1、 Data Cleaning:

Firstly, I look at the data overall and find all the variables(columns) are categorical data so I decide to convert them to dummy variables. There are two kinds of columns which are single choice and multiple choice. Deal with the single choice first.

From the figure below we can find there is no nan value in Q1-Q6, so we directly turn them to dummy variable.

```
Q1    0
Q2    0
Q3    0
Q4    0
Q5    0
Q6    0
dtype: int64
```

Then look specifically into the Q2, 'Prefer to self-describe' and 'Nonbinary' choices have too small number compared to the total amount of data(about 10000), and I think those data will not be representable so delete rows with those two answers.

```
Man                        8872
Woman                      1683
Prefer not to say           131
Prefer to self-describe      23
Nonbinary                    20
Name: Q2, dtype: int64
```

Variables below are single choice with nan values. For Q8,Q11,Q13,Q15,Q25, the number of missing values is not much so we drop the rows with nan. As for why I didn't replace them with mode or something is I think dropping them will have less influence for the data than replacing because of the small number. For Q30,32, I just drop the two columns because too many values missing. As for Q38, I replace the nan with mode because mode has a dominate number and nan is more than Q8-Q25. After dealing with nan values, I still convert them to dummy variables.

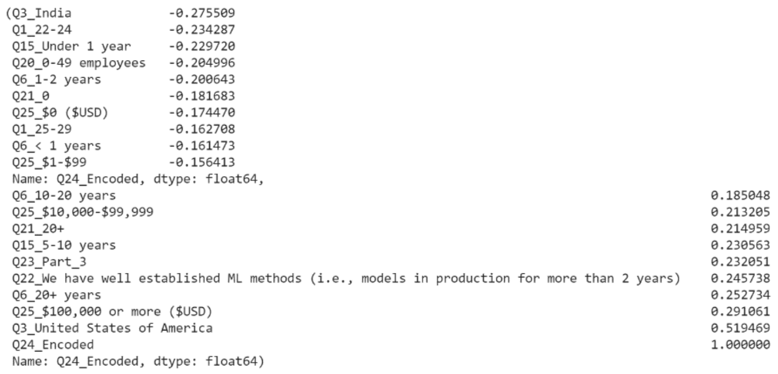
```
Q8      558
Q11     558
Q13     558
Q15     558
Q20       0
Q22       0
Q25     159
Q30    7186
Q32    9192
Q38    1250
dtype: int64
```

For the multiple choice columns, one column just contains one data value or nan. So by turning the value to 1 and nan to 0, we can get dummy variables without get_dummies function. And we still need to clean the nan value. One question with all the parts are nan which means the tester didn't choose any choice of the question are considered nan. After computing we find all questions with A and B part and Q18,Q19,Q36 have too many nan values so they should all be dropped. Q16,17,37,39 do not have not many values missing so we drop the rows. Other columns do not have nan value.

After all the proceeding, we should drop columns which don't have nan value but have no 1 value because of the delete operation before. They are meaningless columns.

2、 Exploratory data analysis and feature selection:

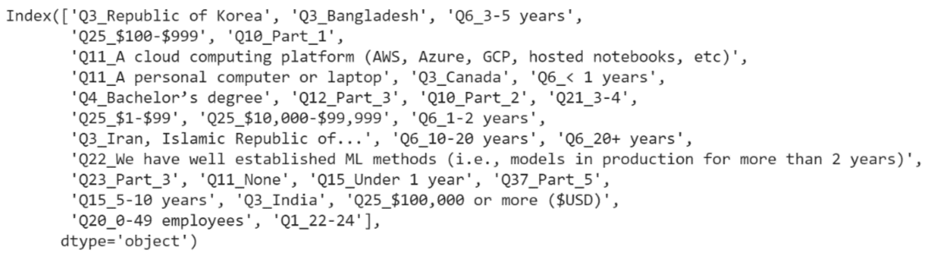
Using correlation matrix to see the feature importance. Print the top ten and last ten features of column Q24, we can see Q3_America and Q3_India are on the two sides. Combining the analysis in assignment 1, I think country is most related to the salary.



```
(Q3_India            -0.275509
 Q1_22-24            -0.234287
 Q15_Under 1 year    -0.229720
 Q20_0-49 employees  -0.204996
 Q6_1-2 years        -0.200643
 Q21_0               -0.181683
 Q25_$0 ($USD)       -0.174470
 Q1_25-29            -0.162708
 Q6_< 1 years        -0.161473
 Q25_$1-$99          -0.156413
 Name: Q24_Encoded, dtype: float64,
 Q6_10-20 years                                                                      0.185048
 Q25_$10,000-$99,999                                                                 0.213205
 Q21_20+                                                                             0.214959
 Q15_5-10 years                                                                      0.230563
 Q23_Part_3                                                                          0.232051
 Q22_We have well established ML methods (i.e., models in production for more than 2 years)   0.245738
 Q6_20+ years                                                                        0.252734
 Q25_$100,000 or more ($USD)                                                         0.291061
 Q3_United States of America                                                         0.519469
 Q24_Encoded                                                                         1.000000
 Name: Q24_Encoded, dtype: float64)
```

Then we should use feature selection to drop some features. When there are too many features in the dataset, some features have little correlation the target and they can only slow down the computing speed and decrease the accuracy. Some features have so little information which make the dataset a sparse matrix and they are also useless. That’s why feature selection is important.

Firstly, I use mutual information regression. MI is a non-negative value and measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. It’s a good way to show the sequence of data importance. Below are the best 29 features with higher importance I choose. They have highest MI value.



```
Index(['Q3_Republic of Korea', 'Q3_Bangladesh', 'Q6_3-5 years',
       'Q25_$100-$999', 'Q10_Part_1',
       'Q11_A cloud computing platform (AWS, Azure, GCP, hosted notebooks, etc)',
       'Q11_A personal computer or laptop', 'Q3_Canada', 'Q6_< 1 years',
       'Q4_Bachelor’s degree', 'Q12_Part_3', 'Q10_Part_2', 'Q21_3-4',
       'Q25_$1-$99', 'Q25_$10,000-$99,999', 'Q6_1-2 years',
       'Q3_Iran, Islamic Republic of...', 'Q6_10-20 years', 'Q6_20+ years',
       'Q22_We have well established ML methods (i.e., models in production for more than 2 years)',
       'Q23_Part_3', 'Q11_None', 'Q15_Under 1 year', 'Q37_Part_5',
       'Q15_5-10 years', 'Q3_India', 'Q25_$100,000 or more ($USD)',
       'Q20_0-49 employees', 'Q1_22-24'],
      dtype='object')
```

I also use some other feature selection algorithms here to see which is better. I use chi2 feature selection and PCA to select features and prepare for step 3 to train the model.

3、 Model implementation:

Model implementation is actually a step interactive with step2. Using ten-fold validation every time when I finish one feature selection method to see the difference. From the results we can find the four feature selection methods have similar results so we choose the most intuitive one-MI regression.

Here I use ski-learn logistic regression model to implement ordinal logistic regression because the ordinal logit regression model packages I found are not stable and have poor performance. I used a for loop with logit regression in each step. In each step, I split the X and y data to two classes. Like firstly, original class0 as class0 and original class larger than 0 as class1 and so on. And by looping for 15 times, we can get a result of 15 groups of binary classes. Then we can get probabilities for each group.

The mean accuracy for 10-fold validation is about 0.39, and the variance is about 0.0004. The accuracy is low and I'll discuss it later. The variance is low too which shows there is not much difference across the folds.

There are actually not many important hyperparameters in logit regression. I simply choose solver and C here. Solver is the method for optimizing and C is the inverse of regulation strength. Using different solver and C value doesn't have much influence on the results.

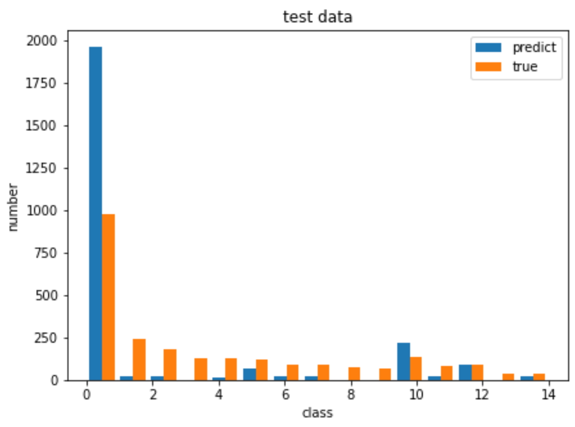
4、 Model tuning:

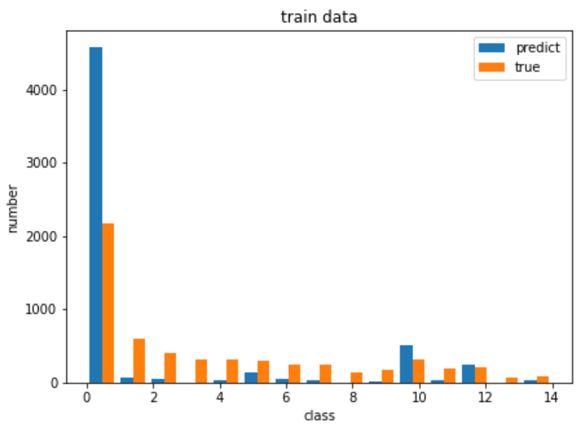
For LR model, some important hyperparameters are penalty method, C value, solver. Max_iter also should be set higher in case it can not run to the final results. Penalty method specify the norm of penalty which may decrease the variance. Small C value may also be helpful to the bias-variance trade off. A good solver can improve the performance of the classifier. Then use the grid search method to find best hyperparameter combination for the highest metrics. Here I use f1_score as the metric. F1 score is a metric that can reflect the result of not only major class but smaller class and a balance between precision and recall. Finally, the accuracy of mean 10-fold validation is about 0.38-0.39.

5、 Testing & Discussion:

Finally, we can put test data into our model. The final accuracy is about 0.40 and test data is a little bit higher than train data in cross-validation. It's not overfitting obviously and I also don't think it's underfitting. Because it's a multiclass classification and as long as a final prediction is close to the true value, it should not be regarded as bad result. So 40% may not imply the underfitting. Figures below are the distributions of prediction and true value of train and test data.

From the figure we can find out we predict too many targets as class 0 and too few as 1-10. That may because the values of features under class 10 are similar and logistic regression is not complex enough to classify them. Another reason I think is important is the one hot encoding of all the categorical features may cause a information loss. It separate one feature to many features which some of them may be harmful to the model training. If we can use the categorical data directly like using decision tree or forest, I think we can improve the performance of the model.


test data
predict
true
number
class


train data
predict
true
number
class