Teesside University

School of Computing, Engineering & Digital Technologies

MACHINE LEARNING IN-COURSE ASSESSMENT

SALARY CLASSIFICATION PREDICTION USING SEVERAL MACHINE LEARNING MODELS

NAME: CHUKWURAH PAUL

STUDENT ID – B1267718

SUBMISSION DATE: 10/05/23

ABSTRACT

This study explores the use of several machine learning models for salary classification prediction, with a focus on identifying the best performing model based on accuracy and the most important features in the prediction. The models were utilized to classify an individual's annual salary as being either <= 50k or >50k. The dataset used was obtained from UCI repository which included 15 attributes, such as salary, workclass, education, occupation, relationship, race, native country, etc. After conducting exploratory data analysis, pre-processing data, and feature engineering, the dataset was divided into test (30%) and train (70%) datasets, and four algorithms were applied, namely Logistic Regression (LR), K Nearest Neighbour (KNN), Random Forest (RF), and Decision Tree (DT). Random Forest produced the highest accuracy of 87% after balancing the dataset and adjusting certain hyperparameters of the model like the max_depth and n_estimators. Thus, it is considered the best-performing model in this study. The most important features in predicting an individual's salary class in the Random Forest model were identified as capital-gain, marital-status, hours-per-week, and education-num.

Keywords: Salary, Classification; Logistic Regression; KNN; Random Forest; Decision Tree

TABLE OF CONTENTS

1.0 INTROUDCTION

In today's data-driven world, predicting an individual's salary classification has become an important task in many industries such as banking, finance, and marketing. By analysing various factors that affect salary, such as education level, years of experience, job title, and industry, companies can gain valuable insights into the economic status of their customers and target their products and services more effectively. Machine learning models have emerged as a powerful tool for predicting salary classification, with the potential to achieve high accuracy rates and improve business decision-making. This study explores the use of several machine learning models for salary classification prediction, with a focus on comparing their performance and identifying the most important features.

Research questions

This report aims to answer the following research questions.

1. Which machine learning model performs the best in predicting salary classification, and why?
2. What are the most important features in predicting salary classification?

1.1 Data Description

The dataset used for this study was obtained from UCI repository. It comprises of 32,561 rows and 15 columns, consisting of 8 categorical labels and 7 numerical labels. The salary column serves as the target variable, indicating if an individual earns <=50k/year or not. The table below gives a brief description of each attribute present in the dataset.

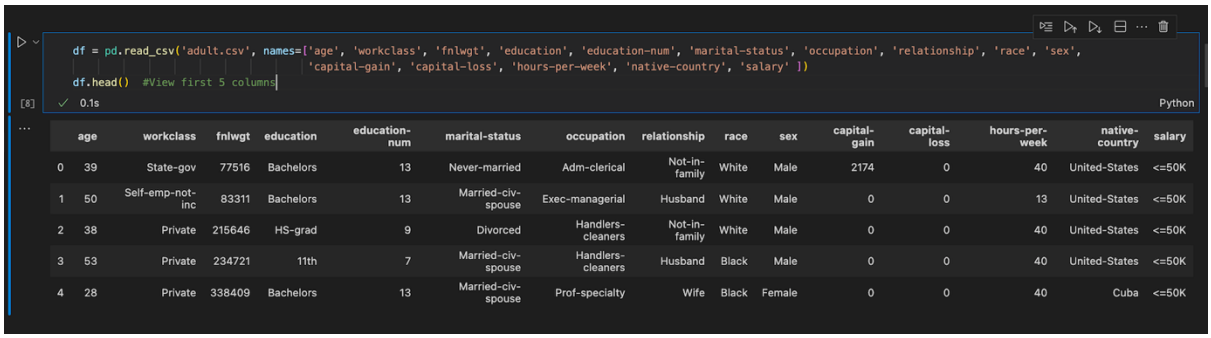
NO	NAME	DESCRIPTION
1	Age	age of persons
2	Workclass	describe work type
3	Fnlwgt	financial weight
4	Education	education level
5	education-num	education level number
6	marital-status	marital status
7	capital-gain	capital gain
8	capital-loss	capital loss
9	Occupation	work or business

10	Relationship	relationship status
11	Race	race
12	Sex	gender
13	hours-per-week	number of hours put in a week
14	native-country	country
15	salary	Whether salary is >50k or not

Table 1: Description of dataset

2.0 DATA EXPLORATION AND FEATURES SELECTION

The first stage of data analysis involves data exploration, where data is visualized to extract insights or detect patterns that warrant further investigation.



```
df = pd.read_csv('adult.csv', names=['age', 'workclass', 'fnlwgt', 'education', 'education-num', 'marital-status', 'occupation', 'relationship', 'race', 'sex',
                                     'capital-gain', 'capital-loss', 'hours-per-week', 'native-country', 'salary' ])
df.head()  #View first 5 columns
```

[8] ✓ 0.1s Python

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K

Fig 1. Viewing the first five rows of the dataset

Observation

- We have both categorical and numerical variables.

```
df.info()
[3] ✓ 0.1s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32561 entries, 0 to 32560
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             32561 non-null  int64
 1   workclass       32561 non-null  object
 2   fnlwgt          32561 non-null  int64
 3   education       32561 non-null  object
 4   education-num   32561 non-null  int64
 5   marital-status  32561 non-null  object
 6   occupation      32561 non-null  object
 7   relationship    32561 non-null  object
 8   race            32561 non-null  object
 9   sex             32561 non-null  object
 10  capital-gain    32561 non-null  int64
 11  capital-loss    32561 non-null  int64
 12  hours-per-week  32561 non-null  int64
 13  native-country  32561 non-null  object
 14  salary          32561 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

Fig 2. Dataset information

- Our target variable is the salary column.
- Salary is an object not an integer.

```
df.describe().T
[13] ✓ 0.0s
```

	count	mean	std	min	25%	50%	75%	max
age	32537.0	38.585549	13.637984	17.0	28.0	37.0	48.0	90.0
fnlwgt	32537.0	189780.848511	105556.471009	12285.0	117827.0	178356.0	236993.0	1484705.0
education-num	32537.0	10.081815	2.571633	1.0	9.0	10.0	12.0	16.0
capital-gain	32537.0	1078.443741	7387.957424	0.0	0.0	0.0	0.0	99999.0
capital-loss	32537.0	87.368227	403.101833	0.0	0.0	0.0	0.0	4356.0
hours-per-week	32537.0	40.440329	12.346889	1.0	40.0	40.0	45.0	99.0

Fig 3. Dataset description

```
df.duplicated().sum()
[4] ✓ 0.0s
24
```

Fig 4. Check for duplicated values.

```
df.drop_duplicates(inplace=True)  # Drop duplicates
[5] ✓ 0.0s
```

Fig 5. Dropping duplicates.

- 24 duplicates were found and removed from the dataset.

Fig 6. Checking for null values.

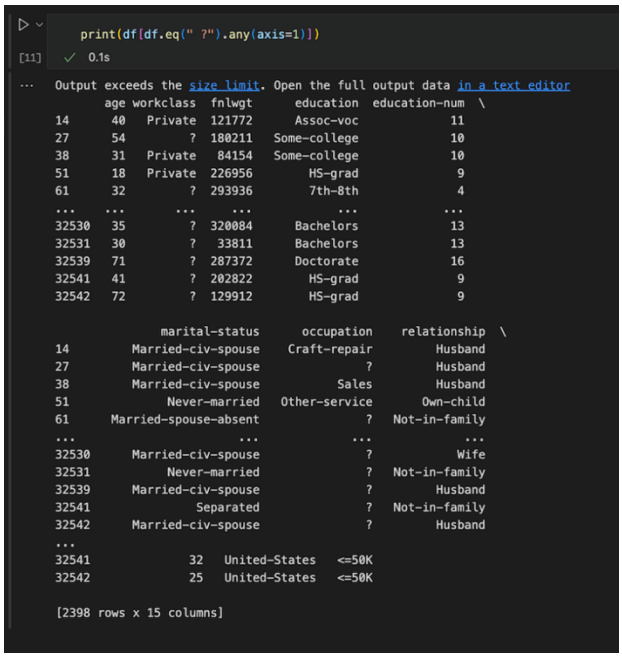


Fig 7. Number of unique values per column

- Although no missing values were identified using the "isnull" method, closer examination revealed that some columns, such as workclass, native country, and occupation, contained missing values represented as question marks. This would affect our training, and therefore, the missing values must be addressed.

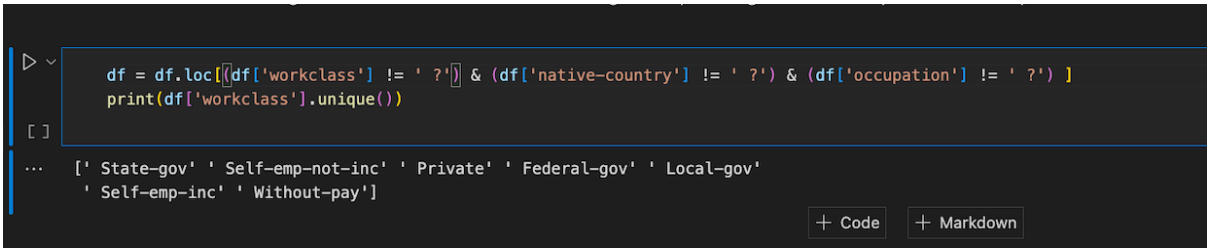


Fig 8. Removing question marks from dataset

Fig 9. Showing the unique values for each column

2.1 Visualisations

2.1.1 Numerical Analysis:

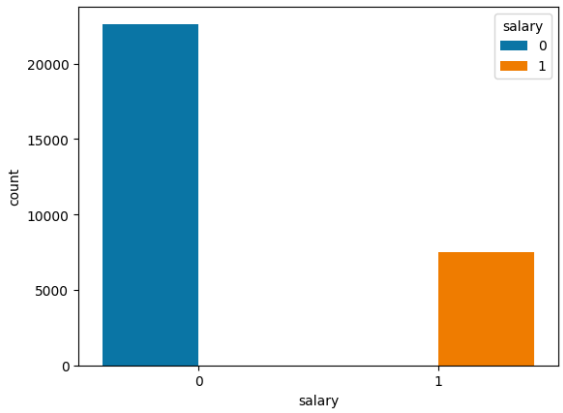


Fig 10. Bar chart showing a summary of the target variable.

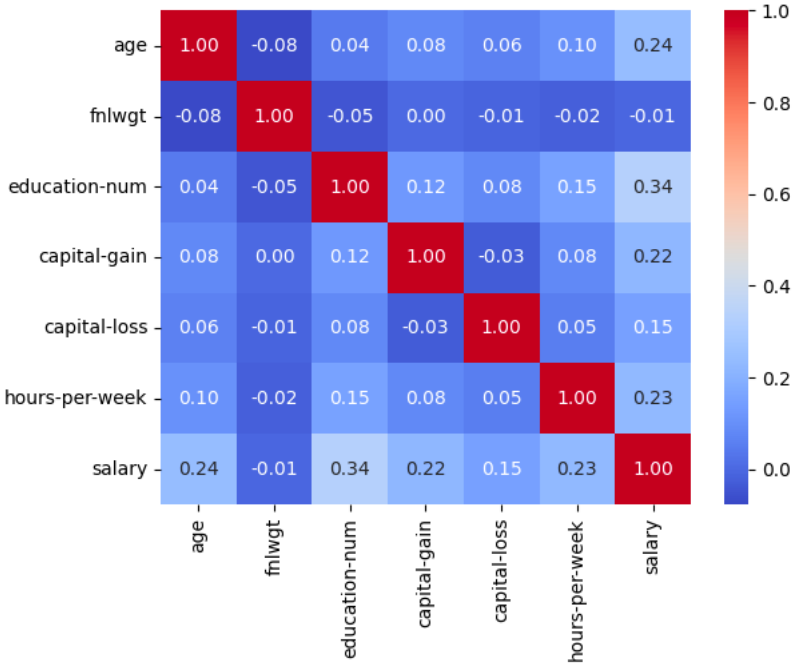


Fig 11. Correlation heatmap between the features.

Observation

- Almost 75% (22654) of the instances of our target variable is '<$50k'
- 7508 (25%) have a salary above $50k
- We have an imbalanced data.
- No highly correlated column observed
- We can use accuracy, precision, recall and F1-score.

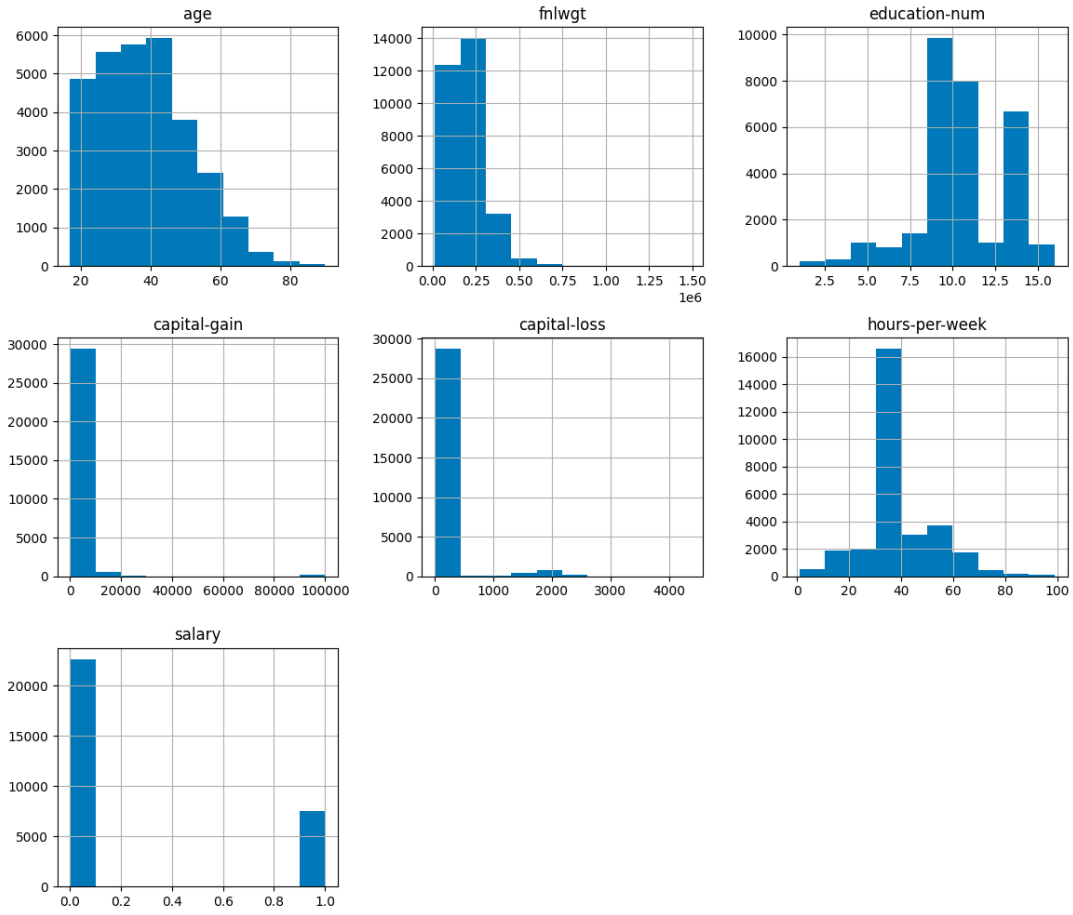


Fig 12. Plot of all numerical labels.

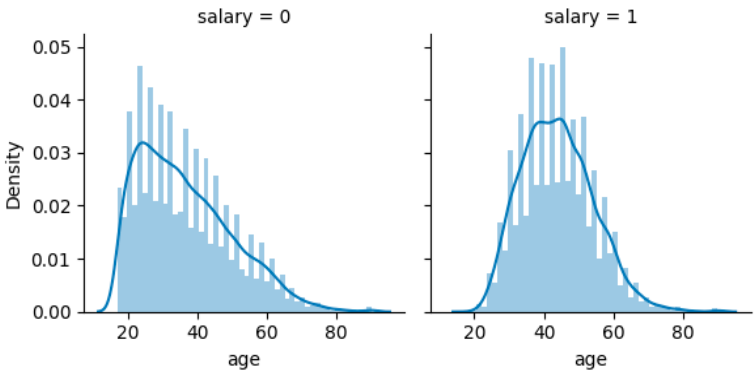


Fig 13. Age vs Salary

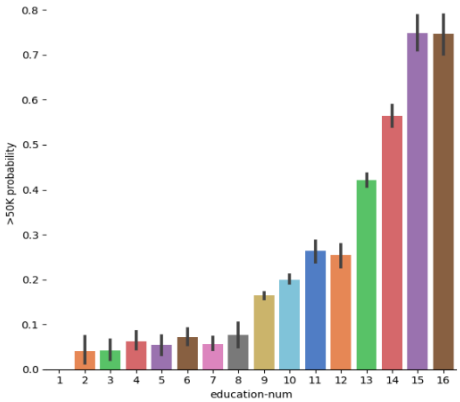


Fig 14. Education-num vs Salary

Observations

- The fnwlgt, capital-gain, capital-loss are skewed towards the right.

2.1.2 Categorical analysis:

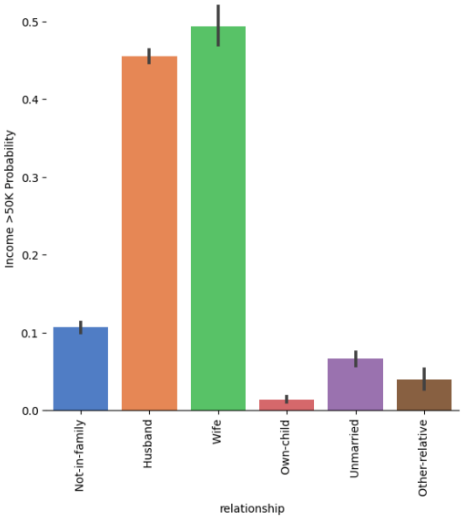


Fig 15. Relationship vs Salary.

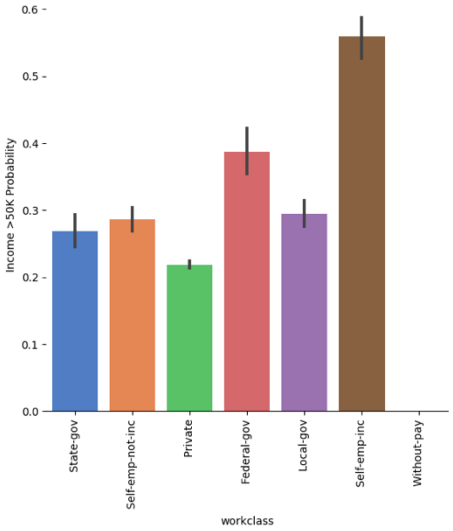


Fig 16. Workclass vs Salary

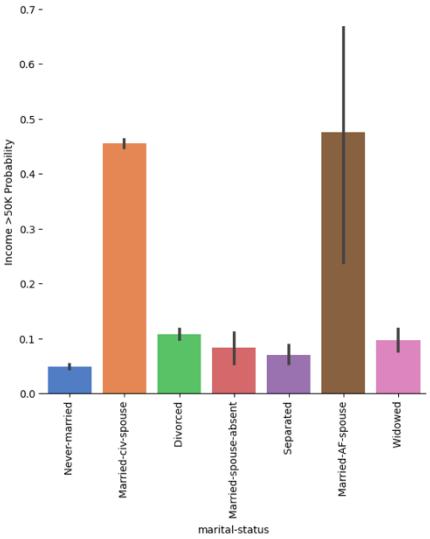


Fig 17. Marital-status vs Salary

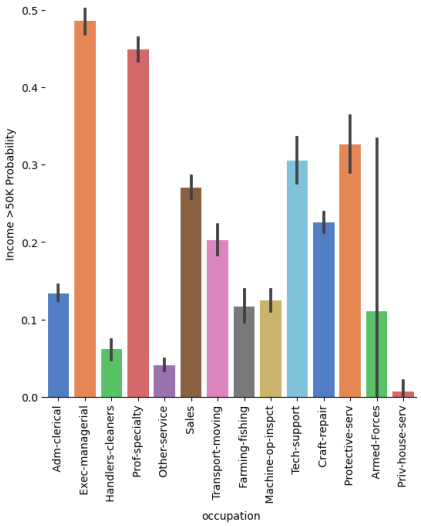


Fig 18. Occupation vs Salary

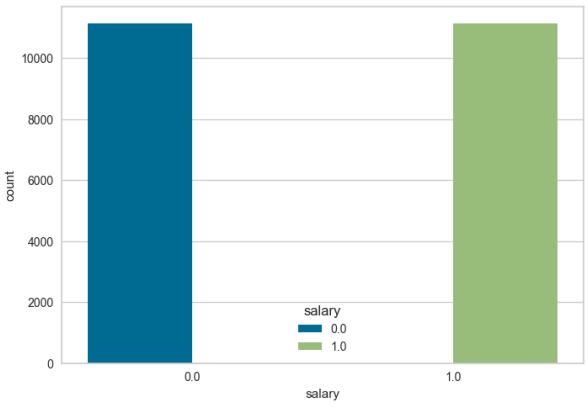


Fig 19. Plot of resampled target using smote

After conducting exploratory data analysis, and dealing with missing values, duplicated values, the next step would be to engineer the data for selection purposes.

2.2 Feature Engineering

The dataset was split into two categories. Numerical and Categorical. The numerical group included fnlwgt, education-num, capital-gain, capital-loss, hours-per-week, salary. While the Categorical group included, education, marital-status, occupation, relationship, race, sex, native-country.

2.2.1 Replacing column values:

The values in the sex column were substituted with 1 and 0, representing male and female, respectively. Similarly, the salary column values were replaced with 1 and 0, indicating <=50k and >50k, respectively. For the marital-status column, values such as Never-married, divorced, widowed, and separated were grouped as single, while values such as Married-civ-spouse, Married-spouse-absent, and Married-AF-spouse were grouped as married. Consequently, single was replaced with 0, and married with 1.

2.2.2 Converting column data type:

The data types for both the marital-status and salary columns were converted from object to integer.

2.3 Feature Encoding

It involves converting categorical data into numerical data.

```
d = LabelEncoder()

for col in df[categorical_labels]:
    df[col]= d.fit_transform(df[col])
df.head()
```

[354] ✓ 12.3s Python

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	39	5	77516	9	13	0	0	1	4	0	2174	0	40	38	0
1	50	4	83311	9	13	1	3	0	4	0	0	0	13	38	0
2	38	2	215646	11	9	0	5	1	4	0	0	0	40	38	0
3	53	2	234721	1	7	1	5	0	2	0	0	0	40	38	0
4	28	2	338409	9	13	1	9	5	2	1	0	0	40	4	0

Fig 20. Label Encoder

2.4 Scaling

Scaling is done to transform the numerical features of a dataset to a common scale. The transformation is achieved by subtracting the minimum value of the feature and then dividing by the range (max value - min value) of the feature. The scaling was applied to the numerical labels.

```
scaler = MinMaxScaler()
df[numerical_labels] = scaler.fit_transform(df[numerical_labels])
df.head()
```

[355] ✓ 19.9s Python

	age	workclass	fnlwgt	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country	salary
0	0.301370	5	0.043338	9	0.800000	0	0	1	4	0	0.02174	0.0	0.397959	38	0.0
1	0.452055	4	0.047277	9	0.800000	1	3	0	4	0	0.00000	0.0	0.122449	38	0.0
2	0.287671	2	0.137244	11	0.533333	0	5	1	4	0	0.00000	0.0	0.397959	38	0.0
3	0.493151	2	0.150212	1	0.400000	1	5	0	2	0	0.00000	0.0	0.397959	38	0.0
4	0.150685	2	0.220703	9	0.800000	1	9	5	2	1	0.00000	0.0	0.397959	4	0.0

Fig 21. Scaling of dataset

2.5 Feature Selection

The next step involved splitting the dataset into features and labels. In machine learning, the features constitute the input to the model, whereas the labels correspond to the output. Typically, they are split into two sets: X for the features and Y for the labels. After conducting several experiments, it was found that the variables "workclass", "relationship", "race", and "native country" did not significantly contribute to the target value. Therefore, these variables were dropped from the feature set (X). The dataset was split into training and testing samples, with 70% of the data assigned to the training set and 30% assigned to the testing set as this was what produced the best result compared to 80% - 20% splitting.

3.0 EXPERIMENTS

The reason for selecting this dataset is that it provides comprehensive information on an individual's health and lifestyle, which can be utilized to predict their salary level. Identifying an individual's salary class is significant as it can aid financial institutions, such as banks and credit agencies, in assessing their creditworthiness and repayment capacity. Moreover, Salary classification is an important tool for understanding the economic and financial status of individuals and can inform a wide range of policy and business decisions.

3.1 Algorithm used and why?

- Logistic Regression (LR)

  It is used to analyse the relationship between a categorical dependent variable and one or more independent variables. It estimates the probability of an event occurring by fitting the data to a logistic curve (Park, H. 2013). It was chosen because it can handle imbalance in the dataset by using techniques such as oversampling, under sampling, and cost-sensitive learning.

- K-Nearest Neighbour (KNN)

  KNN works by finding the K closest data points in the training set to a new data point, and classifying its value based on the majority class or average value of its K nearest neighbours (https://scikit learn.org/stable/modules/neighbors.html). One of its advantages is that it can handle non-linear data and is robust to noisy data which is why it was chosen.

- Random Forest (RF)

  Random forest is an ensemble method that aggregates the predictions of numerous decision trees to provide a final prediction. It is a very robust classification algorithm because it is not prone to over-fitting (Breiman, 2001; Cutler et al., 2007). It was chosen because of its high accuracy, less tendency to overfit and a novel method of determining variable importance.

- Decision Tree (DT)

  Decision tree constructs a tree-like model for decision-making by partitioning the input data into smaller subsets based on the values of the input features. The tree structure is composed of internal nodes that represent tests on feature values, branches that represent the possible outcomes of the tests, and leaf nodes that represent the final decision or classification (Rokach et al., 2006). One of the best reasons for choosing decision tree in binary classification is its ability to handle non-linear relationships between features and the target variable.

3.2 Hyperparameter Tunning

Most machine learning algorithms work well with default hyper-parameters specified in the software library. However, hyperparameter tuning can enhance the performance of these algorithms. Grid Search CV is a popular approach to hyperparameter tuning that involves exploring a range of hyperparameter values exhaustively (James et al., 2013). Grid Search CV was used to tune the hyperparameters for all models. A range of hyperparameter values were supplied to the library, and cross-validation was performed to determine the best combination of hyperparameters. For instance, in KNN, the number of neighbours and the distance measure all had a range of values. For Random Forest, the maximum tree depth, and the number of trees in the forest were given a range of values. Similarly, for Decision Tree, the maximum tree depth, the minimum number of samples required to split, and the maximum number of features were given different values to determine the best combination.

3.3 Experiment 1

Two set of experiments were carried out in this study. In this experiment, the imbalanced dataset was split into features and label and all four models were applied to the dataset. Hyperparameters were tuned and the dataset was re-evaluated using the new combination of hyperparameters obtained. Performance metrics for were then developed for each model.

3.4 Experiment 2 (Addressing the imbalance)

The initial dataset showed a noticeable class imbalance, with around 75% of the samples belonging to class '0' and only 25% belonging to class '1' In this experiment, the imbalance in the dataset was addressed using SMOTE (Synthetic Minority Over-sampling Technique). SMOTE is a popular technique used for balancing imbalanced datasets in machine learning (Chawla et al., 2002). It entails generating synthetic samples of the minority by interpolating additional data points between existing minority class samples. Studies have shown that SMOTE can improve the performance of several classification algorithms, including decision trees, logistic regression, and random forest (Fernández-Delgado et al., 2014). The balanced dataset was divided into features and label and all four models were

applied to the dataset. Hyperparameters were tuned and the dataset was re-evaluated using the new combination of hyperparameters obtained. For each model, a performance metrics was obtained.

3.5 Evaluation Metrics

There are several evaluation metrics commonly used in machine learning. This report focuses on the ones listed below but prioritises accuracy:

Accuracy: calculates the model's percentage of true predictions.

Precision: The proportion of true positive predictions made by the model over all positive predictions made by the model.

Recall: The proportion of true positive predictions in the dataset over all actual positive cases.

F1 score: it is a balanced measurement of both the precision and the recall.

Confusion matrix: It shows how many correct and wrong predictions the model made when compared to the actual results.

SHapley Additive exPlanations (SHAP): is a method for explaining individual predictions made by machine learning models.

4.0 RESULTS AND DISCUSSION

4.1 Experiment 1

Table 2 summarises the performance metrics for each model used. The models clearly achieved comparable results, but further investigation was done using the confusion matrix. The confusion matrix revealed a bias towards class '0', which could be due to the imbalance in the dataset. Class '0' accounted for 75% of the dataset, whereas class '1' accounted for only around 25%. Hence the need to balance the target value and conduct another experiment.

Algorithms	Accuracy	Precision (Macro weighted)	Recall (Macro weighted)	F1-score (Macro weighted)
LR	0.84	0.80	0.74	0.77
KNN	0.83	0.79	0.72	0.72
RF	0.85	0.84	0.75	0.78
DT	0.85	0.82	0.76	0.78

Table 2. Performance analysis on Imbalanced Dataset

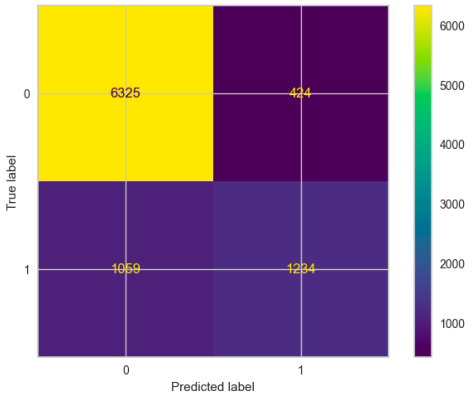


Fig 22. Confusion matrix for LR

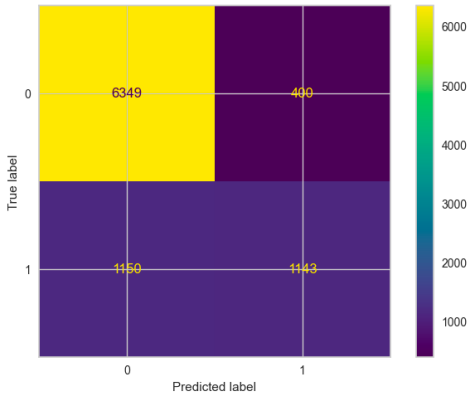


Fig 23. Confusion matrix for KNN

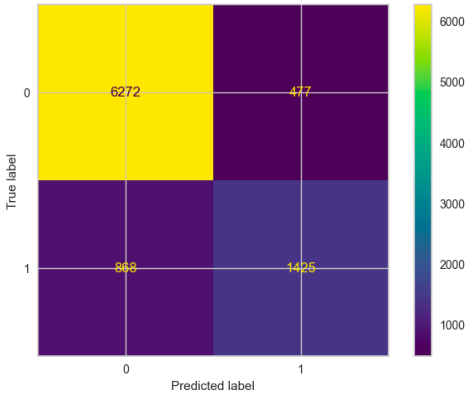


Fig 24. Confusion matrix for RF

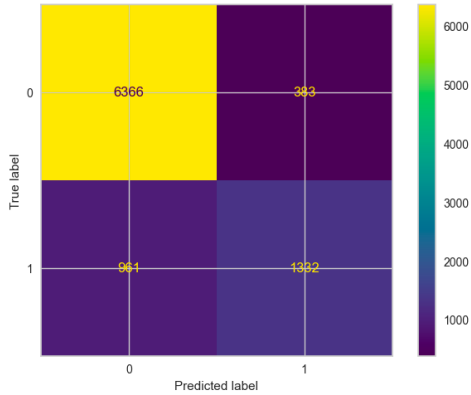


Fig 25. Confusion matrix for DT

4.2 Experiment 2

Table 3 below summarizes the performance metrics of each applied model. It can be seen that Random Forest algorithm outperformed all the other algorithms generally. This can be attributed to its ability to combine multiple decision trees that can capture different features of the data, resulting in a more accurate and resilient model.

Algorithms	Accuracy	Precision (Macro weighted)	Recall (Macro weighted)	F1-score (Macro weighted)
LR	0.81	0.81	0.81	0.81
KNN	0.84	0.84	0.84	0.84

RF	0.87	0.87	0.87	0.87
DT	0.83	0.84	0.83	0.83

Table 3. Performance analysis on Balanced Dataset

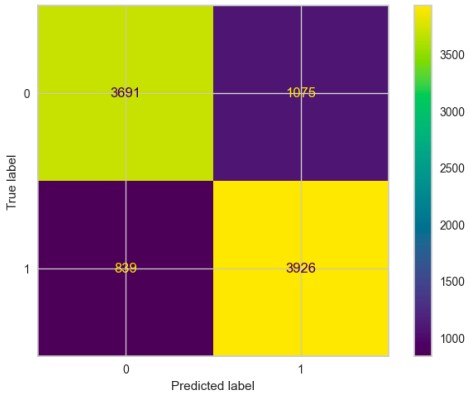


Fig 26. LR - Confusion matrix

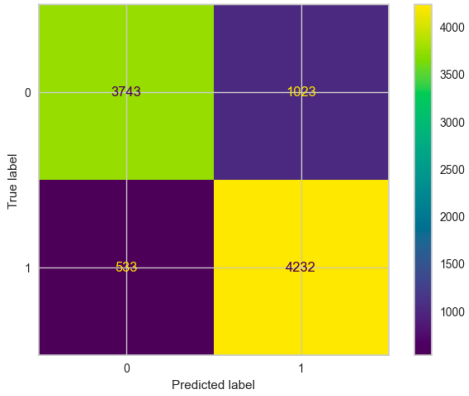


Fig 27. KNN - Confusion matrix

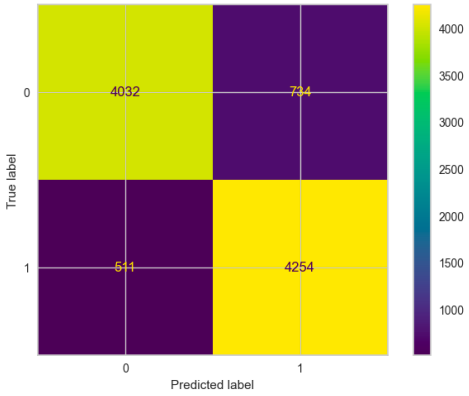


Fig 28. RF - Confusion matrix

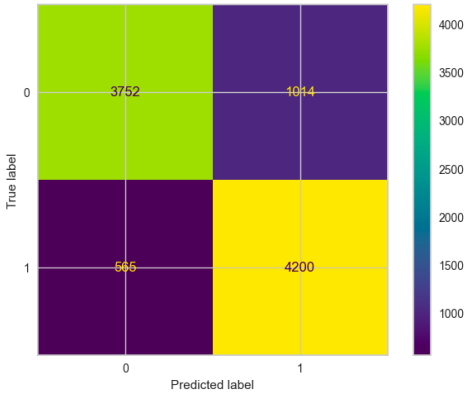


Fig 29. DT - Confusion matrix

K-fold cross-validation with 10 splits was performed for the best performing model (Random Forest) during the training stage. Evaluation metrics were collected using a weighted average, and the cross-validation scores for Random Forest are shown in Figure 28. The highest mean score of 87% was achieved at Kfold of 5. This was done to ensure the model did not overfit.

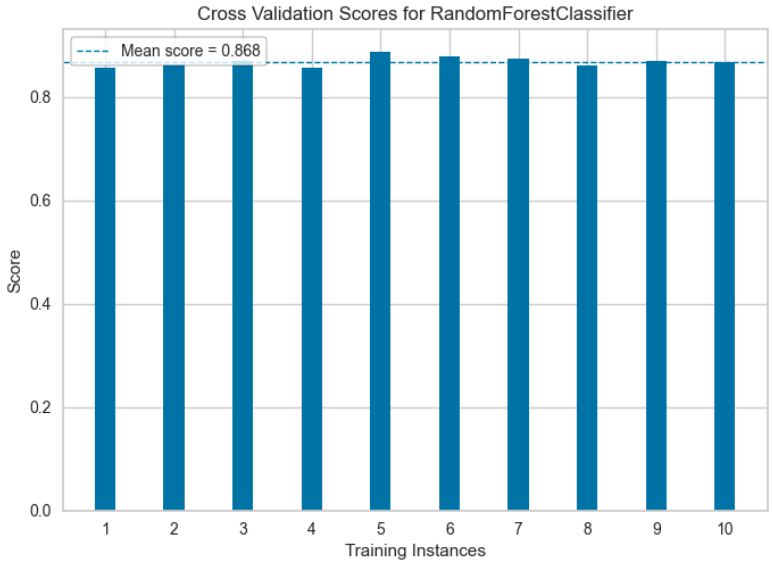


Fig 30. Cross validation Scores for Random Forest

K-fold cross-validation with 10 n_splits was performed for all the models during the training phase, and the evaluation metrics were collected using weighted average. Figure 29 below shows a comparison plot of the best weighted average for both experiment 1 and 2.

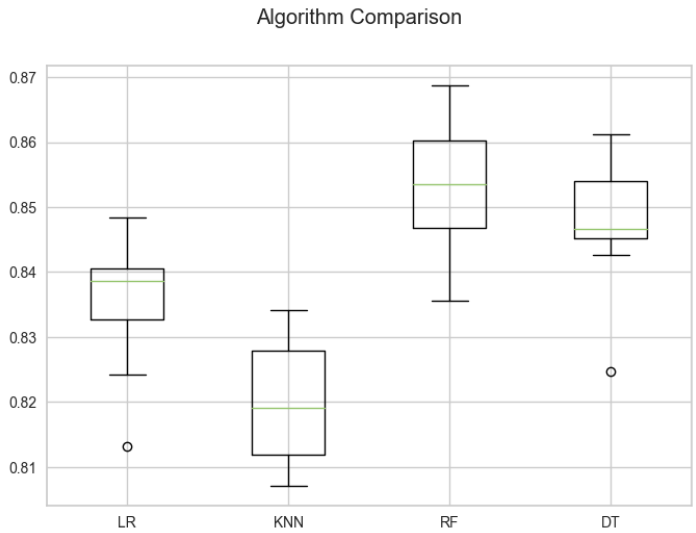


Fig 31. Algorithm comparism

4.3 SHAP

Fig 30. Shows a global interpretation of the Random Forest model using SHAP with capital-gain, marital-status, hours-per-week, and education-num having the highest contribution to the model's prediction.

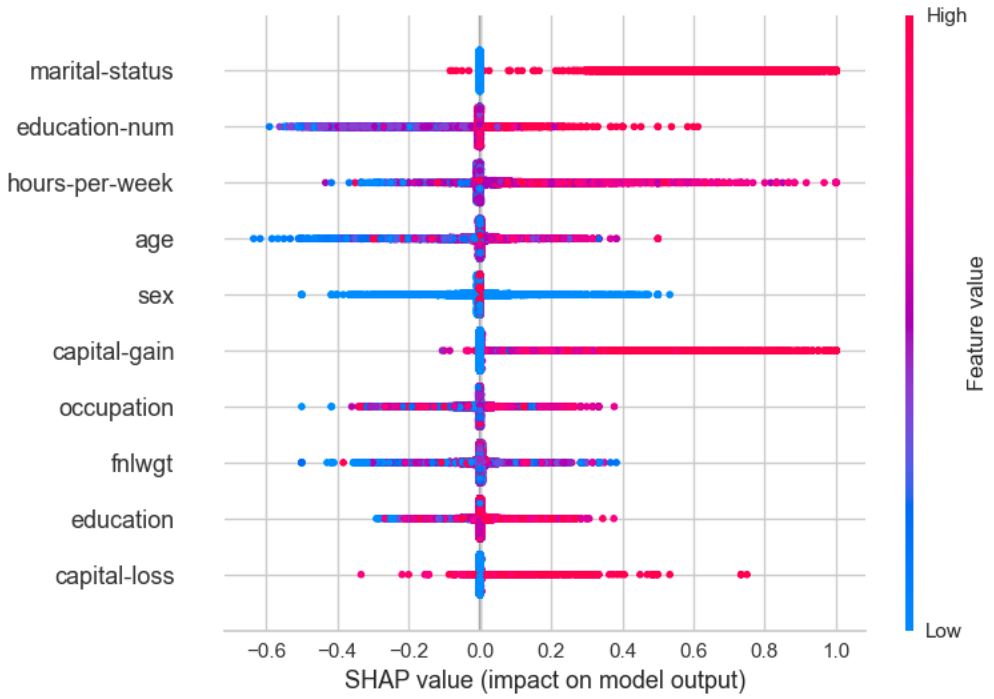


Fig 32. SHAP explanation of RF prediction

5.0 CONCLUSION AND FUTURE WORK

The purpose of this report was to use various machine learning models to perform salary dataset classification predictions and determine the best algorithm, as well as explain the reasons for its superior performance. After balancing the dataset, the Random Forest algorithm achieved an accuracy of 87%, making it the top-performing algorithm. Its success can be attributed to its ability to combine multiple decision trees, each of which can capture different aspects of the data, resulting in a more accurate and resilient model. Additionally, as an ensemble method, it aggregates multiple models to make predictions, which improves generalization and robustness while reducing overfitting risks. The most important features in predicting salary classification were capital-gain, marital-status, hours-per-week, and education-num. The work presented in this report is not perfect, and there is room for further exploration to find the best model for predicting salary. Some possible avenues for future research could include:

• Exploring advanced ML models like XGBoost, LightGBM, etc.

• Investigating ML techniques such as Voting, Stacking, etc.

REFERENCES

1. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
2. Cutler, A., Cutler, D. R., & Stevens, J. R. (2007). Random forests. Wiley interdisciplinary reviews: computational statistics, 1(3), 316-320.
3. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. Springer.
4. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.
5. Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? Journal of machine learning research, 15(1), 3133-3181.
6. Rokach, L., & Maimon, O. (2006). Decision Trees. In *Springer eBooks* (pp. 165–192). https://doi.org/10.1007/0-387-25465-x_9.
7. Park, H. (2013). An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain. *Journal of Korean Academy of Nursing*, *43*(2), 154. https://doi.org/10.4040/jkan.2013.43.2.154.
8. *Nearest Neighbors*. (n.d.). Scikit-learn. https://scikit-learn.org/stable/modules/neighbors.html.
9. Straw, I., & Wu, H. (2022c). Investigating for bias in healthcare algorithms: a sex-stratified analysis of supervised machine learning models in liver disease prediction. *BMJ Health & Care Informatics*, *29*(1), e100457. https://doi.org/10.1136/bmjhci-2021-100457.
10. *Welcome to the SHAP documentation — SHAP latest documentation*. (n.d.). https://shap.readthedocs.io/en/latest/index.html