1 Introduction

This study is based on the UCI Adult dataset and aims to predict whether an individual's annual income exceeds \$50K using the K-Nearest Neighbors (KNN) algorithm. The dataset originates from the US Census and contains multiple socio-economic attributes, including age, education level, working hours, marital status, and more. Due to the significant imbalance in income distribution, the challenge of this classification task is to improve the model's ability to recognize high-income groups while maintaining stable overall classification accuracy.

2 Data Preprocessing and Feature Engineering

In the data preprocessing phase, all categorical variables were converted into numerical encodings to facilitate KNN calculations. The target variable `income` was extracted separately as the classification target. Missing values in numerical features were handled using mean imputation to ensure the completeness of the training data. Additionally, one-hot encoding was applied to categorical features to allow the classifier to fully utilize all categorical information.

Regarding data preprocessing, the dataset includes a mix of continuous variables (*e.g., age, fnlwgt, hours-per-week*) and categorical variables (*e.g., workclass, education, occupation*). Early exploratory analysis revealed several data-related issues: many attributes contained unknown values denoted by `"?"`. The test data's income labels included trailing periods (*e.g., "¡=50K."*), that needed correction for consistent factor levels, and the high dimensionality of the categorical variables required converting them into a numerical format. To address these challenges, I replaced `"?"` with NA and removed rows with missing values in the raw test data, reducing it to the expected 15,060 complete observations and correcting the trailing periods in the income labels so that the outcome variable could be uniformly encoded as a factor with levels "$\leq$50K" and ">50K".

Following these cleaning steps, I applied one-hot encoding using R's `dummyVars` function to transform the categorical variables into binary indicators, which suits the supervised learning referred to as *Missing Data Imputation for Supervised Learning*. This transformation not only provided a clear numeric representation for each category but also ensured that both training and test sets were handled consistently. In my approach, I carefully separated the predictors from the outcome during encoding to ensure that any missing values in the predictors were managed without compromising the integrity of the response variable.

3 Method 1

3.1 K Optimization and Model Evaluation

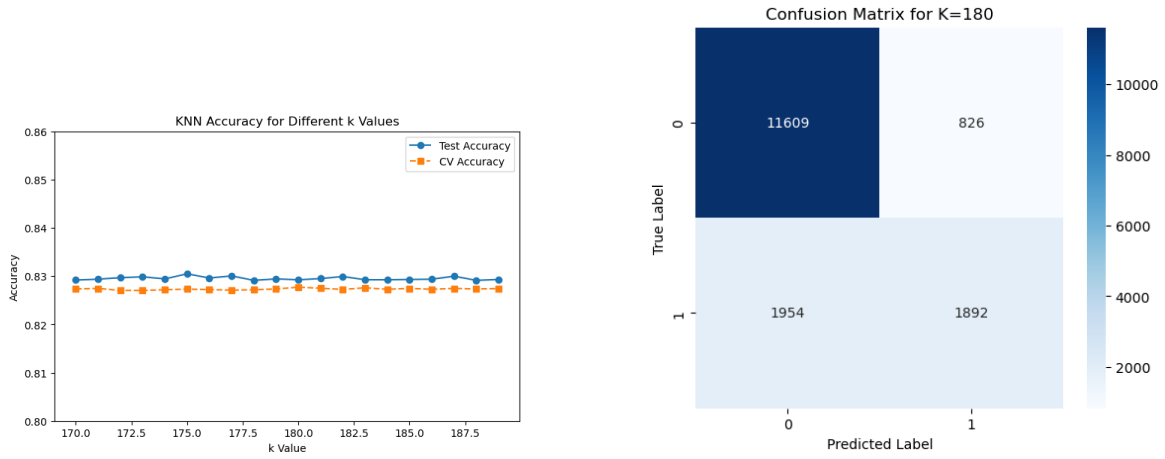
To optimize the performance of the KNN classifier, values of K were varied between 170 and 190 to evaluate their impact on classification accuracy. The model's evaluation metrics included test accuracy, cross-validation accuracy, confusion matrix, and ROC curve. Test accuracy measures the model's performance on new data, cross-validation accuracy helps mitigate overfitting risks, and the confusion matrix provides insights into classification errors across different categories.

3.2 K Selection and Classification Performance

Experimental results indicate that the optimal K value is 180, achieving a cross-validation accuracy of 0.8277, while the highest test accuracy approaches 0.83. As observed in the accuracy curve, the test accuracy is also relatively high at K=182 and K=186, but overall fluctuations are minor, suggesting that K has a relatively stable effect on model performance. Additionally, cross-validation accuracy remains relatively stable across the entire K range but is slightly lower than the test accuracy, which may indicate that the model relies on specific data distributions during training.

3.3 Confusion Matrix Analysis

For K=180, the confusion matrix is as follows:



(a) (b)

Figure 1:

The model performs well in predicting low-income individuals, correctly identifying 11609 instances while misclassifying 826 as high-income. However, 1954 high-income individuals were misclassified as low-income, reflecting a relatively high misclassification rate in this category. This indicates that the model struggles to differentiate high-income individuals, likely due to class imbalance, where the number of low-income samples significantly exceeds high-income samples. As a result, KNN tends to classify more samples into the low-income category.

3.4 ROC-AUC Analysis

Across different K values, the AUC remains around 0.87, indicating that the model has a strong ability to differentiate between income classes. From the ROC curve, the true positive rate (TPR) is relatively high, while the false positive rate (FPR) is relatively low, suggesting that the model is capable of correctly classifying most income categories. However, since KNN relies heavily on sample distribution, the misclassification rate for high-income individuals remains high. Further optimization may require adjusting feature weighting or exploring alternative classification methods.

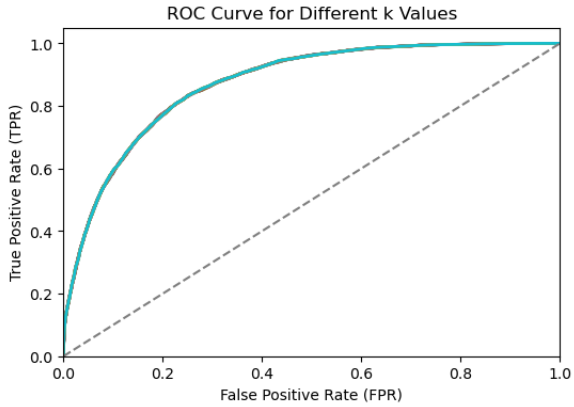


Figure 2: Test and Cross-validation Accuracy for Different K Values

3.5 Key Factors Affecting Income

Based on the experimental results and data analysis, the key factors influencing income levels include education level, occupation type, working hours, marital status, and gender. Individuals with higher education levels are more likely to earn higher incomes, technical and managerial jobs generally pay better than physical labor, and those who work longer hours per week tend to have higher earnings.

Marital status also plays a significant role, with married individuals—particularly men—earning higher wages on average. Additionally, gender disparities exist in income distribution, with women being underrepresented in high-income categories, reflecting potential socio-economic influences on earnings.

3.6 Misclassification Issues and Optimization Directions

Although KNN achieves relatively high accuracy on this dataset, misclassification remains a notable issue, particularly in the high-income group. Potential improvements include enhancing feature engineering by incorporating additional socio-economic variables such as geographic region and industry type. Addressing data distribution through oversampling (SMOTE) or undersampling techniques may help mitigate class imbalance effects. Additionally, given the computational complexity of KNN, alternative models such as logistic regression or random forests could be explored to improve classification efficiency while maintaining or improving accuracy.

4 Method 2

4.1 Pre-Processing

To start we have one-hot encoded data but for the support vector classification method this is not normalised data. A few elements of the data contained NA's so I just made extra columns to correspond to missing values

To do this I simply went in excel and divided each element in a column by the greatest value in that column. I did this rather than doing it between -1,1 because I wanted to have it more in line with the one-hot encoded 0's and 1's.

From here the data was still taking a long time to load under the svm method so I aimed to reduce the number of variables to take down on processing time. As stated earlier in the report we have a list of the importance of the different variables. With this fnlwgt is at the top as most important but obviously fnlwgt doesn't have any predictive power as it merely represents how prevalent a class is so I removed it.

From there I starting looking at correlations within the data. I used a correlation matrix to look at values that were highly correlated to remove them. We find that education can be boiled down to just years in education with no need to have the type of education received as they are highly correlated which makes sense as most people spend a similar amount of time to receive the same education. Due to the fact that 90% of the dataset is Americans we find that nationality is not a great predicter of how much money someone makes so this is removed as well. Since this categorical feature had a lot of categories this really speeds up computation time. Sex and marriage status are very highly correlated to the point where we can remove sex. I think it's because many marriages status' either pool incomes rendering sex (for straight people) irrelevant or are typically held by one gender e.g. Widows being typically women. These are the only ones removed bringing us down to 43 one-hot encoded variables.

Then I loaded my dataset into R set its headers so its readable and made a development set that is about 20% of the training set.

4.2 Processing

The method that I have chosen to try is support vector classification We first set a seed so everything is reproducible. The seed was chosen at random and for the sake of not allowing the possibility of me picking "good seeds" the svc was done entirely in this seed. Comparing to results from others my results look in line with average as well.

I tried out linear, polynomial, and radial kernels also tweaking cost to try work out the best form for

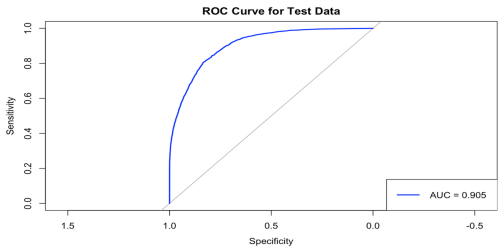
the svc to take.

4.3 Data analysis

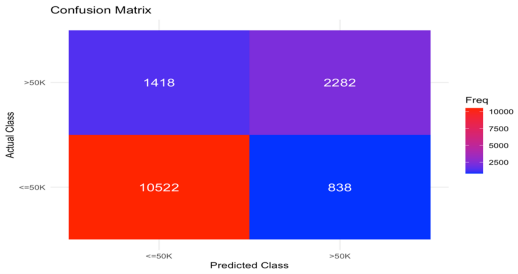
What I found from trying out linear, polynomial and radial kernels is that they all work with similar efficiency. Linear slightly wins out over the other two. This is I imagine due to the high dimensions of the data allowing many curves to separate the data similarly due to the distance between points.

Cost generally affects the distribution of points that are true positive to true negative. As cost increases, we generally get slightly more true positives and slightly less true negatives though this effect is negligible

We get a base of success with 85.1% of points being categorised correctly. There is 24.08% of 50k in the training dataset so we can compare this to our confusion matrix to try and get an idea of how well it is doing. 85% is significantly higher then the 76% we would get by randomly guessing so the method is working. In terms of Sensitivity, we reach a 73% which is well over the 24% from random guessing so we are correctly identifying people who make a lot of money quite effectively. Specificity we reach about 90%, NPV equals 93% . However, PPV only equals about 59% however, this is where we reach the limit of our method. This means that if while we know that if we want to find positives they are generally in the positive class. We find that given something is given the positive class it is still a toss up as to whether it is actually positive or not.



(a) Third picture.



(b) Fourth picture.

Figure 3: Two images side by side in the same line.

4.4 Preprocessing

From my side, I did not scale down `fnlwgt`, and did not exclude any categorical or numerical data subset as my first attempt. But I did generate the variable importance table in the end to reduce noise from data in further training and improvement reference.

One core of my methodology is grounded in the principles of bagging (*bootstrap aggregating*), which form the theoretical basis for random forests. In my implementation, I utilized the `ranger` package integrated within the `caret` framework. I set up a 5-fold cross-validation to optimize hyperparameters such as the number of predictors sampled at each split (`mtry`) and the splitting rule (evaluating "gini" versus "extratrees"). This rigorous cross-validation minimized overfitting and ensured that my final model—using `mtry = 49`, splitrule = "gini", and `min.node.size = 1`—was tuned for robust performance across varied data distributions.

To evaluate my final model, I conducted a comprehensive assessment on the cleaned, one-hot encoded test set. I first examined the model's performance via a confusion matrix, which revealed an overall accuracy of 85.02% and a particularly high sensitivity of 92.62% for detecting individuals earning "≤50K", although the specificity for those earning "$> 50K$" was lower at 61.68%.

Additionally, I plotted the ROC curve, which shows the trade-off between sensitivity and (1 - specificity) across different thresholds. In this part, I retrained the model. The only difference is that in the first model (`model_ranger_forROC`), I explicitly set the argument `importance = "impurity"`, which instructs the ranger algorithm to compute variable importance based on the mean decrease in Gini impurity. The ROC analysis yielded an AUC of over 0.90, indicating that my model is

very effective at ranking individuals so that those earning "$> 50K$" tend to receive higher predicted probabilities than those earning "$\leq$50K".

Finally, I examined variable importance based on mean decrease in Gini impurity; the results highlighted key predictors such as marital status, age, and capital gain—findings that align with known socioeconomic determinants of income.

In summary, my workflow—spanning data cleaning and feature encoding, bagging-based random forest training with cross-validation, and model evaluation via confusion matrix, ROC curve, and variable importance plots—demonstrates a robust and interpretable approach to classifying income levels using the UCI Adult dataset.

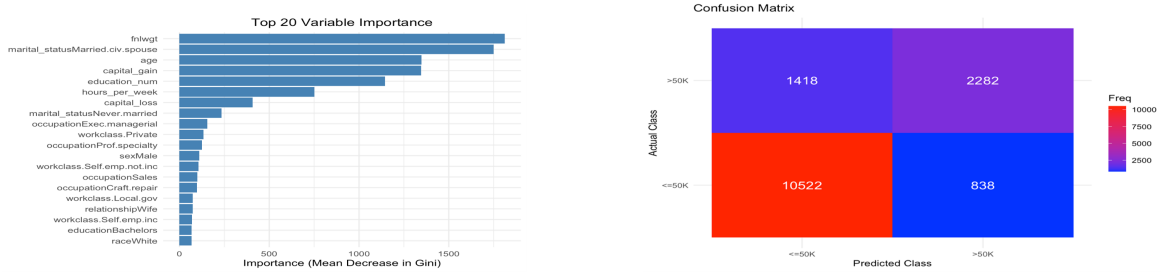
5 Method 3

5.1 Data Cleaning and Feature encoding

Regarding to data preprocessing, the dataset includes a mix of continuous variables (e.g., age, fnlwgt, hours-per-week) and categorical variables (e.g., workclass, education, occupation). Early exploratory analysis revealed several data-related issues: many attributes contained unknown values denoted by "?", the test data's income labels included trailing periods (e.g., "<=50K.") that needed correction for consistent factor levels, and the high dimensionality of the categorical variables required converting them into a numerical format. To address these challenges, I replaced "?" with NA and removed rows with missing values in the raw test data reducing it to the expected 15,060 complete observations and correcting the trailing periods in the income labels so that the outcome variable could be uniformly encoded as a factor with levels "<=50K" and ">50K." Following these cleaning steps, I applied one-hot encoding using R's dummyVars function to transform the categorical variables into binary indicators which is because it suits the supervised learning referred [Missing Data Imputation for Supervised Learning]. This transformation not only provided a clear numeric representation for each category but also ensured that both training and test sets were handled consistently. In my approach, I carefully separated the predictors from the outcome during encoding to ensure that any missing values in the predictors were managed without compromising the integrity of the response variable.

5.2 Bagging-based Model Training

From my side I did not scale down fnlwgt, and did not exclude any categorical or numerical of data subset as my first attempt. But I did generate the variable importance table n the end for reduce noise from data in further training and improvement reference. The core of my methodology is grounded in the principles of bagging (bootstrap aggregating), which form the theoretical basis for random forests. In my implementation, I utilized the fast ranger package integrated within the caret framework. I set up a 5-fold cross-validation scheme to optimize hyperparameters such as the number of predictors sampled at each split (mtry) and the splitting rule (evaluating "gini" versus "extratrees"). This rigorous cross-validation minimized overfitting and ensured that my final model—using mtry = 49, splitrule = "gini", and a minimum node size of 1—was tuned for robust performance across varied data splits.



(a) (b)

Figure 4:

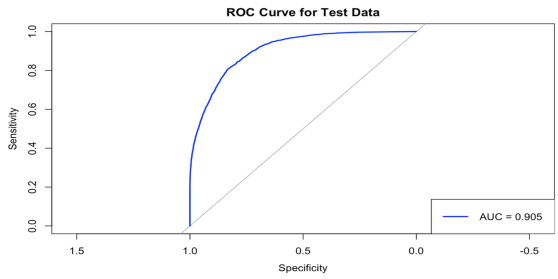


Figure 5: Test and Cross-validation Accuracy for Different K Values

5.3 Model Evaluation

To evaluate my final model, I conducted a comprehensive assessment on the cleaned, one-hot encoded test set. I first examined the model's performance via a confusion matrix, which revealed an overall accuracy of 85.02% and a particularly high sensitivity of 92.62% for detecting individuals earning

"<=50K," although the specificity for those earning ">50K" was lower at 61.68%. Additionally, I plotted the ROC curve, which shows the trade-off between sensitivity and (1 – specificity) across different thresholds. In this part, I retrained the model. The only difference is that in the first model ($model_ranger_forROC$), I explicitly set the argument importance = "impurity", which instructs the ranger algorithm to compute variable importance based on the mean decrease in Gini impurity. The ROC analysis yielded an AUC of over 0.90, indicating that my model is very effective at ranking individuals so that those earning ">50K" tend to receive higher predicted probabilities than those earning "<=50K." Finally, I examined variable importance based on mean decrease in Gini impurity; the results highlighted key predictors such as marital status, age, and capital gain—findings that align with known socioeconomic determinants of income. In summary, my workflow—spanning data cleaning and feature encoding, bagging-based random forest training with cross-validation, and model evaluation via confusion matrix, ROC curve, and variable importance plots—demonstrates a robust and interpretable approach to classifying income levels using the UCI Adult dataset.

6 Conclusion

In this study, three distinct classification methods were applied to the UCI Adult dataset: K-Nearest Neighbors (KNN), Support Vector Classification (SVC), and a bagging-based Random Forest implemented via the `ranger` package. Each method demonstrated its own strengths and limitations.

The KNN approach, after careful optimization of the parameter K, achieved test accuracies around 83% with cross-validation accuracy near 82.77%. However, its performance was sensitive to the choice of K and it exhibited notable misclassification of high-income individuals due to inherent class imbalance.

Similarly, the SVC method (with experiments across linear, polynomial, and radial kernels) delivered comparable performance, with linear kernels slightly outperforming the others. Although SVC provided a robust classification framework, the computational cost and sensitivity to the high-dimensional feature space posed challenges.

The Random Forest model, on the other hand, exhibited a strong overall performance with an accuracy of 85.02% on the test set. Notably, it achieved a high sensitivity of 92.62% for detecting individuals earning $\leq$ 50K while maintaining a respectable ROC-AUC of over 0.90. Furthermore, the variable importance analysis identified key predictors such as marital status, age, and capital gain, which align well with established socioeconomic factors influencing income.

In summary, while each method is viable, the Random Forest approach offers the best balance between classification accuracy, robustness to overfitting (as ensured by rigorous cross-validation), and interpretability. These findings suggest that, for the purpose of income classification using the UCI Adult dataset, the bagging-based Random Forest is the most promising candidate for further development and application.