1 Data Cleaning

In this part, I first filling all the “null” with “unknown”, first of all, I will encode the categorical column by logic. For example in Q16, it asks the participant how long they have been using machine learning, and my encoding process will be like

```
experience_encode = {   'Under 1 year': 1,
    '1-2 years': 2,
    'I do not use machine learning methods': 3,
    '2-3 years': 4,
    '5-10 years': 5,
    '4-5 years': 6,
    '3-4 years': 7,
    '10-20 years': 8,
    'unknow':'unknow'
}
response2["Q16"] = response2["Q11"].map(Q,experience_encode)
```

after converting all the data in to ordinal, we can filling the ”unknown” with the most frequent term in that column, for example: in Q16 there are 684 ”unknown” and the most frequent encode is ”1”,therefor I replace all the ”unknown” with ”1”. For some binary data, if the data is missing, it means the participants did not select this, so I will fill it up with ”0”.By far the data cleaning process is completed

2 Exploratory data analysis and feature selection

Feature engineering is important in this assignment because the data set that has been given to us has the shape 8136*297, which means it has 297 features, that’s a lot but some of the features will mot contribute to improve the model performance on predicting, if we do not perform the feature selection, the model will waste a lot of resources on useless feature. Therefore we have to use some method to identify which feature is important and which are not. In my case, I am going to identify them based on p_value. Since in the classification requirements, it specifies that the subset of data chosen must contain at least 5000 training examples, and the train set is usually 70 percent of the original data which means each original column data should contain at least 7200 data. Therefor, in my first step, I will check the number of “unknown” in each column and drop the column if it doesn’t maintain the requirement. Then we can check the correlation among each feature, p-value comparison or importance histogram.Since those data are categorical, so we can apply chi-square test on those data to test there Independence. If features are independent with the target value, which means it will not affect the prediction result, so those features should be dropped. From the correlation heat map, we can know that salary is largely depend on the Q11 which is how many years participants has been coding.

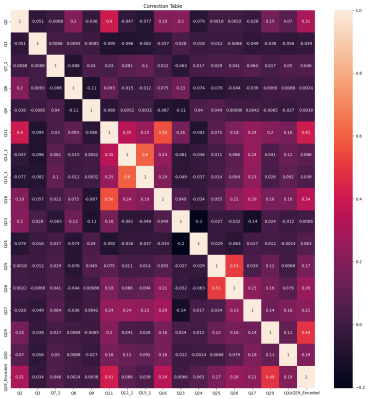


Figure 1: Correlation Heat Map

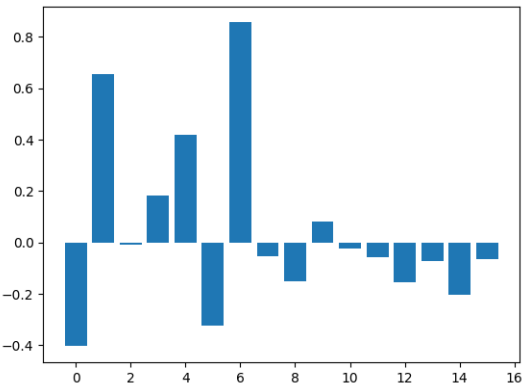
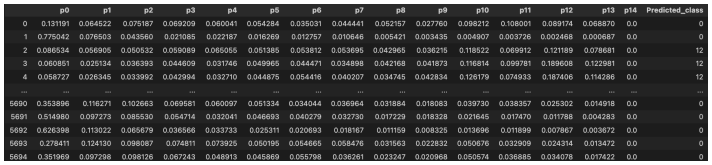


Figure 2: Importance Histogram

3 Model Implementation

Since after the data cleaning and feature selection.The salary bucket is ordinal data with 15 classes, so in order to implement the model, I set up 14 different binary classification model. For example, the first model is set salary bucket 0-9999 to encode 1 and all the other is zero and second model is set 0-9999 and 10000-19999 to 1 and all the rest are 0. After apply Logistic Regression Function on each model, we can get the probability by using predict_prob(), and start from

second model, we should use the returned probability to subtract the previous probability (probability in first model), since the first model has been already included in the second model. By iterating this process, we will get the probability of belonging to each of the salary bucket. The predicted result on the training set is shown below: p0 is the probability



	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	Predicted_class
0	0.131191	0.064522	0.075187	0.069209	0.060041	0.054284	0.035031	0.044441	0.052157	0.027760	0.098212	0.108001	0.089174	0.068870	0.0	0
1	0.775042	0.076503	0.043560	0.021085	0.022187	0.016269	0.012757	0.010646	0.005421	0.003435	0.004907	0.003726	0.002468	0.000687	0.0	0
2	0.086534	0.056905	0.050532	0.059089	0.065055	0.051385	0.053812	0.053695	0.042965	0.036215	0.118522	0.069912	0.121189	0.078681	0.0	12
3	0.060851	0.025134	0.036393	0.044609	0.031746	0.049965	0.044471	0.034898	0.042168	0.041873	0.116814	0.099781	0.189608	0.122981	0.0	12
4	0.058727	0.026345	0.033992	0.042994	0.032710	0.044875	0.054416	0.040207	0.034745	0.042834	0.126179	0.074933	0.187406	0.114286	0.0	12
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5690	0.353896	0.116271	0.102663	0.069581	0.060097	0.051334	0.034044	0.036964	0.031884	0.018083	0.039730	0.038357	0.025302	0.014918	0.0	0
5691	0.514980	0.097273	0.085530	0.054714	0.032041	0.046693	0.040279	0.032730	0.017229	0.018328	0.021645	0.017470	0.011788	0.004283	0.0	0
5692	0.626398	0.113022	0.065679	0.036566	0.033733	0.025311	0.020693	0.018167	0.011159	0.008325	0.013696	0.011899	0.007867	0.003672	0.0	0
5693	0.278411	0.124130	0.098087	0.074811	0.073925	0.060195	0.054665	0.058476	0.031563	0.022832	0.050676	0.032909	0.024314	0.013472	0.0	0
5694	0.351969	0.097298	0.098126	0.067243	0.048913	0.045869	0.055798	0.036261	0.023247	0.020968	0.050574	0.036885	0.034078	0.017422	0.0	0

Figure 3: Predicted Result

belonging to salary encode 1 which is belong to the bucket 0-9999. The Logistic Regression do require the normalization to avoid the vanishing gradient problem in the training phase. After 10-fold cross validation, the accuracy is quiet stable and vary between 83.73% and 76%, and the average accuracy of 10-fold cross validation is 80.185% with the variance 2.63%

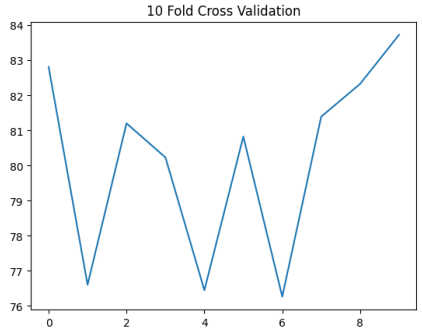


Figure 4: 10 fold cross validation result

4 Model Tuning

The hyper-parameter of logistic regression are penalty, inverse of regularization strength C and solver respectively. Two hyper-parameter that I have chosen for this part are solver and inverse of regularization strength C, the reason why I did not tune the penalty is because some penalties may not work with some solvers, for example: solver "newton-cg" could not work with the penalty "l1". The method I used to tune the hyper-parameter is called Grid Search. To implement this, I have tested different C with different solver, and perform 10-fold cross-validation on the model I built in the previous part with the chosen hyper-parameter, in each fold, I compute the F1 Score and cross validation score, then take the average of them after 10-fold. To choose the optimal model, the model which has the highest F1 Score and Cross Validation Score will be taken. The reason why I did not choose the model based on the accuracy is because the data set that given to us is a unbalanced labeled data, the participant belong to the bucket 0-999 is way much more than the other bucket. If only use the accuracy as the metric, even all the result is predicted as bucket 0-9999, the model still maintain around 30% accuracy. The optimal model that produced by Grid Search is when C = 0.1, solver is "liblinear" which have will cross validation score 80.214% and the F1 Score 0.058 by comparing the importance histogram, we can see that before feature selection, the feature importance have both negative and positive, and after feature selection, the importance are all negative

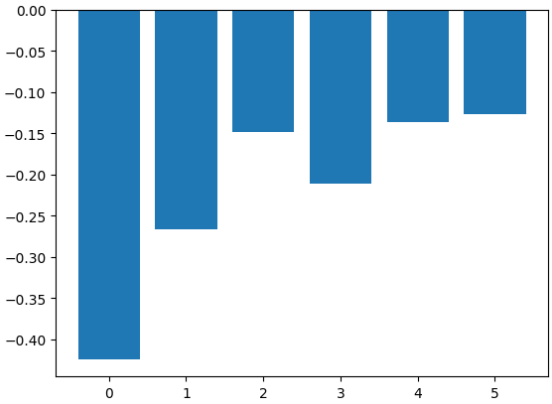


Figure 5: Importance Histogram

5 Testing and Discussion

By applying the most optimal model to the testing data, the accuracy achieves only 36.788%, and looking into the plot of distribution of the true target variable on test and train data, we can see that they do not have much difference, the model perform poorly on both training data and testing data, the majority of prediction are salary encode 0 which correspond to salary bucket 0-9999. Therefore, we can conclude that the model is under-fitting, because the accuracy is pretty low both on training and testing data The reason why the model perform poorly may have several reasons. First of all, the survey is not accuracy, some participant may not answer it based on their true personal information. Secondly, the data collected is a very unbalanced, we can see from the true target, participant fall in the salary range from 0 -9999 occupied majority, and participant who earns more than 300000 is very rare, the unbalanced data will increase the difficult for the model to correctly predict. My advice to improve the performance is change the model, for this type of model, Decision Tree Classifier seems more suitable than Logistic Regression Model.

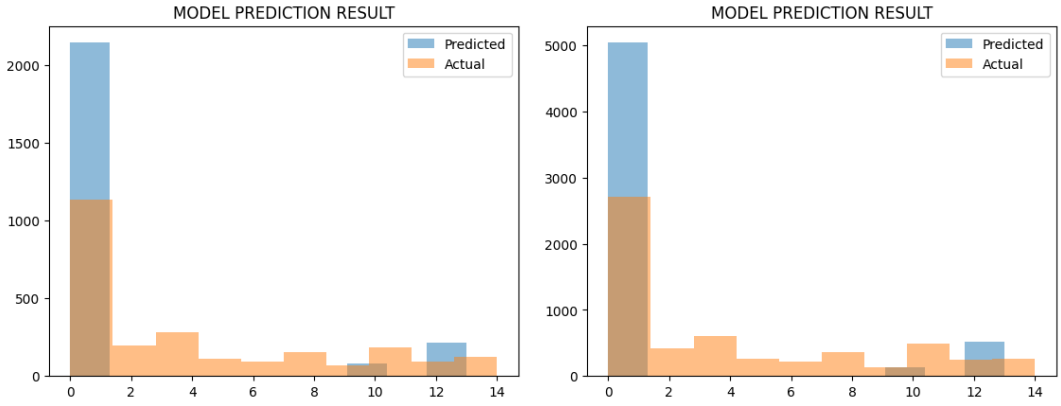


Figure 6: Predict Result On Test(L)/Train(R) Set