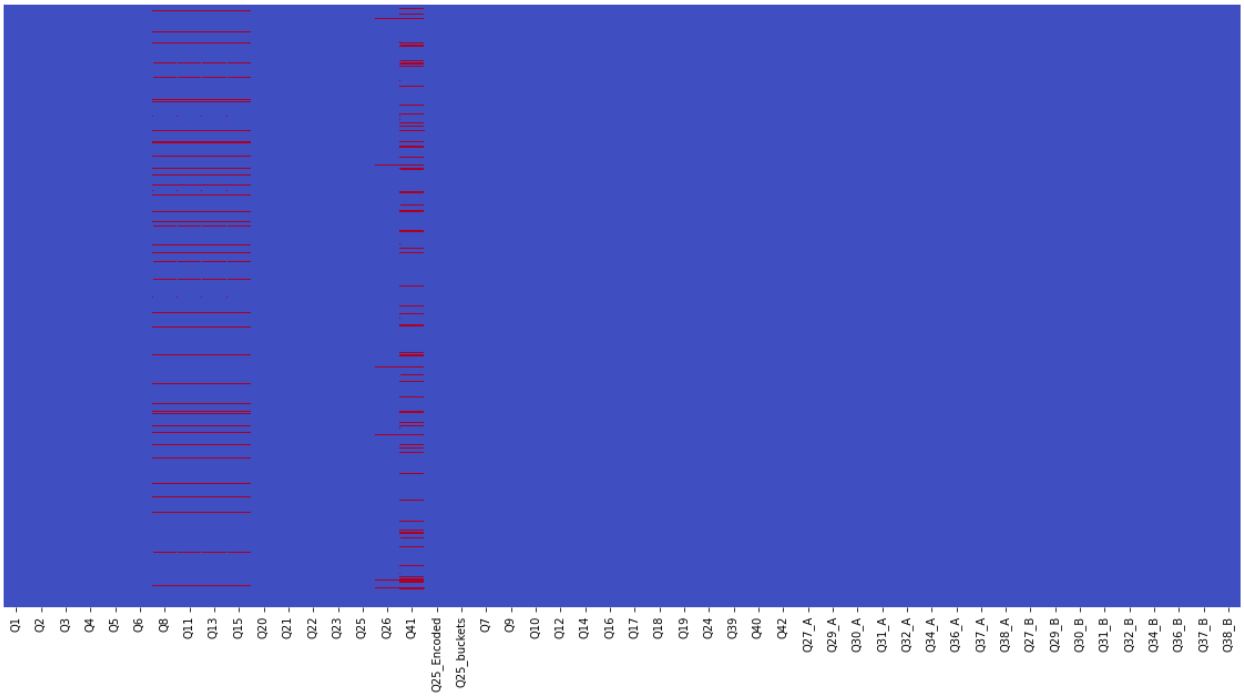
MIE1624 Assignment 2

Data cleaning

Many of the missing value appears because they are a series of choices of one certain question and attendees may just choose one or multiple of them.

I decide to encode each choice inside of a question with 0 and 1, and then sum them up horizontally for each question, to see how many choices the attendee has made for each of this kind of questions.




```
['Q8', 'Q11', 'Q13', 'Q15', 'Q26', 'Q41']
```

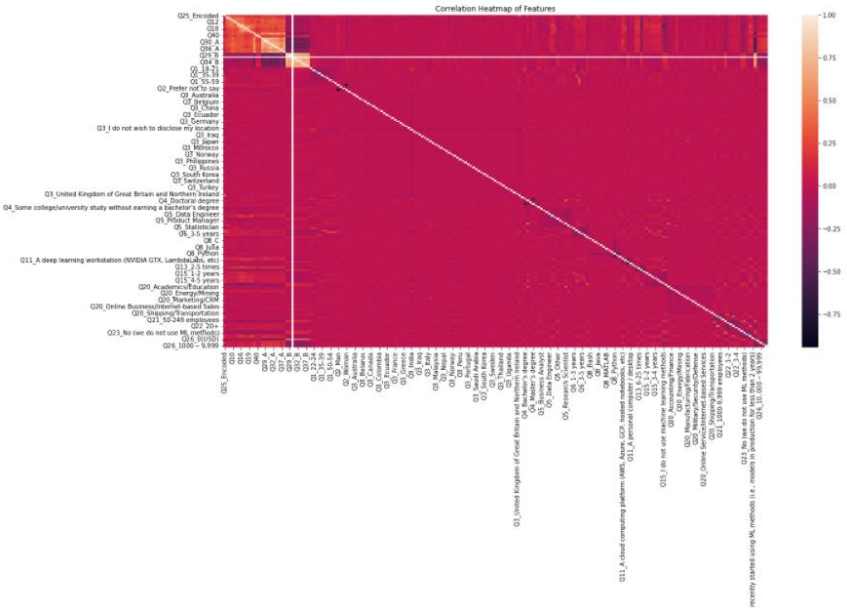
After summing up all the choices, according to the null values heat map, the null values mainly come from the questions above.

Since Q41 is about what editor you use to code, it is not influential to the income. Hence, we directly drop Q41.

For the rest of the questions that has null values, because none of them has the null values that exceed 10% of the column and all of them are categorical columns, we just fill the null values with the most common values in each of the column.

Secondly, we need to encode all the categorical columns into numerical values since scikit-learn only takes numerical values as input in a numpy array. Due to many of the categories don't necessarily have ordered relationships, we choose to encode categorical values using dummy variable by using onehotencoding. Eventually, after dropping the "Q25" and "Q25_buckets", we have 201 features in the data frame.

Exploratory data analysis and feature selection



Based on the correlation heat map, in the row of “Q25_Encoded”, we can see only a small part of features that has some significant difference in the colour of normal red, which means only some of the features has high correlation in absolute value.

	Q25_Encoded
Q25_Encoded	1.000000
Q3_United States of America	0.482180
Q6_20+ years	0.263827
Q23_We have well established ML methods (i.e., models in production for more than 2 years)	0.256236
Q3_India	0.246152
Q15_5-10 years	0.231519

After sorting the “Q25_Encoded” correlation coefficient’s absolute value in the descending order, we select out the highest 5 variables. Among these 5 variables, we can see 2 of them are related to countries, which shows country has a strong correlation with the income bucket. The rest three are most about how many years of experience the attendees have in data science methods and programming, and how good the company is in applying data science.

After removing all the features that have “Q25_Encoded” correlation coefficient’s absolute value being smaller than 0.05, we still have 107 features left. Because too many features can increase the final variance in the regression, we need to use feature engineering to select 10 features that has the most influence on “Q25_Encoded”. In this case, we choose to use the Lasso regression to eliminate the features and only choose the top 10 variables that have the most contribution (have biggest slope coefficients), because it is known for putting slope coefficients to be 0 for those features that have low contributions. After using the 5 folders cross validation, and tuning of hyperparameter λ from 0.1 to 10, we get the top 10 features below.

```
Index(['Q1_25-29', 'Q20_Academics/Education', 'Q6_10-20 years', 'Q3_India',
       'Q1_22-24', 'Q3_United Kingdom of Great Britain and Northern Ireland',
       'Q6_20+ years', 'Q3_Australia', 'Q3_Germany',
       'Q3_United States of America', 'Q25_Encoded'],
      dtype='object')
```

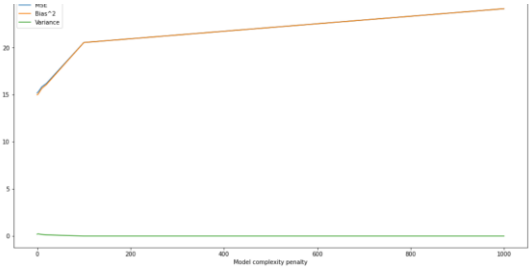
Model implementation

```
Fold 1: Accuracy: 81.797%
Fold 2: Accuracy: 81.981%
Fold 3: Accuracy: 80.702%
Fold 4: Accuracy: 82.551%
Fold 5: Accuracy: 82.424%
Fold 6: Accuracy: 85.167%
Fold 7: Accuracy: 80.958%
Fold 8: Accuracy: 81.051%
Fold 9: Accuracy: 81.34%
Fold 10: Accuracy: 82.105%
Average Score: 82.008%(1.467%)
```

Before we apply any model, the normalization is not necessary here, since all 10 features are binary variables.

For 10-fold cross-validation of logistic regression, the accuracy keeps being stable across the folds. The average accuracy is 82.008% and its variance is 1.467%.

```
Model 1 Score: 46.7
Model 2 Score: 47.199999999999996
Model 3 Score: 47.8
Model 4 Score: 47.599999999999994
Model 5 Score: 48.1
Model 6 Score: 48.1
Model 7 Score: 48.1
Model 8 Score: 48.1
Model 9 Score: 48.1
```




```
c = [0.001,0.01,0.05,0.1,0.5,1,5,10, 100]
```

We treat each value in the list c as a hyperparameter for C with a new model. C represents the inverse of regularization strength. In other words, C means the inverse of penalty on model complexity. The x-axis in the graph is the value of model complexity penalty, the green line represents variance, and the orange line represents the bias. According to the graph, it is obvious to see around 1 at x-axis, there is a decrease in variance and a trend of increase in bias. Hence, at that point, model becomes more fit to the test data and less fit to the train data. Since model complexity penalty is calculated by 1/C, so C=1 when model complexity penalty is 1. The fifth model uses C=1, and it does have one of the highest scores among all the models.

Model tuning

All hyperparameters in logistic regression is solver types, penalty types and C values. In this model tuning, we choose to tune the solver types and C values by using all the possible combinations of them to seek out the logistic regression model that has the highest accuracy. The reason to choose those two hyperparameters is that C value can help in doing the bias-variance trade-off, and solver type can also help as some of the solvers are better at solving small datasets and multiclass problems. I choose the accuracy as the performance metric because it is a very direct sign of how good the model is in making predictions.

```
{'C': 0.001, 'solver': 'newton-cg'}
Best Score: 82.139%(1.317%)
```

```
The optimal log model uses C=0.001, and a newton-cg solver, and has a cross validation score of 82.139% with a standard deviation of 1.317%
```

	Coefficient
Q3_United States of America	0.392790
Q1_22-24	0.334078
Q20_Academics/Education	0.298670
Q3_India	0.289850
Q6_10-20 years	0.276105
Q6_20+ years	0.251366
Q1_25-29	0.171701
Q3_United Kingdom of Great Britain and Northern Ireland	0.106282
Q3_Germany	0.074137
Q3_Australia	0.050048

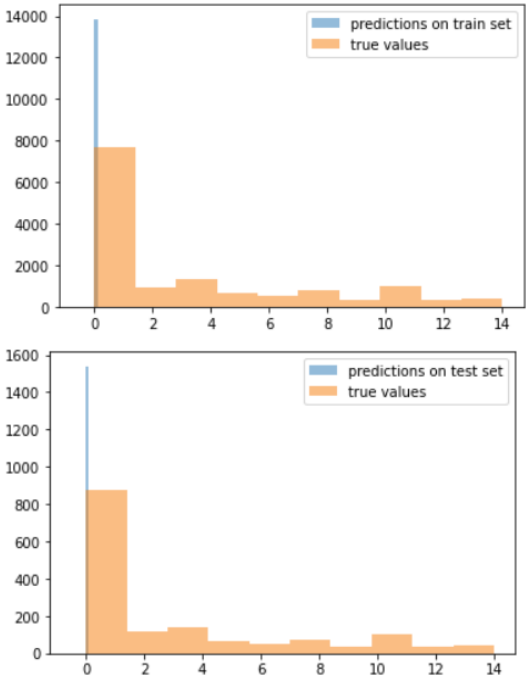
	Q25_Encoded
Q3_United States of America	0.482180
Q6_20+ years	0.263827
Q3_India	0.246152
Q1_22-24	0.208221
Q6_10-20 years	0.200405
Q20_Academics/Education	0.167811
Q1_25-29	0.153092
Q3_Australia	0.132294
Q3_Germany	0.112712
Q3_United Kingdom of Great Britain and Northern Ireland	0.108776

After finding out the best model, we use the model's coefficient's absolute value as the sign of feature importance, and sort them in the descending order as the left graph shows above. We also select out the 10 features' descending order in correlation's absolute value from section 2 as the right graph shows above. In comparison of those two tables, though we can see there are changes in the positions, but the last 4 remains to be last 4, and "Q3_United States of America" remains to be the first.

Testing & Discussion

```
This model got an accuracy of 82.11% on the training set
This model got an accuracy of 82.34% on the testing set
```

After using the best model we get from the last section, it performs almost the same in both training set and test set. This indicates that the model is definitely not overfitting, but underfitting since 82% is not a good accuracy. The method to increase the accuracy is to add more independent variables to make more contributions in prediction of the target variable.



From the histogram of model's predictions in train set and test set, we discover that most of the predictions are salary bucket 0 in both sets.

In this case, to increase the accuracy, we need to add more independent variables that can make significant contributions in prediction salary buckets other than salary bucket 0.