

Omar Alhakeem

Cleaning Dataset

STEPS TAKEN UP TO DATA VALIDATION

- Loaded Data: imported the dataset and performed initial checks on its data types and basic information. Initial Cleaning: I transformed all question mark ("?") characters into null values to correctly represent the missing data.
- Data Quality Checks: checked for missing values, duplicate rows, and performed an initial outlier check using the IQR method.
- Statistical Analysis: calculated basic summary statistics (mean, median, mode, etc.) for my numerical variables and complementary statistics (skewness, kurtosis) to understand my data's shape. I also generated a separate summary for all categorical variables.
- Variable Removal: evaluated and removed four specific columns using the .drop() method: fnlwgt, education (the string version, as I kept the integer version for statistical analysis), capital.gain, and capital.loss.
- Data Visualization: used three types of plots to inspect my data: Histograms to see the distribution of my numerical variables. Box plots to visually identify outliers. Bar plots/Pie charts to check the frequencies of my categorical variables.
- Correlation Analysis: created a correlation matrix and scatter plots to understand the relationships between my numerical variables.
- Validation Planning: planned to perform a "before and after" comparison of the correlation matrices to validate my entire cleaning process.
- The *Race* variable has not been encoded as the education percentage is directly correlated to education

JUSTIFICATION OF REMOVED VARIABLES

- The variable *fnglwgt* is undefined and no documentation was found from the source to explain what this variable does. It is assumed that it is a calculated value from the previous data handler.
- The *education* variable appears to be duplicated in the dataset, but one is a string and the other is integer. We kept the integer version since we can do our statistical analysis with it instead of having to convert the string values.

- The *capital.gain* and *capital.loss* variables both have the value of 0 approx 96% and 98%, respectively, across all rows.

Janice Underwood

Training Multiple Machine Learning Models

To compare different types of algorithms, we trained several models:

- **Logistic Regression**

A simple, interpretable model that acts as a good baseline. A simple linear model often used for binary classification. It tries to find the best line (or surface) that separates the two classes. It works well for linearly separable data.

- Pros: Simple and easy to understand; fast to train and easy to interpret.
- Cons: Struggles with non-linear datasets; it can overfit on high-dimensional data, ex: having many columns; it needs too much pre-processing.

Code Result: Training model: Logistic Regression Accuracy: 0.8206, Precision: 0.6834, Recall: 0.5251, F1Score: 0.5939, ROC-AUC: 0.8713

- **Decision Tree Classifier**

A non-linear model that splits data into branches based on feature conditions. Captures non-linear decision patterns and provides clear decision rules.

- Pros: Easy to interpret ; requires less data processing compared to other algorithms; can handle both numerical and categorical data, as well as missing values.
- Cons: Can overfit if not tuned; single trees are not robust; sensitive to small changes

Code Result: Training model: Decision Tree Accuracy: 0.7943, Precision: 0.6046, Recall: 0.5107, F1 Score: 0.5537, ROC-AUC: 0.7747

- **Random Forest Classifier**

It reduces overfitting and usually performs better than a single tree. An ensemble of many decision trees, offering better accuracy and stability.

- Pros: Resistant to overfitting ; robust to noise, outliers , missing values; helps identify which variables are most influential in the prediction.
- Cons: Slower to train than logistic regression, requires more memory; training the model can be time-consuming especially with large datasets

Code Result: Training model: Random Forest Accuracy: 0.8060, Precision: 0.6263, Recall: 0.5538, F1 Score: 0.5878, ROC-AUC: 0.8471

Kim Nguyen

K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and XGBoost are three popular machine learning models, each with unique strengths and weaknesses. KNN is a simple, instance-based method that makes predictions by comparing new data to its closest neighbors, but it can be slow and sensitive to irrelevant features. SVC focuses on finding the best separating boundary with maximum margin and performs well in high-dimensional spaces, though it can be slow to train and requires careful tuning of parameters. XGBoost is a powerful ensemble boosting algorithm known for its high accuracy and speed, especially on structured data, but it can be complex to tune and may overfit if not properly regularized. Together, these models represent a range from simple to highly sophisticated techniques used in modern machine learning.

XGBoost

It builds tree one at a time, where each new tree reduces errors of previous trees. It also helps to prevent overfitting better than standard Gradient Boosting Machines. And it works well with non-linear data.

Accuracy: 0.8219

Precision: 0.6709

Recall: 0.5637

F1 Score: 0.6126

ROC–AUC: 0.8756

K-Nearest Neighbors Classifier

This model works well with small and low-dimensional dataset. It is very simple and easy to understand.

Accuracy: 0.8024

Precision: 0.6204

Recall: 0.5386

F1 Score: 0.5766

ROC–AUC: 0.8222

Support Vector Classifier (SVC)

This is effective for clean, well-separated datasets. It is suitable for datasets with many features. It makes decisions based on a small subset of important data points.

Accuracy: 0.8215

Precision: 0.7245

Recall: 0.4605

F1 Score: 0.5631

ROC–AUC: 0.8735

Maria Jose Viveros Riquelme

Maria Jose Viveros contributed by conducting the comparative analysis of model validation and the evaluation of ensemble strategies. Instead of training the initial models, she focused on transforming raw metrics into analytical insights that supported final decision-making. This role allowed the project to progress from isolated numerical outputs toward a structured and evidence-based understanding of performance.

She began by reviewing the validation results of all classifiers trained by Janice and Kim, including Logistic Regression, Decision Tree, Random Forest, XGBoost, KNN, and SVC. Her analysis relied on a multi-metric framework integrating accuracy, precision, recall, F1-score, and ROC-AUC. This approach ensured a more comprehensive evaluation, capturing not only overall correctness but also each model's ability to detect positive cases and manage classification errors.

During the evaluation process, she identified that the Decision Tree Classifier was the worst-performing model in terms of balanced performance, particularly due to its lower recall and ROC-AUC. This finding illustrated how algorithms that are highly interpretable may still struggle when dealing with complex or noisy data. Highlighting these limitations helped contrast the stronger and more stable behavior of models such as XGBoost, Random Forest, and Logistic Regression.

A central part of her contribution involved emphasizing that accuracy alone was not a sufficient indicator of model quality. She highlighted the importance of observing how precision and recall interact, especially in contexts where class identification carries significant weight. She also interpreted ROC curves as visual evidence of how models behave across varying thresholds, connecting theoretical evaluation concepts with practical implications.

Another key responsibility was selecting the Top 3 models for the ensemble phase. This decision shaped the entire modeling structure because ensemble performance depends heavily on the strength and diversity of its base learners. By grounding the selection in validation data, she ensured that the ensemble design was guided by objective evidence rather than assumptions.

After identifying the strongest models, she evaluated the ensemble strategies, including the Average Ensemble, Bayesian Ensemble, and Stacking Ensemble. Through comparative analysis, she explained why the Bayesian Ensemble, although theoretically robust, did not outperform the strongest individual models. In contrast, the Stacking Ensemble achieved the highest overall performance, with an accuracy of 0.8271, a precision of 0.6991, and a

ROC-AUC of 0.8797. She clarified that this advantage came from its meta-learning architecture, which integrates predictions in a more dynamic and adaptive manner.

Finally, her contribution involved organizing and communicating the results in a clear, structured narrative that connected classroom concepts, literature insights, and empirical findings. By integrating technical evaluation with conceptual reasoning, she ensured that the final decision to select the Stacking Ensemble was both analytically justified and pedagogically meaningful.