Assignment 2

Question 1:

Before starting data cleaning, I examined the uploaded Excel file to identify the number of rows and columns. From this initial review, I determined that the first column (Duration in seconds) and the fourth column (Q5) needed to be dropped.

Next, I identified all single-column responses and checked for any missing values. Confirming that the output matched the single-column responses, I reviewed each one with missing data. Since this dataset originated from a survey, I opted to retain the single-column responses and replaced any missing values with the mode, as it represents the most common value, making it a logical and computationally efficient choice. Additionally, I corrected an issue in column Q26, where some values were incorrectly interpreted as dates.

For encoding, I applied ordinal encoding to columns with hierarchical responses and used one-hot encoding for the remaining columns. In the case of multi-column responses, where some surveyors did not select certain options, I replaced missing values with 0 and selected responses with 1. Keeping these data points is crucial, given that survey participants may have left certain questions blank for various reasons. This retained data will be useful in identifying the best features for analysis in Question 2.

Question 2:

Following the provided template, I dropped the target variable, split the data, and then performed feature engineering on columns that were logically meaningful. I initially created a new feature by calculating the ratio between Q11 (Years of coding/programming experience) and Q2 (Age). This ratio provides insight into a person's coding experience relative to their age, offering a normalized perspective of experience level. Another new feature I generated was the ratio between Q30 (Spending on ML/cloud services over 5 years) and Q26 (Data science team size). This feature gives insight into the spending per team member on ML and cloud services, indicating how much each data science team member benefits from the organization's budget for these resources. Both features add predictive value by providing deeper context and capturing relationships between variables that may not be immediately obvious in the raw data.

For feature selection, I performed logistic regression with grid search to identify the most relevant features in my dataset. Features were ranked by their importance, represented by the absolute values of the model's coefficients, which indicate each feature's contribution to the model's predictions. I used an importance cutoff of 0.1, which helped retain a manageable number of meaningful features while preserving the model's performance and interpretability. Figure 1 and Table 1 display the list of selected features along with their importance scores.

Question 3:

When performing 10-fold cross-validation, the accuracy across all 10 folds was consistent. As shown in Table 2, the highest accuracy achieved was 88.1%, while the lowest was 84.4%. The average accuracy across the folds was 86.1% (Table 3), indicating stable model performance.
One hyperparameter that directly impacts the bias-variance trade-off is C (regularization strength). I ran a loop with different C values to identify the optimal value for my dataset. Table 4 shows the C values

tested and their associated average bias and variance. Based on this analysis, I found that C=1.0 works best for my dataset.

Figure 2 illustrates the Bias-Variance Trade-Off for Ordinal Regression. Bias (in blue) decreases as C increases, which is expected because a higher C value corresponds to weaker regularization, allowing the model to fit the training data more closely and thus reducing bias. Conversely, variance (in orange) increases with C, as lower regularization (higher C) makes the model more flexible and prone to overfitting, leading to higher variance. The red dashed line represents the recommended C value, where the balance between bias and variance is optimal for this model. This point minimizes generalization error, achieving a balance between model flexibility and predictive stability.

Yes, scaling/normalization is generally necessary for a logistic regression task, especially when using regularization and when features vary significantly in scale. The first reason is that, in our ordinal logistic regression model, we apply regularization (with a regularization parameter C), and the scale of features can impact the regularization effect. Regularization penalizes large coefficients, and without scaling, features with larger values could dominate the regularization term, causing the model to underutilize smaller-scale features. The second reason is that scaling ensures logistic regression converges faster and more reliably when features are on a similar scale. Finally, since we are analyzing the bias-variance trade-off, maintaining a consistent scale across features allows the model to treat all features equally, enabling a fair assessment of each feature's contribution to performance without being disproportionately influenced by scale differences.

Question 4:

Accuracy may not be a suitable performance metric for this problem because the dataset is imbalanced, meaning one class significantly outnumbers the others. In such cases, accuracy can be misleading, as a model that always predicts the majority class can achieve high accuracy without learning meaningful patterns related to the minority class. Another limitation of accuracy is its lack of sensitivity to class imbalances and types of errors. Accuracy gives equal weight to all predictions, making it inadequate when correctly identifying the minority class is critical. Metrics like precision, recall, and F1-score are better suited for evaluating model performance in imbalanced contexts, as they separately account for true positives, false positives, and false negatives. Furthermore, relying solely on accuracy can lead to a model biased toward the majority class, achieving a high score by ignoring or misclassifying minority cases. This often results in a model that lacks generalizability and performs poorly on rare or minority events.

The hyperparameters for the ordinal logistic regression model include C, penalty, solver, max_iter, tol, and multi_class. I selected C and penalty for tuning. The combination of these two hyperparameters allows me to explore different regularization types and strengths, directly impacting model performance. Together, C and penalty provide the flexibility to control model complexity, select important features, and prevent overfitting. This tuning is particularly beneficial in logistic regression, where regularization plays a crucial role in balancing the fit to the training data with generalizability to new data.

When comparing Figure 3 with Figure 1, we can observe that some features have been reordered in terms of importance. Certain features have increased in importance, while others have become less significant.

Question 5:

Looking at Table 6, the high training accuracy suggests that the model has learned the patterns in the training data effectively, without underfitting. The training F1-score of 86.62%, which is close to the training accuracy, indicates balanced performance in terms of precision and recall on the training data. The similarity between the training and test accuracies suggests that the model generalizes well and is not overfitting. Additionally, the test F1-score of 86.35% reinforces that the model maintains precision and recall when moving from training to test data, further indicating good generalization.

The model shows no signs of overfitting or underfitting, as indicated by the close training and test scores discussed above. To further improve the performance on both the training and test sets, I would consider expanding the range of options for regularization strength and penalty. Additionally, adjusting max_iter and experimenting with different solvers may lead to enhanced performance. Another option is to create new features by combining existing ones that appear meaningful together. Lastly, I would consider dropping columns with a high percentage of missing values, which could reduce noise in the dataset. Figure 4 presents a comparison between the true and predicted class distributions for both the training and test sets in an ordinal logistic regression model. The left plot shows the distribution of true versus predicted values within the training set, where blue bars represent the true values and orange bars represent the predicted values for each class. In the training set, the model's predictions closely align with the actual class distribution, indicating a good fit to the training data. The right plot provides a similar comparison for the test set. Here, the alignment between true and predicted distributions remains strong, with predicted values nearly matching the true values for each class. This suggests that the model generalizes well to unseen data, maintaining consistent performance without significant loss in accuracy or class balance. The overall similarity between distributions in both training and test sets implies that the model is neither overfitting nor underfitting, achieving a balance between fitting the data and generalizing beyond it. This balanced performance across classes highlights the model's reliability and suitability for this classification task. Through analyzing the dataset and training a classification model, I gained several key insights. The model highlighted specific features—such as cloud platform usage, professional roles, and geographic indicators—as strong predictors of the target variable. These influential attributes emphasize the importance of experience, resources, and location in determining outcomes within this dataset. Although the dataset showed some class imbalance, the model handled it well, achieving consistent performance across both training and test sets. This consistency, reflected in the close alignment of predicted and true distributions across classes, indicates that the model generalizes effectively without overfitting.

Additionally, logistic regression offered interpretability, allowing straightforward identification of important features. By tuning regularization strength (C) and selecting relevant features, I achieved a balanced bias-variance trade-off, resulting in a model that is both accurate and simple. Overall, these insights suggest that a well-regularized logistic regression model is effective for this classification task, providing reliable predictions and valuable interpretability for stakeholders.

Appendix

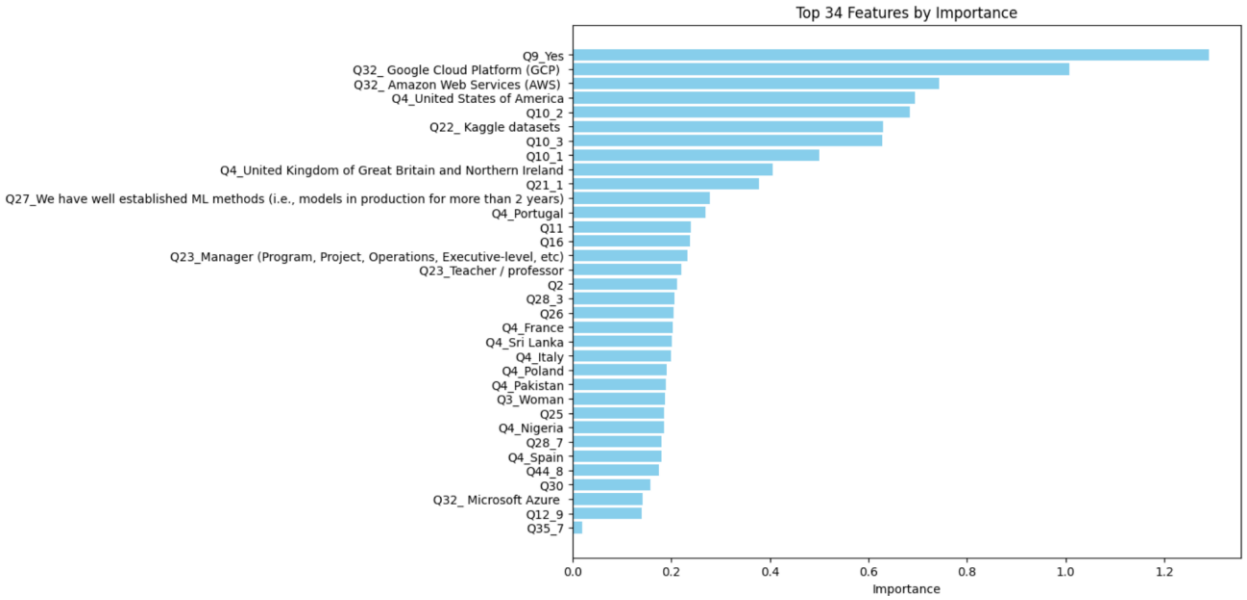

Figure 1: Top 34 Feature by Importance

Table 1: Selected Features by Importance

Feature	Importance
Q9_Yes	0.968236
Q32_ Google Cloud Platform (GCP)	0.919008
Q32_ Amazon Web Services (AWS)	0.83764
Q22_ Kaggle datasets	0.609141
Q4_United States of America	0.572759
Q10_2	0.443041
Q10_3	0.373672
Q10_1	0.337151
Q11	0.274164
Q4_United Kingdom of Great Britain and Northern Ireland	0.256241
Q27_We have well established ML methods (i.e.,...)	0.241875
Q30	0.237432
Q4_Pakistan	0.214326
Q4_Sri Lanka	0.197861
Q4_Nigeria	0.19348
Q23_Manager (Program, Project, Operations, Executive)	0.191263
Q16	0.190995
Q4_Portugal	0.181779
Q2	0.173172

Q4_France	0.154269
Q28_7	0.150602
Q28_3	0.147195
Q25	0.146599
Q4_Italy	0.144706
Q26	0.139247
Q4_Spain	0.137021
Q21_1	0.13635
Q12_9	0.133679
Q32_ Microsoft Azure	0.132628
Q23_Teacher / professor	0.130998
Q44_8	0.130086
Q4_Poland	0.129496
Q3_Woman	0.12674
Q35_7	0.126191

Table 2: K-Fold Accuracy

Fold	**Accuracy**
1	0.8526
2	0.8439
3	0.8719
4	0.8491
5	0.8614
6	0.8752
7	0.8699
8	0.8506
9	0.8524
10	0.8805

Table 3: Average Accuracy and Variance of K-Fold

Average Accuracy	0.860759411710295
Variance of Accuracy	0.000146840781381918

Table 4: C values and their Average Bias and Variance

C	Avg Bias	Avg Variance
0.001	0.1537	0.0066
0.01	0.1380	0.0088
0.1	0.1302	0.0089
1	0.1286	0.0094
3	0.1286	0.0093
5	0.1284	0.0094
10	0.1284	0.0094
100	0.1283	0.0094

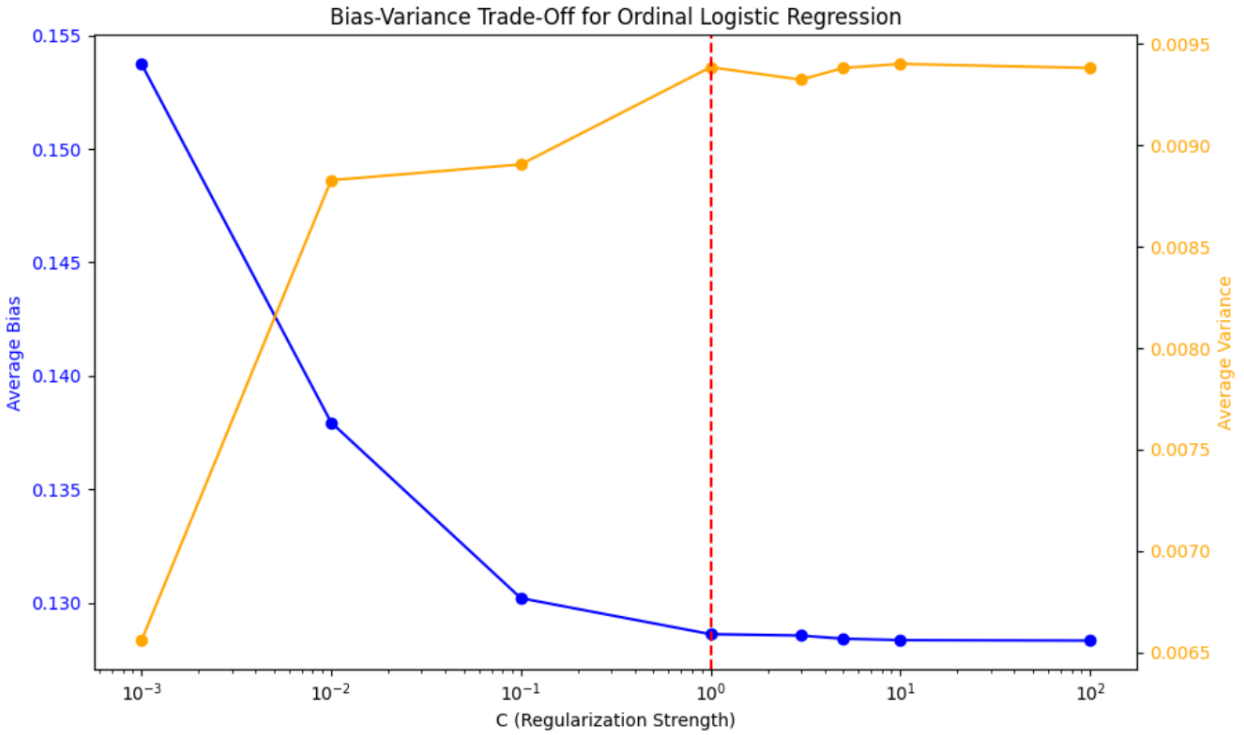

Figure 2: Bias-Variance Trade-Off for Ordinal Logistic Regression

Table 5: Best Hyperparameters

Best Parameter	C	penalty
Best Parameter Values	5	l2
Best Score	0.8624568997848922	

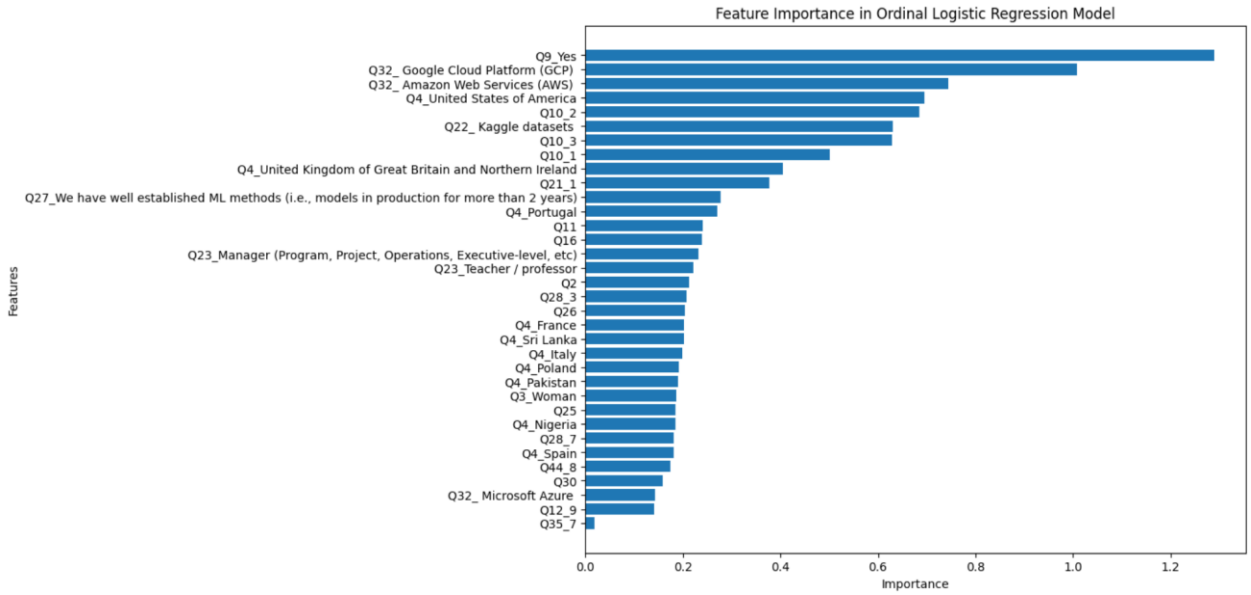

Figure 3: Feature Importance in Ordinal Logistic Regression Model

Table 6: Training & Testing Accuracy & F1-Score using best model

Training Accuracy:	0.8646180860403863
Training F1-Score	0.866240766009059
Test Accuracy	0.8619418271200328
Test F1-Score	0.8635406630580077

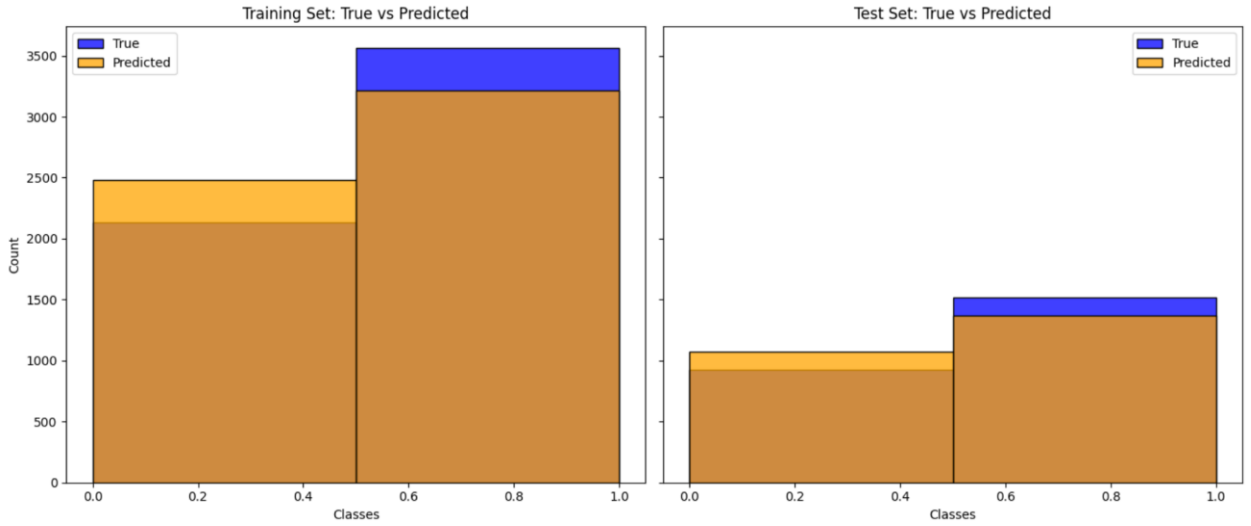

Figure 4: True vs Predicted for Training and Test Sets