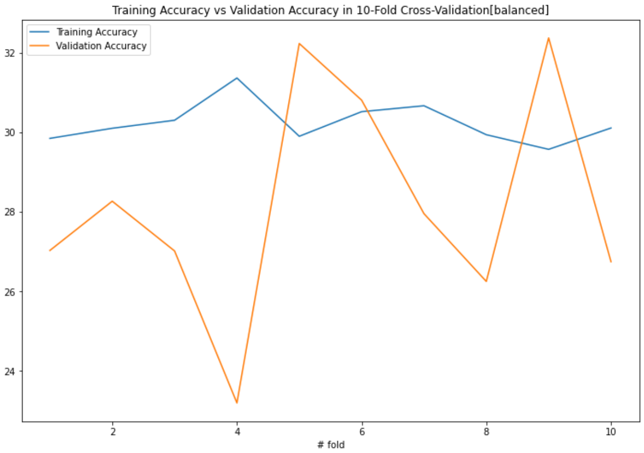
The method chosen to implement feature selection is through random forest classifier. We take X as the train data and y as the target, then fit X to y and use the RandomForestClassifier.feature_importances_ to get the importance of features. Since there are in total 116 columns of features, the bar plot is hard to read. The importance of features can be listed descending to show the most important features after encoding. I firstly try to cut the features that has importance below 0.01, this gave me 21 features left, but in the model implementation phase, the result is not ideal (only around 26% accuracy on train data). Therefore, I decided to only cut the features with importance below 0.001. This gives me 104 columns left, and the heatmap which shows the correlation of features is still hard to read with 104 features. But the list shows the most relevant feature is Q4_United States of America. Then we use the train test split to split the dataset into 70% train, 30% test data. The test data will be untouched until the last Testing phase.

4. Model Implementation

To predict the yearly compensation of survey respondents by ordinal logistic regression, with 10-fold cross-validation on training set, a function with input of x_train, y_train, x_test, C and solver is defined. The function will output the prediction of the salary buckets based on probability. As a result, using the parameter C = 1, the validation accuracy fluctuates between 28.232% and 39.867% , the training accuracy is relatively stable around 38% across the folds. Also, the average training accuracy is 38.172% with variance of 0.184% while the average validation accuracy is 33.409% with variance of 8.038%.

Scaling or normalization is not needed for this dataset, since all the categorical data was transferred into dummies with 0 or 1. Applying standardization to the dataset will make the number lose its meaning.

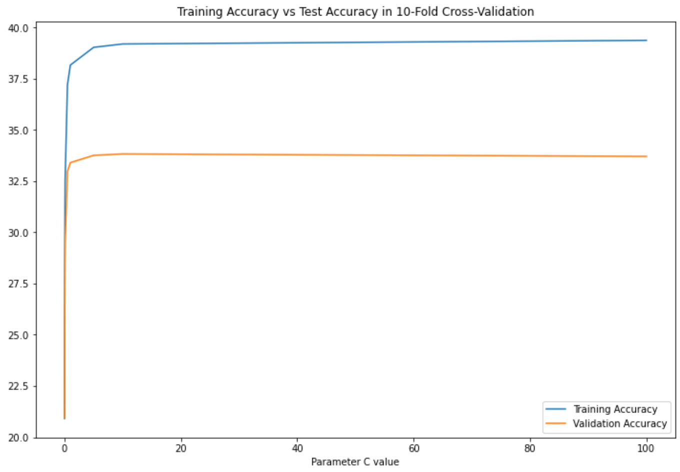




5. Model Tuning

The class_weight parameter is set to be None by default, we can try to set it to be ‘balanced’ to account for imbalanced data. With all other parameters stays the same, the result of balanced class weight is worse than the original one. The average train score is 30.238% and the average validation score is 28.19%. Therefore, this model tuning should be void.

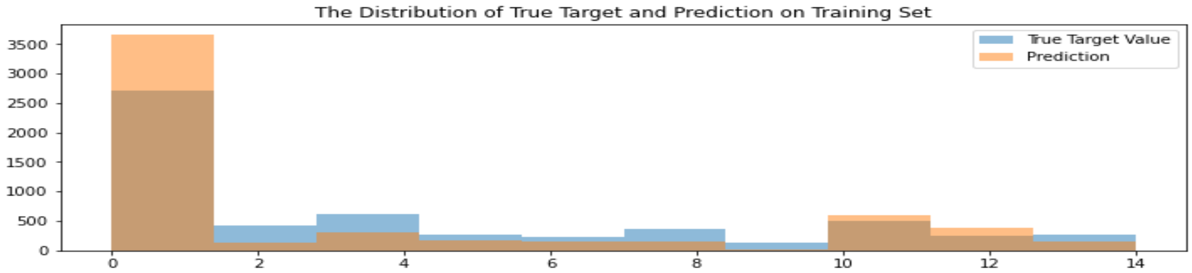
Another important parameter C may have impact on the accuracy of prediction. Pick parameter C as the hyperparameter and treat different C as a new model. With a grid search method, the best model is when C is equal to 10. More specifically, the highest accuracy is 33.833%% with variance of 7.596%. According to bias-variance trade-off, the figure illustrates that the model might underfit at point 0.01 since both training and validation accuracy are low which means the error are high. However, at the point around 10, both training and validation accuracy are relatively high which means the error are relatively low. It also indicates the model is neither overfitting nor underfitting, therefore, choose the model with C is equal to 10 as the best model.

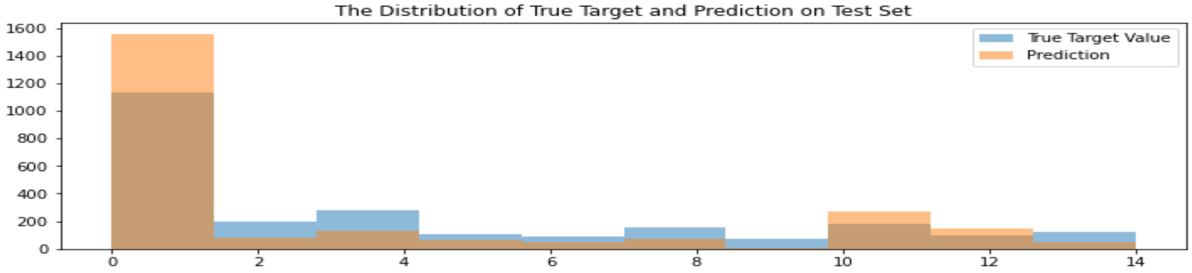


Another grid search is performed to select both the best c and the best solver. Theoretically, 'newton-cg','lbfgs','liblinear','sag' will all give the same results. The grid search is done by searching the highest F1 score of validation set with 10-fold cross-validation under different C and solver method. The performance measure is using F1 score with parameter ‘average’ is weighted which return F1 score and take imbalanced data into account. Since the dataset is imbalanced, F1 score can adjust this issue. As a result, the best model is with hyperparameter c = 10, and the best solver is ‘lbfgs’. The best solvers are different each time running the model, it means that all the solvers give the same result. The best model has a cross validation score of 33.835% with a variance of 7.597%. The heat map is still hard to read with 104 columns, but we can tell by color that ‘Q4_United States of America’ is the most relevant feature which is the same from the previous feature selection step using the random forest and heat map.

6. Testing and Discussion

Using the best model with hyperparameter c = 10 and solver = ‘lbfgs’, we can use it to test the test set which was set aside at the train test split step. The best model train score is 38.644%, the best model test score is 31.643%, which indicates that the model is overfitting. However, the train score of the model is less than 40% which is not a valid prediction. With fewer features selected, the model would perform even worse on the train set, and not necessarily any better on the test set. The reason for the poor estimation accuracy could be the badly distributed of the target data. One method to increase accuracy could be using other classification algorithms and compare their scores to choose the optimal model. The low accuracy hints that ordinal logistic regression might not fit well in this dataset. The distribution of the prediction and the original target values shows that distribution is highly imbalance with majority of them have lower income. Overall, the ordinal logistic regression’s performance on predicting the bucket salary is bad.


The Distribution of True Target and Prediction on Training Set
True Target Value
Prediction


The Distribution of True Target and Prediction on Test Set
True Target Value
Prediction