Introduction to Machine Learning Group Assignment

Sahaya Mercy Victor Babu, Yawen Dang, Vivian Ballestro, Rohith Rajeev
Group G55

1 Introduction

This paper intends to predict the income of individuals earning above and below $50k annually across a wide number of countries based on personal attributes. The motivation behind this problem references the use of a United States Census data which added credibility, coupled with the fact it's a relevant topic in modern day times. Additionally, the dataset is highly referenced, thus affirming its usability in Machine Learning models.

Since this is a classification problem, the team chose Decision Tree, Support Vector Machines, Neural Networks, and Logistic Regression models to predict income. To widen the scope of the research, although Logistic Regression is not ideal for classification problems, this has been investigated in the report.

Since the Support Vector Machines allow for non-linear decision boundaries, this model was expected to perform the best which formulated our first research question. While there were many features, not all features are suitable for all models to perform the best, which motivated our second research question on embedded feature selection. Lastly, apart from accuracy, precision and recall are two common metrics used to measure performance in classification problems, hence this was investigated in our third research question.

1.1 Research Questions

- Is SVM the best-performing model for predicting income >$50k compared with Logistic Regression, Deicison Tree and Neural Network models?
- How well do the features chosen using embedded feature selection predict income >$50k?
- How well do precision-recall tradeoffs differ across the analyzed models?

1.2 Literature Review

The Adult Census Income dataset[1] has long served as a benchmark for predicting income above $50k based on different personal attributes. Dataset characteristics such as class imbalance and missing datapoints shapes our preprocessing and evaluation choices. When dealing with skewed binaries, Precision-Recall (PR) analysis has been demonstrated to be more informative than ROC [2], motivating our use of precision, recall alongside accuracy to evaluate under class imbalances.

Additionally, Decision Trees are classic, easy-to-read models whose depth can be tuned to control under and over-fitting[3]. This matches our depth sweep, and selection of a small and stable tree. Logistic regression is a standard linear baseline with interpretable coefficients, which is used to check whether a linear boundary is sufficient. Support Vector Machines (SVMs) can learn non-linear boundaries with the RBF kernel[4], justifying the use of SVM with a grid over C and λ with the best picked on validation. Neural Networks (NNs) are a modern option, but on medium tabular tasks like Adult, they don't always beat strong classical models[5], so NNs are used as comparators without the implication of improved performance.

Lastly, Adult includes 'sex' and 'race', so fairness is often discussed with this dataset. A common ides is equality of opportunity, which asks that True Positive (TP) rates be similar across groups[6]. We do not run a full fairness audit here, but our threshold and confusion-matrix reporting make the future analysis straightforward.

2 Methods

2.1 Overview of Dataset

The Adult dataset predicts if the annual income of an individual exceeds $50K. It contains 14 features that include the 'age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation', 'relationship', 'race', 'sex', 'capital-gain', 'capital-loss', 'hours-per-week', 'native-country' and 'income', the target variable. The target variable has 2 classes: '1' for if income exceeds $50K per year and '0' for otherwise. The dataset contains 48842 instances including missing datapoints.

2.2 Preprocessing

This section introduces the team's shared data preprocessing procedures. Some model-specific operations based on the need for models may be later applied individually.

2.2.1 Handling Missing Data

Records with missing data were removed from features 'workclass', 'occupation', and 'native-country' eliminating 3620 rows (7.41%), with total valid rows of 45222 after dropping. We verified that class prevalence changed from 0.239 to 0.248.

2.2.2 Feature Encoding and Selection

All categorical variables were converted to numerical variables using label encoding. The "fnlwgt" (final person weight) feature was removed, as it is an individual weighting placed on each datapoint, not an original feature of the Adult dataset. This individual weighting of the datapoint may influence machine learning algorithms unintentionally. Furthermore, weighting to each datapoint is not applicable to all models. If weighting is needed, the models have the capacity to assign weights.

The initial feature selection includes a visualization of an ANOVA table to understand the relevance of each feature at a higher level. The key metric used to filter features is Information Gain. All features with less than 0.02 IG were disregarded, leaving 10 features in total after preprocessing. Features removed in the team processing are 'workclass', 'race', 'native-country', 'has_capital_loss'. To smooth the skewness in 'capital-gain' and 'capital-loss', with few big values and mostly 0s, a log1p transformation was applied.

2.2.3 Imbalance Handling and Stratified Split

Analysis of the target variable showed the dataset is moderately imbalanced, with approximately 24% of instances belonging to the positive class (income > $50K) and 76% to the negative class (≤ $50K). This imbalance can cause models to be biased toward the majority class. To mitigate this issue, the parameter 'class_weight' = 'balanced' can be applied to assign proper classification cost when training. For model evaluation, PR-AUC was chosen as the primary performance metric, as it better reflects model quality under imbalance rather than overall accuracy. Accuracy was still reported for comparison, though it is less reliable when one class dominates.

Based on sufficient valid data records, the dataset was split into training, validation, and testing subsets using a stratified 80/10/10 division. This maintained consistent class proportions and ensured the evaluation remained representative of the true data distribution.

2.3 Machine Learning Methods

2.3.1 Decision Tree

One of the models chosen was the Decision Tree as this is a classification problem. The Adult dataset[1] also includes details about the different suitable machine learning models and one of them is the random forest classification. In an attempt to generalize the recommended model, the team decided to use the basic decision tree model to interpret the Adult dataset in this report to compare them and check the differences.

Hyperparameter tuning of the depth of the decision tree was performed to check the optimal depth that provides a balance between bias and variance. In general, a small tree (shallow depth) has a high bias and low variance and vice-versa for a big tree (deeper depth). Both extremes of tree depths give rise to distinct problems that this research aims to avoid. A smaller tree consists of less features and becomes too simple to model complex dataset relationships and a larger tree becomes overfitting of the dataset, thus, leading to a poor model in both cases.

Since this dataset contains 10 features, depths ranging from 1-11 have been considered. The criterion for the decision tree classifier was fixed to be entropy across all depths. This ensures that the decision at each node is made based on the most information gain from each feature. Future research recommends the use of Gain Ratio to

select features at roots to reduce bias of majority class.

The different Decision Trees of varying depths were compared using the standard metrics: accuracy, precision, and recall. Figure 1 is a visualization of the metrics for each decision tree classifier of varying depths to identify the optimal depth.

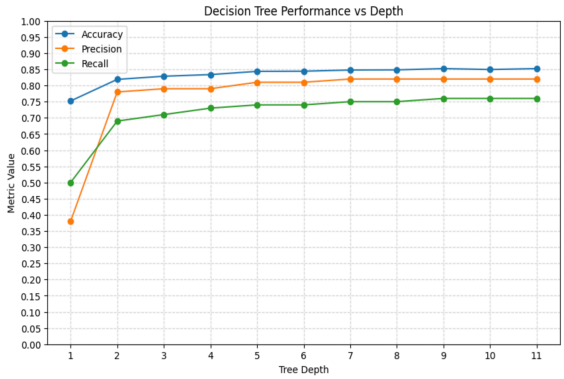

Figure 1: Decision Tree Performance at different tree depths

Initially, the plot shows that accuracy, precision, and recall increase rapidly as the depth of the tree increases to 2. After a depth of 2, there is a gradual and minor increase in performance until a depth of 5. The accuracy, precision and recall plateaus after a depth of 5 which depict no significant increase in performance for every new depth. The final parameter was trained on a dataset that included the training and validation combined and tested on the test data.

Thus, a decision tree classifier with a maximum depth of 5 was chosen as the best performing decision tree with a clear balance between bias and variance where the tree is not too simple and not overfitting the data.

2.3.2 Support Vector Machine

We include a Support Vector Machine with a Radial Basis Function (RBF) kernel as a non-linear classifier to complement linear/logit models and axis-aligned trees. SVMs capture complex boundaries in the Adult feature space. Class imbalance (24% positives) was handled with 'class_weight' = 'balanced'.

Preprocessing was achieved using the common pipeline from Section 2.2, that is, dropped rows with missing values, applies log1p to 'capital-gain/capital-loss', label-encoded categorical variables, and standardized all features (fit scaler on the training split only).

The validation protocol was performed using stratified 80/10/10 split (train/validation/test). Hyperparameters were selected on the validation set by macro-F1 using a grid $C \in \{0.1, 1, 3, 10\}$, $\gamma \in \{\text{scale}, 0.01\}$. For speed, tuning used 'probability=False' and the default SVM decision rule. The team then re-trained with probabilities for threshold selection and final evaluation.

Hyperparameters were selected using the best validation macro-F1 obtained at $C = 10$, $\gamma = \text{scale}$ (macro-F1 = 0.588, accuracy=0.659). Figure 2 shows that $\gamma = \text{scale}$ consistently outperforms $\gamma = 0.01$ across C, and performance improves with larger C (weaker regularization).

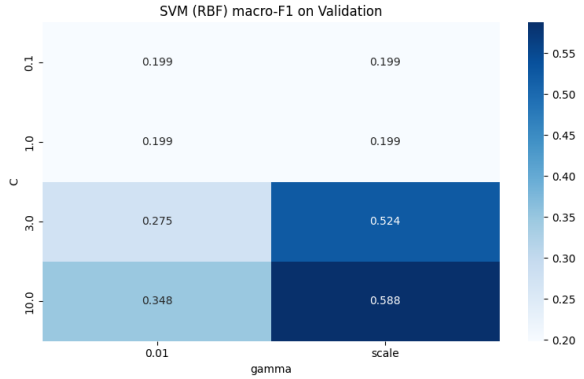

Figure 2: Validating macro-F1 for RBF-SVM across C and γ.

The probability threshold τ was chosen on validation using a model trained only on the train split to avoid leakage (with selected C, γ). We considered two operating points: (i) max-F1 (select τ that maximized F1 on validation), and (ii) precision-targeted (smallest τ achieving Precision ≥ 0.70 on validation). The team then refit SVM on train + validation with the same C, γ and applied the chosen τ to the held-out test set once.

2.3.3 Logistic Regression

This section uses a Logistic Regression model as a linear baseline classifier and provides interpretable model coefficients that are meaningful to study adult census data with multidimensional social attributes.

The highlight of the Logistic Regression model is its ability to explain, stable convergence and low computational cost. It assumes a linearly separable relationship between features and the

target. Each feature corresponds to a regression coefficient. A positive coefficient indicates the feature enhances the probability of high income, and a negative coefficient indicates the opposite effect. The magnitude of the coefficient reflects the strength of the influence. Unlike information gain that reflects overall correlation between individual features and the target, logistic regression coefficients represent the net effect after controlling other features. However, the limitation of Logistic Regression is its inability to capture non-linear interactions between features. It may not be as good as kernel or tree-based models with complex feature relationships.

In this model, based on the preprocessed features, the derived variable 'has_capital_gain' was also excluded as it is highly correlated with 'capital_gain'. Retaining both could result in extreme coefficient values. However, 'education' is a label encoding feature and 'education-num' is an ordinal numerical feature, so both are kept.

The model was implemented using scikit-learn with the following parameters: max_iter = 2000, class_weight = 'balanced', and random_state = 42. The primary hyperparameter in Logistic Regression is the regularization strength C that governs the trade-off between model complexity and generalization capability. As C value grows, regularization weakens, leading to more flexible models. This model employs C as its only tuning parameter, undergoing stratified five-fold cross-validation on the training set for C in {0.01, 0.1, 0.5, 1, 5, 10}, with F1-score serving as the evaluation metric. Results indicate that the model's F1-score remains around 0.61 across different C values and suggests its low sensitivity to regularization strength. C = 0.01 was finally selected as the optimal value when F1-score is the highest. During model validation, the classification threshold was tuned on the validation set by computing F1-scores for all thresholds returned by the Precision-Recall curve function. Using the decision function in Logistic Regression model, the final threshold is -0.075 on the logit scale.

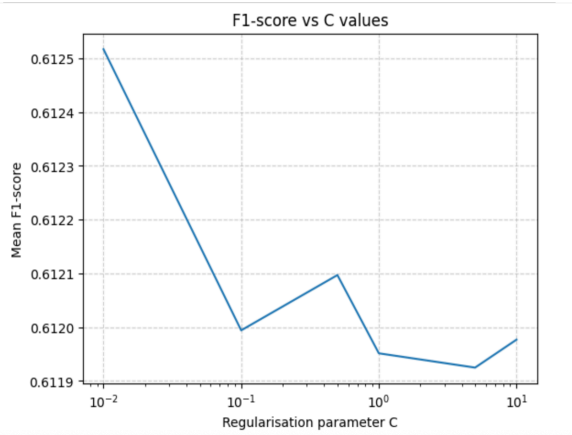

Figure 3: Tuning hyperparameter C on F1-score

2.3.4 Neural Networks (Sequencing)

Neural Networks is included to help capture complex, non-linear relationships in the adult feature space that simpler models may miss. Class imbalances were handled by assigning class weights to give higher importance to the minority class during training.

Preprocessing methods used were the same for both Neural Network models, it mainly followed the pipeline described in Section 2.2, missing values were dropped, numeric features were scaled using standardization, and categorical variables were encoded using the label encoder. Input features and output labels were converted to float 32 to ensure stability. The activation (ReLU) and loss functions (binary cross-entropy) for both NNs are fixed.

2.3.4.1. Lightweight Neural Network

For this non-linear classification problem, a sequential neural network with 3 hidden layers has been used. The neurons in each layer contain 128, 64 and 32 neurons respectively in line with the common rule of thumb to have no more than ½ the previous layer. The activation function used on the hidden layers is Rectified Linear Unit (ReLU) to add non-linearity and to model complex relationships. The output layer uses the sigmoid activation function as this is a binary classification problem. The dropout technique has been used for regularization to reduce overfitting of the data while training the model. The dropout rate for the 1st and 2nd hidden layers are 0.3 and 0.2 respectively.

The model uses the Adam optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function, a typical function in binary classification

problems. The accuracy of the model is tracked while training the model in Keras. Early stopping during training has also been done to reduce overfitting. Model has been trained for 50 epochs and a sample of 32 each time.
Thresholds ranging from 0.1 to 0.9 were checked. The parameter was tuned on the basis of accuracy, precision and recall. The threshold parameters were tuned using the validation dataset.

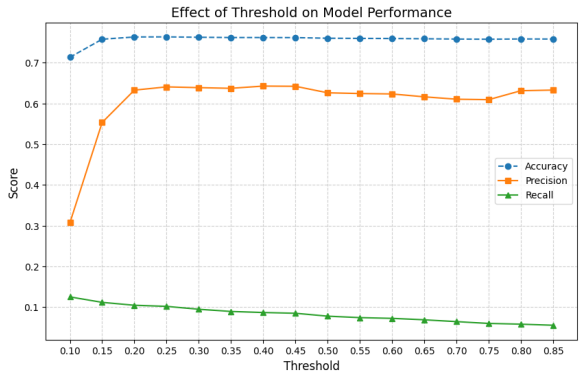

Figure 4: Tuning hyperparameter Threshold for Lightweight Neural Network

In Figure 4, it is evident that 0.5 is the most ideal threshold as the accuracy is the maximum at this point and a balance between the precision and recall have been achieved. There's no significant change in the metrics after 0.20 threshold which could be due to possible overfitting.

2.3.4.2. Dense Neural Network

For the second model a deep sequential neural network architecture was employed to capture complex interactions within the dataset. The model consisted of four hidden layers with 512, 256, 128 and 64 neurons, respectively. Each hidden layer uses the hyperbolic tangent (tanh) activation function to model non-linear feature relationship and enable a smoother gradient flow compared to ReLU in certain contexts. To stabilize and accelerate training, Batch Normalization is applied after each hidden layer. This standardizes intermediate activations, mitigating internal covariate shift and improving generalization.
To further prevent overfitting, dropout regularization was used after the first three hidden layers, with dropout rates of 0.4, 0.3, and 0.2, respectively. The output layer employs a sigmoid activation function producing probabilities for the binary income classification task.
Like the previous Neural Network model, the Adam optimizer with a learning rate of 0.001 was used along with the binary cross-entropy loss function. Model performance during training was monitored using accuracy as a primary metric. Class imbalance (approximately 23% positive samples) was addressed by applying class weights computed from the training labels, ensuring that the minority '>50K' class is given importance.
Post training the model was analyzed by varying the decision threshold from 0.1 to 0.9. For each threshold, accuracy, precision, and recall were computed on the validation set. The results are shown in Figure 5.

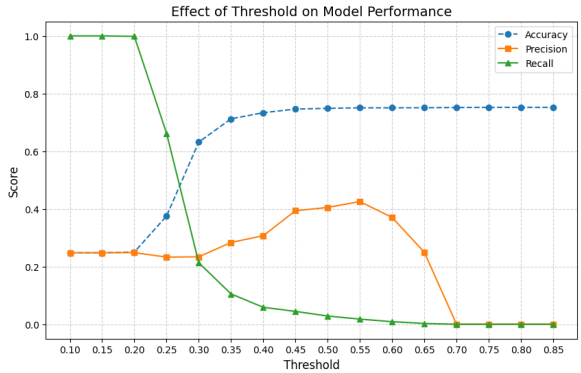

Figure 5: Tuning hyperparameter Threshold for Dense Neural Network

In Figure 5, the high recall that starts at 1.0 which shows that the model predicts nearly all samples as belonging to the positive class correctly, but the precision is low due to the large number of false positives. As the threshold increases, the model's recall takes a sharp drop and the precision increases. A threshold of 0.5 was chosen since it provided limited trade-off between precision and recall while maintaining a decent accuracy.

3 Results

3.1 Decision Tree

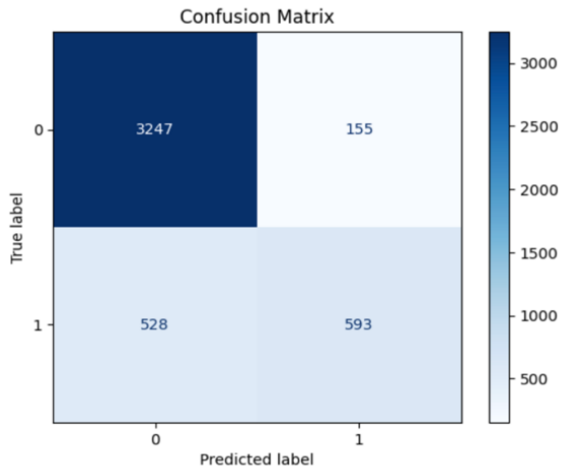

Figure 6: Confusion Matrix for the Decision Tree Classifier

The results shown are for a decision tree classifier that uses entropy with a maximum depth of 5.
The model achieved Accuracy = 0.85, Precision = 0.83, and Recall = 0.74.
Both the accuracy and precision are reasonably high compared to recall. This accuracy of this model is the same as the random forest model mentioned in the dataset. The precision of this model is higher than the precision of the random forest[1]. Noticeably, the number of false positives is higher than the true negatives in Figure 6.
The precision-recall curve (AUC = 0.705) was generated based on average precision scores seen in Figure 7.

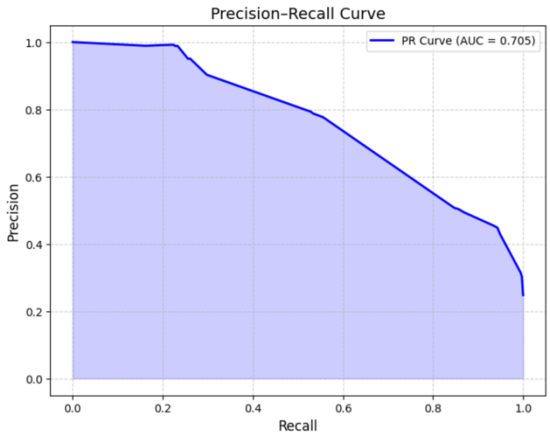

Figure 7: Precision-Recall Curve for the Decision Tree

3.2 Support Vector Machine

The first operating point, Max-F1 threshold (chosen on validation) can be seen in Figure 8, Confusion Matrix demonstrating True Positive rate at 2,274, False Positive at 1,128, False Negative 90, and True Positive 1,041. Metrics can be seen as Precision = 0.480, Recall = 0.929, F1 = 0.633, and Accuracy = 0.733. This setting prioritizes recall (few false negatives) at the cost of more false positives.
The second operating point, Precision-targeted threshold (validation constraint Precision ≥ 0.70, and τ = 0.515) obtained Confusion Matrix in Figure 9, with True Positive at 2,911, False Negative at 491, False Positive 300, and True Negative 821. Metrics from the matrix are Precision = 0.626, Recall = 0.732, F1 = 0.675, and Accuracy = 0.825. Enforcing a precision constraint substantially reduces False Positives, improving both overall accuracy and F1 relative to the max-F1 point, while lowering recall.
The test Precision-Recall curve in Figure 10 has AUC = 0.721, well above the positive-class prevalence (0.239), indicating useful ranking quality.

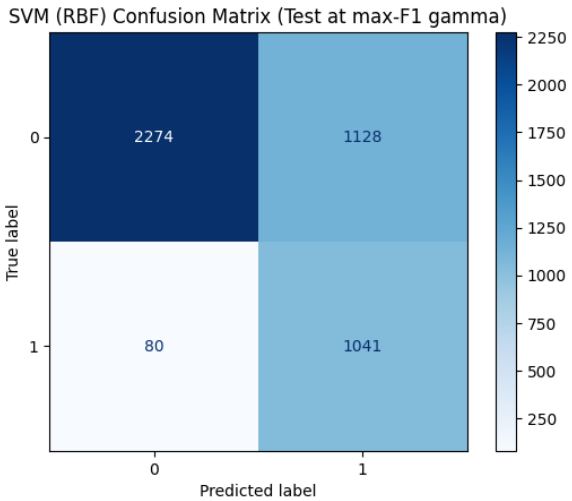

Figure 8: Confusion matrix on the test set at the max-F1 operating point for SVM classifier

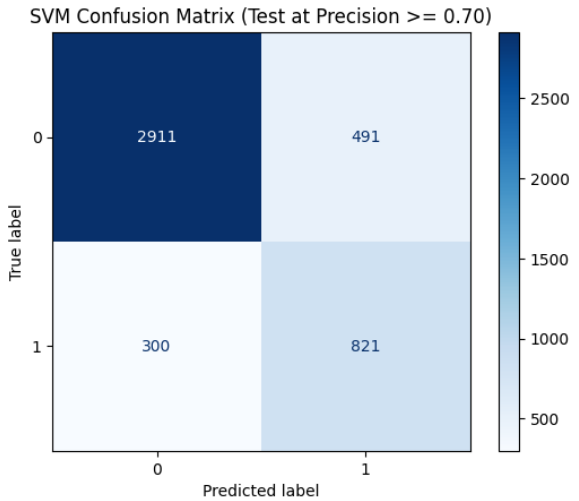

Figure 9: Confusion matrix on the test set at the Precision ≥ 0.70 operating point for SVM classifier

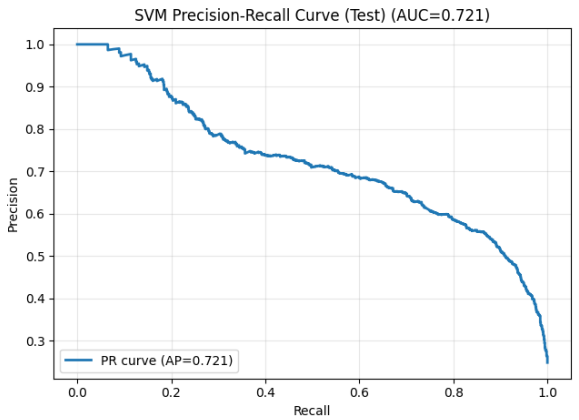

Figure 10: Precision-Recall Curve for the SVM Classifier

3.3 Logistic Regression

The model achieved Precision = 0.49, Recall = 0.77 and Accuracy = 0.74 as evident in Figure 11. It maintained stable and moderate predictions, but its linear assumption limited its ability to capture more complex feature interactions.

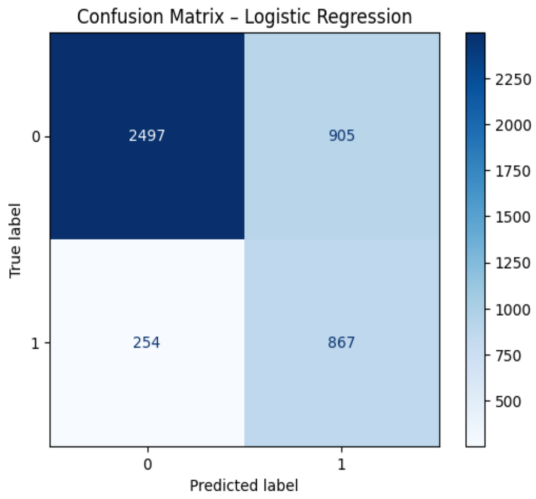

Figure 11: Confusion Matrix for the Logistic Regression Classifier

The coefficient results show that sex has the strongest influence on classification. Education (num) is also positively correlated with classification. Capital gain and capital loss suggest investment-related features are positive contributors to classification. Relationship-related features are negative contributors.

Top 10 feature coefficients

	feature	coef	abs_coef
6	sex	0.808062	0.808062
2	education–num	0.335886	0.335886
3	marital–status	−0.257892	0.257892
7	capital–gain	0.203652	0.203652
8	capital–loss	0.160920	0.160920
5	relationship	−0.108717	0.108717
0	age	0.040028	0.040028
9	hours–per–week	0.032897	0.032897
1	education	0.016010	0.016010
4	occupation	−0.000210	0.000210

Table 1: Interpretable coefficients from the Logistic Regression model

The precision-recall curve (AUC = 0.649) was generated based on average precision scores seen in Figure 12.

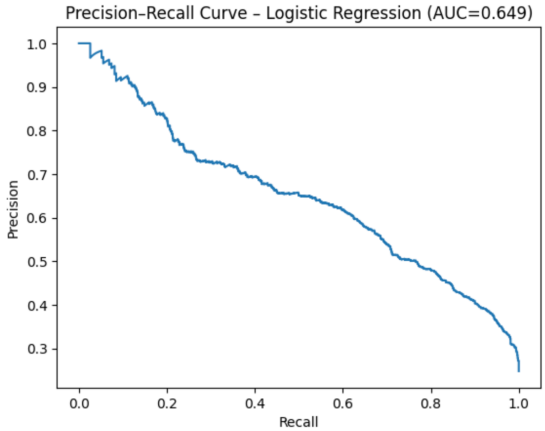

Figure 12: Precision-Recall Curve for the Logistic Regression Model

3.4 Lightweight Neural Networks

The best threshold chosen was 0.5. The model achieved an accuracy = 0.84, precision = 0.75 and recall = 0.54 seen in Figure 13. It achieved the second highest number of true positives is the. Both accuracy and precision are reasonably high, but recall is low. The very high false positives (467) reduce the performance of this model.

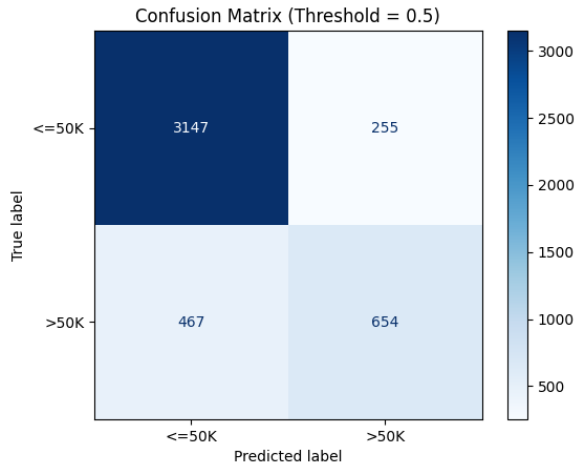

Figure 13: Confusion Matrix for the Lightweight Neural Network Model

The average precision scores of the test data have been used to generate the precision-recall curve (AUC = 0.749) in Figure 14.

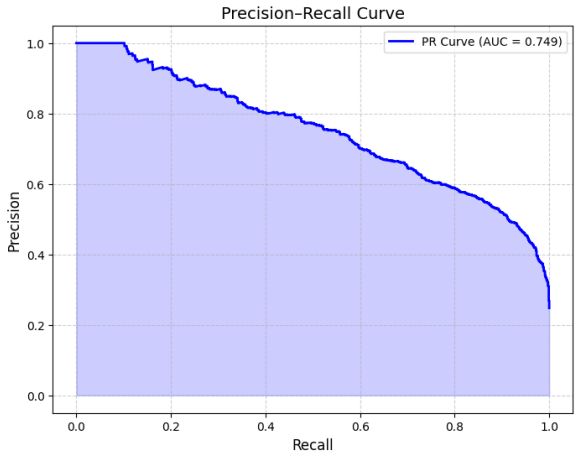

Figure 14: Precision-Recall Curve for the Lightweight Neural Network Model

3.5 Dense Neural Networks

The best threshold chosen was also 0.5. Both confusion matrices derive similar results as seen in Figure 15.

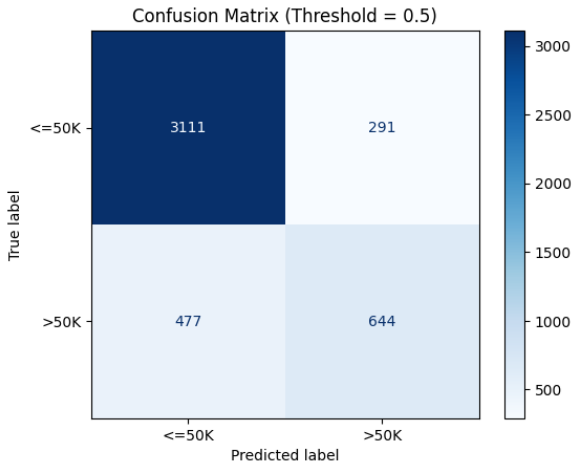

Figure 15: Confusion Matrix for the Dense Neural Network model

The average precision scores of the test data have been used to generate the precision-recall curve (AUC = 0.725) in Figure 16. Accuracy = 0.83 Precision = 0.68 Recall = 0.62, and PR-AUC = 0.72.

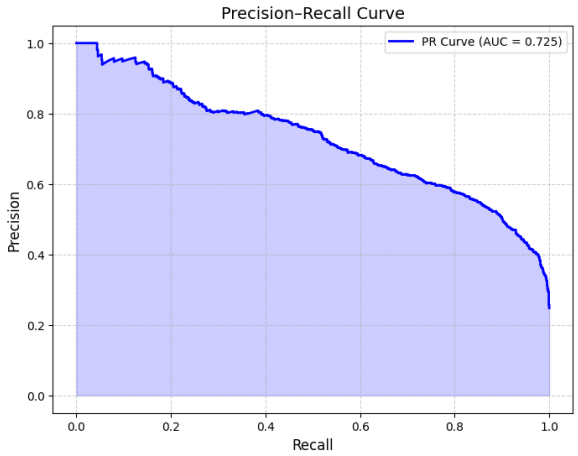

Figure 16: Precision-Recall Curve for the Dense Neural Network Model

3.6 Results Summary of all Analyzed Models

All the models analyzed have checked the accuracy, precision, recall, and F1-score to evaluate the best model after hyperparameters' tuning. However, the final analysis of this report mainly focuses on 3 metrics – accuracy, precision and recall while also considering the confusion matrix to determine the best performing model.

Model	Accuracy	Precision	Recall
Decision Tree	0.85	0.83	0.74
SVM	0.73	0.48	0.93
Logistic Regression	0.74	0.49	0.77
Lightweight Neural Network	0.84	0.72	0.57
Dense Neural Network	0.83	0.68	0.62

Table 2: Summary of all the metrics for each analyzed model

4 Discussion

Based on the results in Table 2, it is evident that the Decision Tree classifier outperforms all models as its accuracy and precision are significantly higher compared to the other models while also has a good recall.

Both the decision tree and both neural network models have approximately the same accuracy which suggests that these models are ideal to predict the dataset based on our analysis. The precision of the decision tree is much better than both neural networks, but the recall of both the neural network models are quite low meaning the false negatives were very high. Both the SVM and Logistic Regression models are indifferent based on accuracy and precision. Although recall is quite high, the precision of both these models is only around 50% which is very low. The high number of false positives in these models decrease their reliability for prediction. The recall metrics of SVM is exceptionally higher than all other models since the false negatives are the lowest compared to the other models. Conclusively, the decision tree model has a balanced performance in predicting this dataset.

Since the data was not linearly separable, we decided to analyze suitable models such as the Decision Tree, SVM and Neural Networks. Logistic Regression was performed to validate our intuitive conclusion of the data being non-linear. Knowing that the data was not linearly separable, we had chosen a non-linear kernel RBS for the SVM model to find a suitable non-linear decision boundary. A reasonable variety of C values were used from 0.1 to 10. Increasing C has evidently increased performance in this analysis. Future research in this model could possibly trial even more values of C to find a better optimum as C=10 was chosen for the final SVM. A major limitation of both SVM and Neural Networks was the basic feature selection done using information gain. Generally, one-hot-encoding improves the performance of Logistic Regression, SVMs, and Neural Networks as categorical data is in a non-ordinal numeric form. Unfortunately, due to time constraints, the preprocessing step could not include this feature selection process. This is one of the drawbacks of this analysis as this may have distorted the results, proposing a recommendation for future research.

Since there is a notable imbalance between those <$50K and >$50K income, both the SVM and Neural Networks opted for balanced classifiers to reduce any bias towards <$50K class. Different architectures for the Neural Networks were investigated.

As learnt from lectures, Logistic Regression assumes that the features are independent and there is a linear relationship between them which is incorrect in this case. For example, 'education' and 'occupation' are two features that are very dependent on each other. So, we expected the logistic regression model to perform the worst but surprisingly the model is indifferent to the SVM except for the recall metric. This could be due to the deviation from the other models with the usage of cross-validation (5-fold) to tune the model. Since a large dataset was available, train-

validation-test split was deemed sufficient, but team member decided to trial this approach for this model as it was known to perform the worst.

Two Neural Network models were trained, one using 'ReLU' activation function, and the second using 'hyperbolic tan'. Checking more Neural Network models with different architectures for future research may possibly provide different results. While the sequencing model is suitable for non-linear data, it is the simplest Neural Network model available. It only supports linear layer stacking. It also has limited flexibility which could potentially be the reason why the decision tree classifier was slightly better than the sequence Neural Networks.

Comparing all the confusion matrices, the confusion matrix for SVM (Figure 9) depicts the highest number of false negatives which degrades its performance relative to the other models. We can safely conclude that SVM is not the best performing model, which could partly be explained by the pitfalls in feature encoding.

In general, the team had decided to do additional embedded feature selection for each of the models individually to check which features provide the best models. This was naturally followed by the Decision Tree as features are selected at each root. However, this was not the case for all the other models. The only additional change made to the features was scaling using a standard scaler for both the SVM and the neural network. This would be an extension task for any future research for more meaningful results and thorough comparison.

Regarding the precision-recall curves, the team decided that this would be ideal to validate the performance of a binary classification model with imbalanced classes. Figure 7 shows that the PR curve is reasonably close to the top right corner for the Decision Tree classifier. This manifests the fact that the decision tree classifier is indeed a good performing model. Interestingly, Figure 14 shows that the Lightweight Neural Network model PR curve has an AUC close to 0.75 which is slightly higher than the AUC of 0.70 for the decision tree. In fact, the AUC of all the models except Logistic Regression is higher than the AUC of the Decision Tree PR curve. In this case, performance was primarily evaluated based on the proximity of the curves to the top right corner of the graph. The AUC criterion was secondary as the discrepancies were considered insignificant to determine the performance of the models.

5 Conclusion

Based on the report analyses, it is clear that the Decision Tree classifier is the best performing model for this dataset, which defeats the initial assumption of SVM outperformance. The second research question was partially answered, and all features included in the models after preprocessing were relatively sufficient to train the models. In future, subsets of features could be tested to further validate this research. The AUC of all the PR curves did not differ much between the models, but the shape of the curves was noticeably different. The tradeoffs between precision and recall have been depicted reasonably well by these curves across all models. Lastly, the analyses demonstrate the following ranking of Machine Learning models for the chosen task: Decision Tree, Neural Networks, SVMs, and Logistic Regression, though further research has been recommended.

6 Limitations

- The Decision Tree classifier uses information gain instead of gain ratio to prioritize features at every node. Using gain ratio removes the bias of favoring features with more instances.
- Using one-hot-encoding during preprocessing would have been better feature selection for the SVM and Neural Network models.

Generative AI Usage

Generative AI was used for guidance in the formulation of Research Questions and idea generation for discussion of different models. It was not used in any of the report or writing. All analysis was done as a team without the influence of AI.

Bibliography

[1] https://archive.ics.uci.edu/dataset/2/adult

[2] Davis, Jesse, and Mark Goadrich. "The Relationship between Precision-Recall and Roc Curves." Paper presented at the Proceedings of the 23rd international conference on Machine learning, 2006.

[3] Breiman, Leo, Jerome Friedman, Richard A Olshen, and Charles J Stone. Classification and Regression Trees. Chapman and Hall/CRC, 2017.

[4] Cortes, Corinna, and Vladimir Vapnik. "Support-Vector Networks." Machine learning 20, no. 3 (1995): 273-97

[5] Shwartz-Ziv, Ravid, and Amitai Armon. "Tabular Data: Deep Learning Is Not All You Need." Information Fusion 81 (2022): 84-90.

[6] Hardt, Moritz, Eric Price, and Nati Srebro. "Equality of Opportunity in Supervised Learning." Advances in neural information processing systems 29 (2016).