Income Classification System Report

This project is an income classification system that classifies individuals into one of two categories: those earning more than $50K annually and those earning $50K or less annually, based on various demographic and financial attributes. I used the "Adult" dataset from the UCI Machine Learning Repository, which is frequently used to benchmark classification tasks. This served as the source of the data for this research. The UCI Machine Learning Repository dataset was selected because it offers a wide range of characteristics that are very important for forecasting income levels, such as demographic traits (educational attainment, marital status, work hours per week, age, gender, and native country). Its widespread use in research also validates its thoroughly documented preprocessing procedures and makes comparisons with pre-existing models easier. This project's iterative process demonstrates how a machine learning pipeline develops from a basic implementation to a final draft that has been fine-tuned. Each draft builds upon the previous one, incorporating new techniques and optimizations to improve accuracy and efficiency. This report documents the development process, explaining the purpose of each draft and the methods used in every iteration along with the challenges encountered and how they were addressed.

Implementing strong preprocessing procedures, choosing and fine-tuning a logistic regression model, and attaining accuracy gains through iterative refinement are the main goals of this classification method. Three versions are summarized in this report: draft 1, draft 2, and the final draft.

Draft1:

The project's first draft was created to establish the framework for the machine learning pipeline. The goal of this version was to create a simple categorization system that could make predictions with little modeling or preprocessing. Setting up the dataset, putting a simple logistic regression model into practice, and making sure the pipeline could process the data from beginning to end were the top priorities.

The CSV library was used to load the data from a file for this project. This library reduced the possibility of problems resulting from incorrectly structured entries by making sure the records were correctly parsed into a format appropriate for additional processing.

The UCI dataset, which included numerical and categorical variables, is well known for classification tasks. Little preprocessing was done to simplify the process and allowed me to focus more on creating an operational classification model. I utilized an ordinal encoding technique that was used to represent categorical characteristics, such as employment type ("Private," "State-gov"). This approach made it possible for the model to efficiently process these non-numeric data points by allocating an integer value to each distinct string based category according to their order of presentation. However, any possible connections between categorical values—for example, how certain categories might be more closely related or have more comparable meanings than others—were not captured by the encoding technique. For

instance, "State-gov" and "Federal-gov" might be more similar than either is to "Private," but these underlying relationships were ignored by the simple encoding, which gave them unrelated numerical values. The logistic regression model in this iteration employed numerical features like age and years of schooling exactly as they were, without any normalization or scaling.

A logistic regression approach was used with default settings for the machine learning model. This method avoids complicated parameter optimization in favor of concentrating on the mechanics of building and assessing a simple classification system. A single training and testing data split, with 80% of the dataset set aside for training and the remaining 20% for testing, was used to evaluate the model's accuracy. This simple assessment offered a preliminary comprehension of the model's performance, giving opportunities for enhancement in later rounds. This methodology, however, could have been more precise as it did not incorporate techniques like cross-validation to alleviate the risks of overfitting or underfitting. By using simple preprocessing and a logistic regression model, this draft created a foundational framework for the project, providing a strong basis for future iterations that would polish and improve it.

Biggest Challenges:

Managing categorical characteristics in this draft was a major difficulty. Ordinal encoding was employed, but it may have misrepresented the data by unintentionally labeling data into categories without a natural hierarchy. Also, the lack of feature scaling led to significant differences in the ranges of numerical features, which impeded the model's ability to train efficiently and perform at its best.

Performance:

Draft 1's accuracy of roughly 72% was quite impressive, demonstrating its fundamental methodology. Although the model's capacity to identify intricate patterns was constrained by the use of simple ordinal encoding for categorical data and the absence of feature scaling, it served as a helpful foundation for subsequent iterations.

Draft 2:

Many of the problems found in the first draft were fixed in the second draft, which added major enhancements to the preprocessing and modeling pipeline.This iteration included feature scaling, hyperparameter tuning, and enhancing the encoding of categorical features to better represent their hierarchical relationships.

One of the most impactful changes was the use of min-max normalization for scaling numerical features. By standardizing their values to fall between 0 and 1, this preprocessing step ensured that features with larger magnitudes, such as age or income, did not disproportionately influence the model’s training process. This allowed the model to analyze inputs more effectively and converge on meaningful patterns. The introduction of scaling was a major improvement over the first draft, where raw numeric features were left unscaled, leading to increased biases in the training process.

Categorical features were also encoded using a more flexible approach, which avoided the rigid assumptions of ordinal encoding. This change helped the model better understand non-ordinal data, such as job categories and education levels, without imposing artificial hierarchies that could misrepresent the relationships between these categories. By capturing meaningful relationships in the data, this adjustment improved the model's ability to generalize.

In addition, hyperparameter tuning was introduced to optimize model performance. Specifically, the training pipeline explored different values for the regularization parameter α, testing options such as 0.1, 1.0, and 10.0. This iterative process allowed the model to identify the value that maximized training accuracy, and the best-performing configuration was kept for evaluation. Hyperparameter tuning represented a significant shift from the default parameters used in Draft 1, demonstrating a more systematic approach to model optimization.

The combination of better preprocessing, feature scaling, and hyperparameter optimization led to measurable gains in accuracy compared to Draft 1. This iteration laid a solid foundation for further refinements, demonstrating the importance of thoughtful data preparation and model parameterization.

Big Challenges:

Despite the progress made, Draft 2 revealed the need for an improved method for encoding categorical features. While the enhanced encoding approach increased flexibility, it still introduced some implicit assumptions about relationships between categories, which could mislead the model in certain scenarios. Finding a method that better preserves the semantic independence of categories remains a challenge to be addressed in the final iteration. Additionally, hyperparameter tuning significantly improved model performance, but it introduced more computational overhead. The process of testing multiple configurations also required more time and resources.

Performance:

Draft 2 demonstrated the effects of increased preprocessing and hyperparameter tuning by increasing accuracy to about 74%. A significant improvement over the first draft was made to the model's generalization through the use of min-max scaling for numerical features and a more adaptable encoding technique for categorical variables.

Final Draft:

The final draft, which combined sophisticated preprocessing methods, a refined logistic regression model, and a strong evaluation framework, was the result of previous iterations. By using one-hot encoding for categorical variables, enhancing feature scaling to better manage outliers, and optimizing hyperparameters, this version addressed the drawbacks noted in previous iterations. Cross-validation was also implemented in the evaluation technique, offering

a more trustworthy indicator of the model's accuracy. These improvements were made to optimize the system's prediction capabilities and guarantee reliable, consistent outcomes across a range of datasets.

The final draft introduced comprehensive preprocessing techniques to ensure the model could leverage the full potential of the dataset. One of the standout features was the implementation of one-hot encoding for categorical variables. One-hot encoding converted each category into a binary vector, eliminating any implicit relationships between categories. For example, categorical features like work class or marital status were transformed into independent binary columns, allowing the model to treat them without assuming any hierarchical order. Additionally, min-max normalization was applied consistently to numeric features, such as age and hours worked per week. This ensured that all numerical values fell within the range of 0 to 1, preventing features with larger magnitudes from heavily influencing the model during training. By standardizing both categorical and numeric data, this draft addressed critical limitations in the earlier iterations and provided a cleaner dataset for the model.

In terms of model optimization, this draft adopted the LBFGS solver, a quasi-Newton optimization method known for its efficiency with large datasets. This solver improved the convergence of the logistic regression model during training, resulting in better parameter tuning and faster computation. Hyperparameter tuning was further refined by expanding the range of regularization strengths tested. This iterative process explored a broader set of values to identify the configuration that delivered the highest accuracy without overfitting the data. The evaluation strategy also became more rigorous, as the model's performance was assessed on a completely separate testing set. This method ensured that the reported accuracy reflected the model's ability to generalize to unseen data as opposed to its memorization of the training set. These combined improvements in preprocessing, optimization, and evaluation marked a significant leap forward, delivering a more precise and reliable income classification system.

Biggest Challenges:

One-hot encoding resulted in an increased feature set, which added significant complexity to the final document. The dataset's dimensionality was significantly enhanced by converting category variables into binary vectors, which presented computational difficulties in the training and evaluation stages. Also, in addition to demanding greater memory, this increase in dimensionality prolonged computing times. This broader feature space heightened the risk of the model learning noise from the data, making it crucial to find a healthy balance between model complexity and the risk of overfitting.

Another significant challenge in the final draft was maintaining the interpretability of the model as it became more complex due to one-hot encoding and feature scaling. These preprocessing steps did enhance the model's ability to accurately classify data, but they also made it harder to understand how specific input features contributed to the predictions. In income classification, it is important to know the impact of features like education level or occupation on the model's decisions. However, the transformations applied during

preprocessing, such as converting categorical data into binary vectors, made these relationships less obvious.

Performance:

The final draft achieved an accuracy of approximately 77%, representing a significant improvement over earlier versions. This was attributed to the combination of robust preprocessing, advanced feature encoding, and systematic hyperparameter tuning.

Conclusion:

This project demonstrated the importance of a methodical approach in machine learning workflows, leveraging iterative refinement to enhance the system's overall accuracy and performance. Starting with a baseline model, each iteration included more complex methods to improve the system's accuracy, dependability, and resilience. The progression demonstrated how careful adjustments can greatly improve performance, starting with the initial draft's simple ordinal encoding and unscaled numeric features and ending with the final draft's complex preprocessing techniques like one-hot encoding, feature scaling, and hyperparameter optimization. The balance between simplicity and complexity, as well as the trade-offs associated with model creation, was made clear in the iterative process. Expanded feature sets and one-hot encoding, for example, boosted classification accuracy but decreased model interpretability and computational costs. Ultimately, the project achieved an impressive accuracy of 77%, offering a reliable system for income classification based on demographic and occupational data.