MIE1624 Assignment 2

Part 1: Data Cleaning

In the initial dataset, we began with 8137 rows and 298 columns, which included the encoded "Salary." After removing the first row to eliminate the survey question, we categorized the columns into those with null values (sub-columns with "_" indicating the choice not made by observations), null values with sub-columns, and non-null values. We then cleaned the columns with null values by dropping those with more than 50% null values, filling nulls based on value distribution, and encoding ordered and binary columns. For columns with null values and sub-columns, we replaced nulls with 0 and non-nulls with 1. In columns without null values, we removed uninformative columns and encoded the remaining ones, considering their binary or ordered nature. As a result, we obtained a final dataset consisting of 8136 observations, 285 features, and 3 target columns: "Q29," "Q29_Encoded," and "Q29_buckets." It's worth noting that while filling null values based on value distribution ensures data completeness, it carries the potential drawback of introducing bias that could impact model performance.

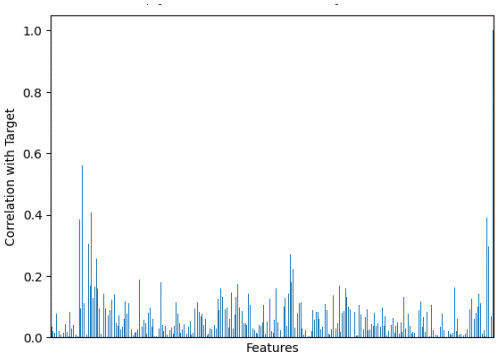
Part 2: Exploratory Data Analysis (EDA) and Feature Selection

In this part, we perform exploratory data analysis and feature selection to our cleaned dataset. Our goal is to select insightful features for our model.

Insights from Summary Table:

- The mean values for binary columns provide insightful information. Each mean presents the proportion of participants who selected the answer that this column represents.
- The median of **Q29_Encoded** is 2 meaning 50% participants' salaries land in buckets larger than 2. Therefore, our dataset is probably unbalanced.

Visualize correlation (Feature Importance) and Feature Selection:



- We observe that not all feature columns have relatively strong correlations with our label column **Q29_encoded**. We dropped features having correlation with **Q29_encoded** under 0.1. Only 58 features remaining.
- **Insights from top 3 features**: years of machine learning method experience, years of writing code, and whether the surveyor is based in North America tend to impact salary.

Justification of Feature Engineering and Feature Selection:

In our previous data cleaning steps, we employed two key techniques for feature engineering on categorical features: **One-Hot Encoding** and **Label Encoding**. These techniques augment our model with additional features, potentially bolstering prediction accuracy. Furthermore, we applied **Binning** to categorize participants' citizenships by continents, simplifying the model's complexity.

We also leveraged **correlation analysis** to gauge feature importance and guide our feature selection process. Correlation quantifies the relationship between features and label variables, making features with strong correlations crucial for our model. However, it's essential to remember that low-correlation features are not necessarily unimportant. They may play a vital role when combined with other features.

Part 3: Model Implementation

In this section, we apply ordinal logistic regression with 10-fold cross-validation on a 70/30 train-test split. Recognizing the **disparity** in feature scales from **EDA**, we address this by **rescaling** our features, a standard practice in machine learning.

Implement models and Make prediction:

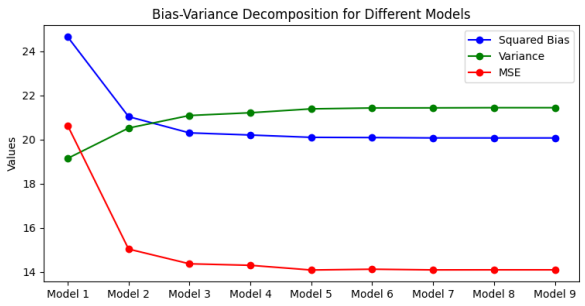
Our ordinary multi-class classification model is built upon binary logistic regression. The probabilities for each ordinal category relative to the baseline category. The probability of each ordinal category is represented by the difference in the probability of its baseline category to the previous baseline category.

Evaluation with 10-fold Cross-Validation:

An accuracy of 0.43 may indicate subpar model performance, but it's not conclusive due to the training dataset's influence. To mitigate this, we used 10-fold cross-validation, yielding an average accuracy of about 0.4, consistent with a single train-test split. The small variance of 0.0003 across folds signifies stable performance. To enhance model performance, we need a more effective approach, as outlined below.

Implement Models after Modifying One Hyperparameter:

In order to gain better performance of our model, we can test fit our model with other hyperparameters instead of using default values. We modified **C** to check whether we can receive better performance. We should be able to identify whether this hyperparameter has a direct impact on the bias-variance trade-off. We built 9 different models on different **C**s (0.001, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 100).



- We successfully identified trade-offs between MSE and Variance. MSE decreases from **Model 1** to **Model 2** while variance increases. Also, the trend remains the same when **C** is increasing.
- From this perspective, **Model 3（C=0.05）** is the best model.
- As we saw some improvement by modifying one hyperparameter, we can further improve our model by tuning which involving more hyperparameters.

Part 4: Model Tuning:

Since our binary classifications in OLR used **LogisticRegression** from **sklearn**, we can tune our model by modifying hyperparameters available in **LogisticRegression**. All available hyperparameters are **penalty, dual, tol, fit_intercept, intercept_scaling, class_weight, random_state, solver, max_iter, multi_class, verbose, warm_start, n_jobs, l1_ratio.**

In our case, we will modify C and Penalty to tune our model. However, we also notice that not all penalties are supported by the default solver = 'lbfgs'. Therefore, we will also modify solver which allows us to test more models. The reasons for choosing these two parameters are explained as follows:

- **C**: The parameter **C** plays a crucial role in balancing training data fitting and preventing overfitting. To enhance our model's predictive performance, it's essential to fine-tune **C** as it directly impacts the risk of overfitting (high-variance) or underfitting (high-bias).
- **Penalty**: We're presented with two penalty options for our model: **L1**, or Lasso Regularization, which identifies important features by assigning non-zero coefficients and automatically setting

unimportant features to zero, resulting in a simpler model that prevents overfitting; and **L2**, or Ridge Regularization, which includes all features without feature selection, considering every feature in the model.

Performance Measures Selection and Justification:

In the previous section, we employed **Accuracy** as our performance metric, a common choice. However, for imbalanced datasets, **Accuracy** can be misleading. Given our unbalanced label distribution with most observations having the label 0, we recognize the importance of a more suitable metric. In our case, the preferred metric is the **Macro-Averaged F1-Score** due to the multiple labels in our dataset. It calculates the **F1-score** for each class separately and takes the unweighted average of these class-specific **F1-scores**. This metric helps assess the quality of predictions across different labels. In our Grid Search for the best model, we prioritize **Macro-Averaged F1-Score**, while also considering the **Micro-Averaged F1 Score** (not engaging model selection) to gauge the impact of data imbalance.

- The **Macro-Averaged F1-Score** of the Best Tuned Model is 0.183 which is higher than the best model from the previous section (0.175). Although the gap is small, we can still **see some improvement** from model tuning. The parameters are **'C': 1, 'penalty': 'l2', 'solver': 'liblinear'**.

Compare Feature importance in our model with Section 2:

We used the absolute value of the normalized value of feature coefficients to reflect the feature importance.

- For each binary classifier, although there are some variations, we notice that the distribution of feature importance in every classifier is similar to the feature importance in section 2.
- In the majority of binary classifiers, **continents_North America** is the most determining in model predictions. Only in binary classifier 14, **Q30_encoded** has the same importance as **continents_North Amarica**.

Part 5: Testing & Discussion

	True Label	Predicted Label	Salary_bucket_0	Salary_bucket_1	Salary_bucket_2	Salary_bucket_3
0	3	0	0.304603	0.093777	0.115842	0.153796
1	7	10	0.071676	0.032995	0.040510	0.028796

We successfully constructed our ordinal logistic regression model by developing binary classifiers based on the baseline labels, incrementally expanding the baseline label set with each model. This approach allows us to effectively model ordinal data using binary classification. We also define the most determining feature allowing us to conclude that whether the surveyor based in North America influences the surveyor's the most.

We ensured that our model's accuracy remains consistent across different training sets through cross-validation. To enhance performance, we fine-tuned key parameters via Grid Search and adopted the F-1 score as our performance metric, given the imbalanced label set observed during exploratory data analysis.

Despite these efforts, the final model still exhibits underfitting from the observations from confusion matrixes and the distribution of the true label and their prediction on both training and test sets, likely due to an imbalanced dataset, an unsuitable feature selection, and potential algorithm selection issues. This experience underscores the importance of following well-defined procedures, both before (data cleaning, EDA and feature selection) and after (evaluation, tuning, reflection) implementing machine learning algorithms to classification tasks.