MIE 1624 Assignment 2 **Yao Tong #1007410295**

Question 1: Data cleaning: First, Based on the report of missing data heatmap and null values list. Most of the features in this original data frame have over half data are null. Except for the simple question feature such as Q1 and Q2, other features can be each choice in a multiple-choice question such as question 7 which support 13 features in this dataset. Also, most of these features owned very high null values such as Q7_Part_12 has 10626 null values, because people may select the other choices in this question. This might be a big challenge in data cleaning and feature selection. Therefore, we design to ignore these features currently. However, the multiple-choice question features also have a connection with the target feature Q24. The model prediction accuracy might impact by these features missing.

Then, based on the sorted null value list, the simple question features with small null space number. I select the following features for the next data cleaning step: Q1, Q2, Q3, Q4, Q5, Q6, Q8, Q11, Q13, Q15, Q20, Q21, Q22, Q25, Q38, and the target feature column Q24_Encoded. In addition, these simple question features got a high connection with the target feature Q24, and they will be used to do feature engineering and feature selection process.

For the next data cleaning step: feature engineering process. Based on the selected features, I split them into non-null features and features with null spaces. Non-null features: Q1, Q2, Q3, Q4, Q5, Q6, Q20, Q21, Q22. Features with null spaces: Q8, Q11, Q13, Q15, Q25, Q38.

For both two-part features, I select two different feature engineering methods on categorical data: label encoding and dummy coding. For example, feature Q1 has 11 types of years level which increases from 18 years to over 70 years. I use label encoding to label them into two new features named Q1_encoded and Q1_buckets. Q1_encoded has 11 categorical labels from 0 to 10 which can be used in future data analyzation. As same as Q1, I encoded the features: Q6, Q20, Q21, Q13, Q15, Q25 using the label encoding method to convert categorical data into numerical data.

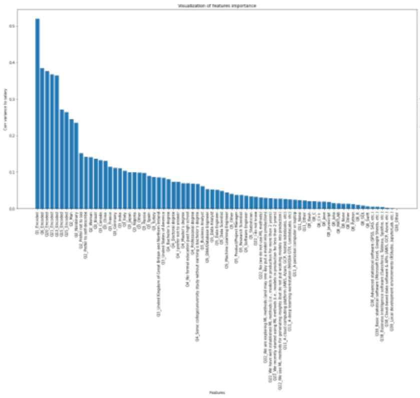
For another type of encoding method: dummy encoding. feature Q3 has categorical data with different countries. I use dummy encoding to create dummy features for categorical data in this feature. However, in Q3 there are too many countries, if dummy encodes all categorical data will increase the programming workload. Based on the report of the country counts list (shows in Jupyter notebook). I build dummy features for countries whose counts over 150, the other categorical country data build a feature named other. Like the feature Q3, I encoded the features: Q2, Q4, Q5, Q22, Q11, Q8, Q38 using label encoding method to convert categorical data into numerical data.

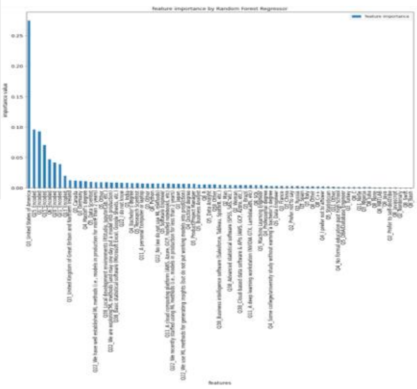
For features with null spaces, filling the missing value is challenging. During this assignment, I select two main methods to fill null spaces. The first way is filling the missing values with a certain value that has maximum possibilities. For example, in feature Q13, based on the result of Q13 and Q24_Encoded relationship plots (shows in Jupyter notebook). I design to fill the Q13 null spaces with the value “Never”. Because most of the categorical data in feature Q13 is “Never”. Like the feature Q13, I filled the features: Q8 and Q38 using label encoding method to convert categorical data into numerical data.

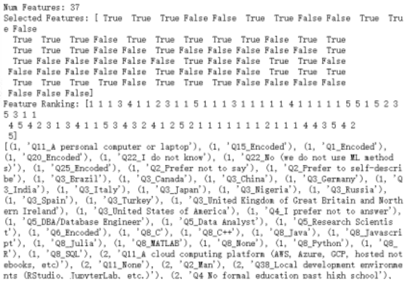
The second way is building a stepped label function when the main salary label of different categorical label is different. (the three types of relationship plots show in Jupyter notebook) Then use the function to fill all null spaces. I filled the Q11, Q15, and Q25 features in this way.

The last data cleaning step is a combination of all encoded features. The cleaned data frame includes 76 columns and no null spaces. The heatmap plot shows in Jupyter notebook.

Question 2: Exploratory data analysis: Do analyze the cleaned data with describe function. Then report the order list of feature importance in the cleaned data frame, and plot feature importance image (with .corr() function) to do feature analyzation (Figure 1). The results show feature Q1_encoded has the highest feature importance through all features in this model.







```
Num Features: 37
Selected Features: [ True  True  True False False  True  True False False  True  Tru
e False
  True  True  True False  True  True  True  True  True False  True  True
  True  True  True False False  True False False False False False  True
  True False False False False False  True False False  True  True False
 False False False False False  True False False False  True  True  True
  True  True  True  True False  True  True  True False False False False
 False False False]
Feature Ranking: [1 1 1 3 4 1 1 2 3 1 1 5 1 1 1 3 1 1 1 1 1 4 1 1 1 1 1 5 5 1 5 2 3
5 3 1 1
 4 5 4 2 3 1 3 4 1 1 5 3 4 3 2 4 1 2 5 2 1 1 1 1 1 1 1 2 1 1 1 4 4 3 5 4 2
 5]
[(1, 'Q11_A personal computer or laptop'), (1, 'Q15_Encoded'), (1, 'Q1_Encoded'),
(1, 'Q20_Encoded'), (1, 'Q22_I do not know'), (1, 'Q22_No (we do not use ML method
s)'), (1, 'Q25_Encoded'), (1, 'Q2_Prefer not to say'), (1, 'Q2_Prefer to self-descri
be'), (1, 'Q3_Brazil'), (1, 'Q3_Canada'), (1, 'Q3_China'), (1, 'Q3_Germany'), (1, 'Q
3_India'), (1, 'Q3_Italy'), (1, 'Q3_Japan'), (1, 'Q3_Nigeria'), (1, 'Q3_Russia'),
(1, 'Q3_Spain'), (1, 'Q3_Turkey'), (1, 'Q3_United Kingdom of Great Britain and North
ern Ireland'), (1, 'Q3_United States of America'), (1, 'Q4_I prefer not to answer'),
(1, 'Q5_DBA/Database Engineer'), (1, 'Q5_Data Analyst'), (1, 'Q5_Research Scientis
t'), (1, 'Q6_Encoded'), (1, 'Q8_C'), (1, 'Q8_C++'), (1, 'Q8_Java'), (1, 'Q8_Javascri
pt'), (1, 'Q8_Julia'), (1, 'Q8_MATLAB'), (1, 'Q8_None'), (1, 'Q8_Python'), (1, 'Q8_
R'), (1, 'Q8_SQL'), (2, 'Q11_A cloud computing platform (AWS, Azure, GCP, hosted not
ebooks, etc)'), (2, 'Q11_None'), (2, 'Q2_Man'), (2, 'Q38_Local development environme
nts (RStudio, JupyterLab, etc.)'), (2, 'Q4 No formal education past high school'),
```

Figure 1 Feature importance visualization Figure 2 Feature importance (RFR) Figure 3 Recursive Feature Elimination report

Feature engineering: In this assignment, feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. feature engineering can increase the predictive power of machine learning algorithms by creating features from raw data. In assignment data, most columns filled with categorical data, and the feature engineering methods on categorical data are very important in this assignment. I created the encoded, buckets, and dummy features and remove the multiple-choice features. As the previous discussion part, I used two encoding methods: label and dummy.

Feature selection: In this assignment, I chose two methods to select features. Method 1 reports feature importance by Random Forest Regressor model. The feature importance graph (Figure 2) shows feature Q3_United States of America has the highest feature importance through all features in this model, which is different from the .corr() method. Method 2 reports feature importance list and select features by Recursive Feature Elimination (RFE) model. The feature selection report shows in Figure 3 shows the selected features with value “1” in the Feature Ranking array.

The selected features of these two methods are different, based on the confrontation between these two results and my understanding of the features connection. I selected the following features as future analyzation features in this assignment: Q1_Encoded, Q15_Encoded, Q20_Encoded, Q25_Encoded, Q6_Encoded, Q21_Encoded, Q13_Encoded, Q3_United States of America, Q3_United Kingdom of Great Britain and Northern Ireland, Q3_Canada, Q3_Germany, Q5_Data Analyst, and Q4_Master’s degree.

Question 3: Model implementation: Before the model implementation and model tuning, I combine all selected encoded features as a new data frame called: selected_feature. This data frame has 14 columns. I split train and test data frame from the seclected_feature data frame, then split train and test data frame (Xs_train, Xs_test, ys_train, ys_test) from the selected_feature_train data frame. Then, rescaling values with scaler function.

For model implementation: I build an ordinal logistic regression model with cross-validation which Kfold value is 10. For the accuracy calculation through the cross-validation loop, I used the accuracy classification score function: accuracy_score() to compute subset accuracy in multilabel classification. The model output shows the accuracy value across each fold. The mean cross-validation accuracy is 0.431, standard Deviation value is 0.0112, and the variance value is 0.000125. The result shows screenshot shows in Figure 4.

For hyperparameters selection. I build a parameter dictionary called parameter, which includes the list of hyperparameters C, penalty types and solver types. Input the model GridSearchCV for selecting hyperparameters. Then treating each value of hyperparameters as a new model to select the best model. This part combined with the model tuning part in question 4. The best model hyperparameters selection result is {'C': 0.04281332398719394, 'penalty': 'l2', 'solver': 'newton-cg'}, and the train data frame (Xs_train) accuracy is 0.4398, the test data frame (Xs_test) accuracy is 0.4356.

The principle of best model selection based on the bias-variance trade-off method. In this ordinal logistic regression algorithm, the tuning hyperparameters C or called model complexity is inverse of regularization strength "lambda". The prediction error of logistic regression algorithm can be broken down into bias error, variance error and irreducible error. The trade-off can be changed by increasing the C parameter value, and the bias error is increasing but the variance error is decreasing. When getting the minimum total error of this logistic regression algorithm, report the best hyperparameters C value. For example, during this step, the lowest algorithm total error when C is 0.04281332398719394.

Question 4: Model tuning: During this model tuning part. I selected three hyperparameters in my model which include C, penalty, and solver. Select a final optimal model through previous steps but using grid search based on different metrics. In this assignment, I used 4 different metrics to report optimal model's parameter and prediction accuracy as shown in the below table.

	C	penalty	solver	train accuracy	test accuracy
accuracy	0.042813324	l2	newton-cg	0.439857967	0.435623058
precision_macro	0.112883789	l2	newton-cg	0.441041574	0.438039351
f1_macro	8.858667904	l1	liblinear	0.436307146	0.428028996
Recall_macro	0.483293024	l2	newton-cg	0.441189525	0.437348982

Based on the results from four models. The highest train accuracy model is when using recall metric with average: 'macro' which is "scoring = 'recall_macro'" in programming. 'macro' means the recall function will calculate metrics for each label.

Question 5: Testing & Discussion: After make classification on the test set with optimal model. The train accuracy 0.441189525 and test accuracy 0.437348982. The original algorithm's prediction accuracy on train set is 0.440893623, the selected parameters increase 0.002 (0.2%) prediction accuracy.

The overall fit of the model, the prediction accuracy can increase by changing the hyperparameter C value, as shown in Figure 4. In addition, the model prediction accuracy on test set will increase to fit the train set accuracy when the training example values increase as shown in Figure 6. The Figure 6 also shows the cross-validation score value (test set accuracy) always lower than the training score. Therefore, this model is underfitting, which means this model not tunned to be the best yet. The distribution of true target variable values and their predictions on both the training set and test set shows in Figure 5. Based on this figure, the prediction results of salary level 10 are too high. That may cause by the missing features at the data cleaning part and features selection part. From this assignment. I awareness of the importance of previous data cleaning, null value filling and features selection. These parts will increase the basic model prediction accuracy. Garbage in garbage out!

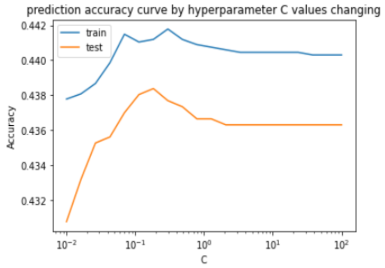


Figure 4 Prediction accuracy curve

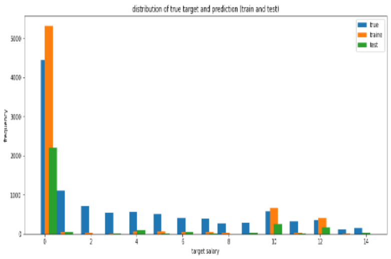


Figure 5 distribution of true target and prediction

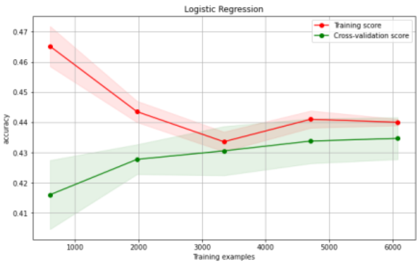


Figure 6 Learning curve