BRAC UNIVERSITY

Inspiring Excellence

CSE422: Artificial Intelligence

Project Name: Classification-based Salary Allocation using a Multi-Model Machine Learning Approach

Submitted by: Group 04, Section 03

MD Faisal Iftekhar (22299116)
K M Ramim Azim (22299011)

Submitted to:

Farhan Faruk
Asif Hasan Choudhury

Lecturer, BRAC University

Table of Contents

Introduction

Understanding factors that lead to economic activity are vital for making informed fiscal and monetary decisions and policies. The following dataset outlines how various social and economic factors such as occupation, location and education can influence the annual income of a person. Our model aims to predict through classification whether a person given their specific factors makes more or less than a certain amount of income annually. Moreover, the factors which are definitive or have more influence on the person's income are also studied. The relationships between these factors and their correlations are also analysed. This classification based approach will allow governments to make relevant economic policies for taxation, and also markets to adopt their strategies to appeal to the consumer.

Dataset Description

Source:
https://drive.google.com/file/d/1fa47DUtjxh2lOex_aN5yhawVHh87mLyJ/view

The dataset details adult annual income, which is based on various features such as Employment, Native Country, Education and Marital Status, amongst many others. The dataset contains 48842 data points and has 15 features. The target variable here is either greater than 50000, or less than or equal to 50000, which makes it a classification problem, or more specifically, a binary classification problem. It contains both categorical and quantitative features and has null values in multiple columns, which are later adjusted for during preprocessing. The dataset in its raw format was ineligible for training and testing purposes, however, upon scaling, imputing and encoding of categorical data to quantitative data, the data was fed to our learning models and the accuracy of the models then compared. A summary is attached below.

```
Number of data points: 48842

Number of features: 15

Dataset loaded. First 5 rows:
   Age         Workclass  Final Weight  Education  Education Number of Years  \
0   39         State-gov         77516  Bachelors                         13
1   50  Self-emp-not-inc         83311  Bachelors                         13
2   38           Private        215646    HS-grad                          9
3   53           Private        234721       11th                          7
4   28           Private        338409  Bachelors                         13

       Marital-status         Occupation   Relationship   Race     Sex  \
0       Never-married       Adm-clerical  Not-in-family  White    Male
1  Married-civ-spouse    Exec-managerial        Husband  White    Male
2            Divorced  Handlers-cleaners  Not-in-family  White    Male
3  Married-civ-spouse  Handlers-cleaners        Husband  Black    Male
4  Married-civ-spouse     Prof-specialty           Wife  Black  Female

   Capital-gain  Capital-loss  Hours-per-week Native-country target
0          2174             0              40  United-States  <=50K
1             0             0              13  United-States  <=50K
2             0             0              40  United-States  <=50K
3             0             0              40  United-States  <=50K
4             0             0              40           Cuba  <=50K
```

```
DataFrame Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Age                        48842 non-null  int64
 1   Workclass                  48842 non-null  object
 2   Final Weight               48842 non-null  int64
 3   Education                  48842 non-null  object
 4   Education Number of Years  48842 non-null  int64
 5   Marital-status             48842 non-null  object
 6   Occupation                 48842 non-null  object
 7   Relationship               48842 non-null  object
 8   Race                       48842 non-null  object
 9   Sex                        48842 non-null  object
 10  Capital-gain               48842 non-null  int64
 11  Capital-loss               48842 non-null  int64
 12  Hours-per-week             48842 non-null  int64
 13  Native-country             48842 non-null  object
 14  target                     48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

```
Summary Statistics for Numerical Features (from documentation list):
               Age  Final Weight  Education Number of Years  Capital-gain  \
count  48842.000000  4.884200e+04               48842.000000  48842.000000
mean      38.643585  1.896641e+05                  10.078089   1079.067626
std       13.710510  1.056040e+05                   2.570973   7452.019058
min       17.000000  1.228500e+04                   1.000000      0.000000
25%       28.000000  1.175505e+05                   9.000000      0.000000
50%       37.000000  1.781445e+05                  10.000000      0.000000
75%       48.000000  2.376420e+05                  12.000000      0.000000
max       90.000000  1.490400e+06                  16.000000  99999.000000

       Capital-loss  Hours-per-week
count  48842.000000    48842.000000
mean      87.502314       40.422382
std      403.004552       12.391444
min        0.000000        1.000000
25%        0.000000       40.000000
50%        0.000000       40.000000
75%        0.000000       45.000000
max     4356.000000       99.000000

Target Variable Distribution:
target
<=50K    0.760718
>50K     0.239282
Name: proportion, dtype: float64
```

	Age	Workclass	Final Weight	Education Number of Years	Marital-status	Occupation	Relationship	Capital-gain	Capital-loss	Hours-per-week	Native-country	target
count	48842.000000	48842	4.884200e+04	48842.000000	48842	48842	48842	48842.000000	48842.000000	48842.000000	48842	48842
unique	NaN	8	NaN	NaN	7	14	6	NaN	NaN	NaN	41	2
top	NaN	Private	NaN	NaN	Married-civ-spouse	Prof-specialty	Husband	NaN	NaN	NaN	United-States	<=50K
freq	NaN	36705	NaN	NaN	22379	8981	19716	NaN	NaN	NaN	44689	37155
mean	38.643585	NaN	1.896641e+05	10.078089	NaN	NaN	NaN	1079.067626	87.502314	40.422382	NaN	NaN
std	13.710510	NaN	1.056040e+05	2.570973	NaN	NaN	NaN	7452.019058	403.004552	12.391444	NaN	NaN
min	17.000000	NaN	1.228500e+04	1.000000	NaN	NaN	NaN	0.000000	0.000000	1.000000	NaN	NaN
25%	28.000000	NaN	1.175505e+05	9.000000	NaN	NaN	NaN	0.000000	0.000000	40.000000	NaN	NaN
50%	37.000000	NaN	1.781445e+05	10.000000	NaN	NaN	NaN	0.000000	0.000000	40.000000	NaN	NaN
75%	48.000000	NaN	2.376420e+05	12.000000	NaN	NaN	NaN	0.000000	0.000000	45.000000	NaN	NaN
max	90.000000	NaN	1.490400e+06	16.000000	NaN	NaN	NaN	99999.000000	4356.000000	99.000000	NaN	NaN

Check for Null and Duplicate Values

```python
data.isnull().sum()
```

	0
Age	0
Workclass	0
Final Weight	0
Education Number of Years	0
Marital-status	0
Occupation	0
Relationship	0
Capital-gain	0
Capital-loss	0
Hours-per-week	0
Native-country	0
target	0

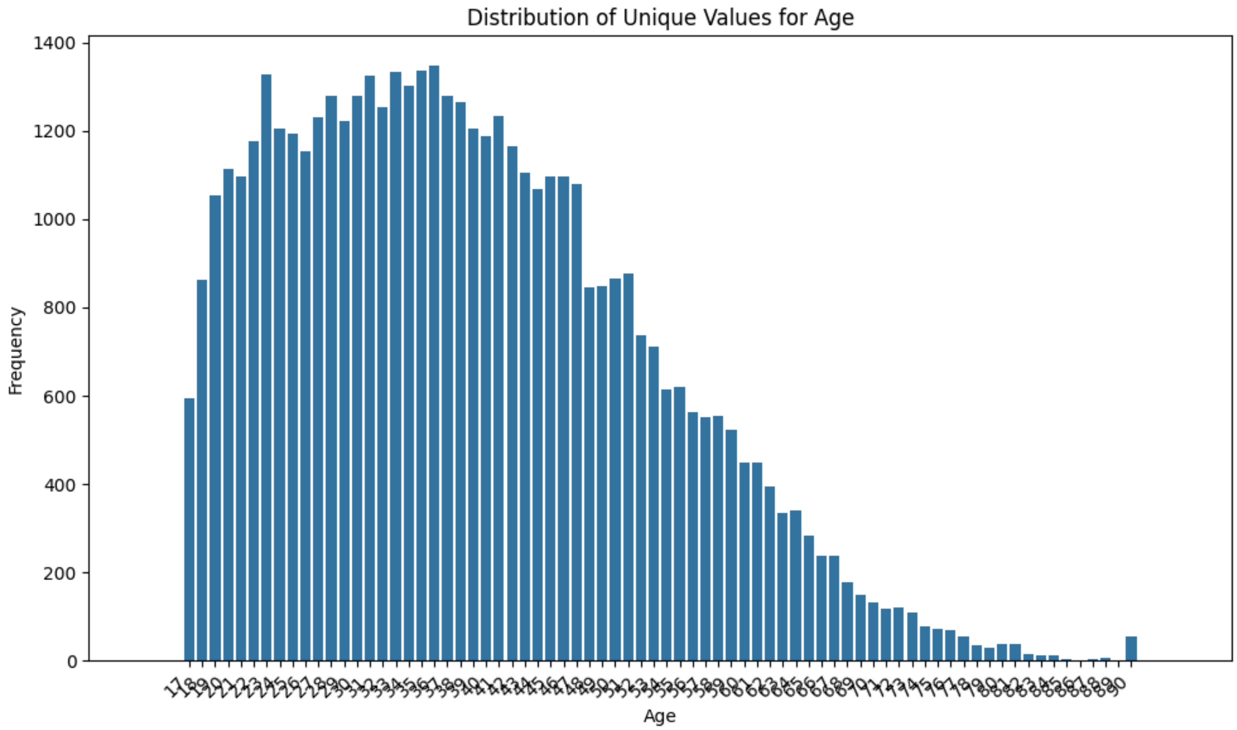
dtype: int64

```python
data.nunique()
```

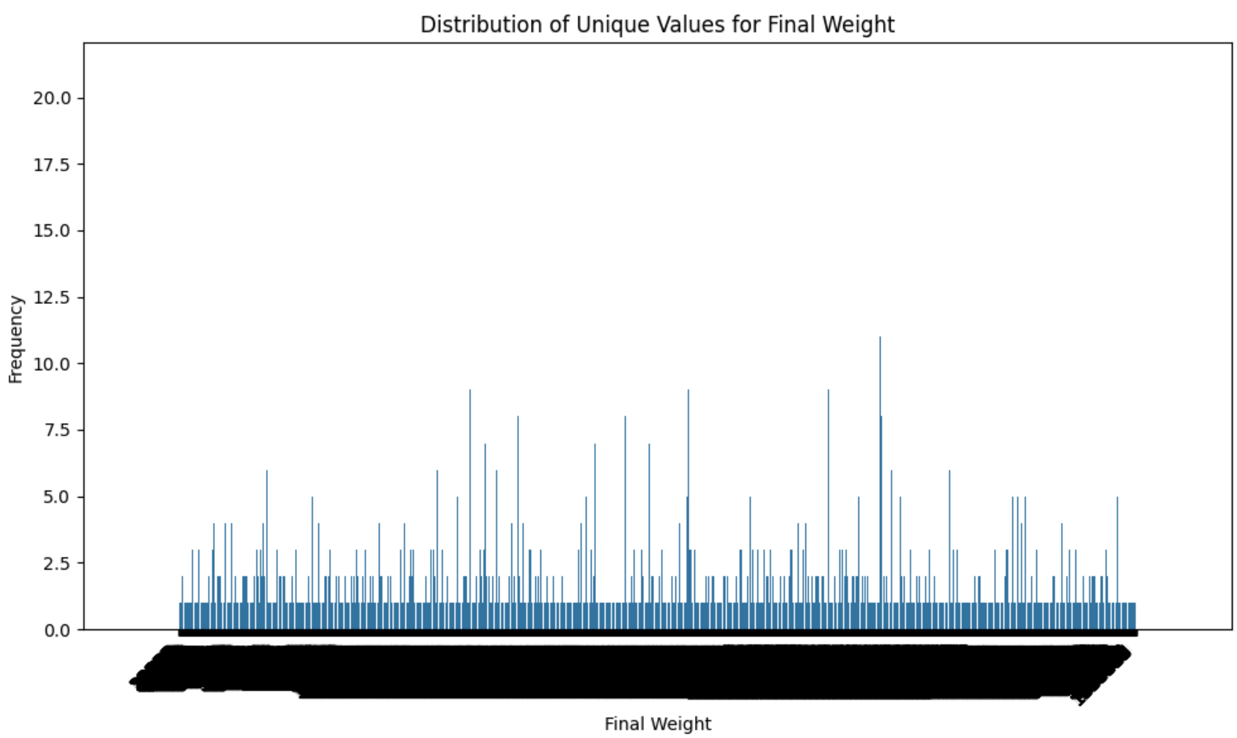
	0
Age	74
Workclass	9
Final Weight	28523
Education	16
Education Number of Years	16
Marital-status	7
Occupation	15
Relationship	6
Race	5
Sex	2
Capital-gain	123
Capital-loss	99
Hours-per-week	96
Native-country	42
target	2

dtype: int64

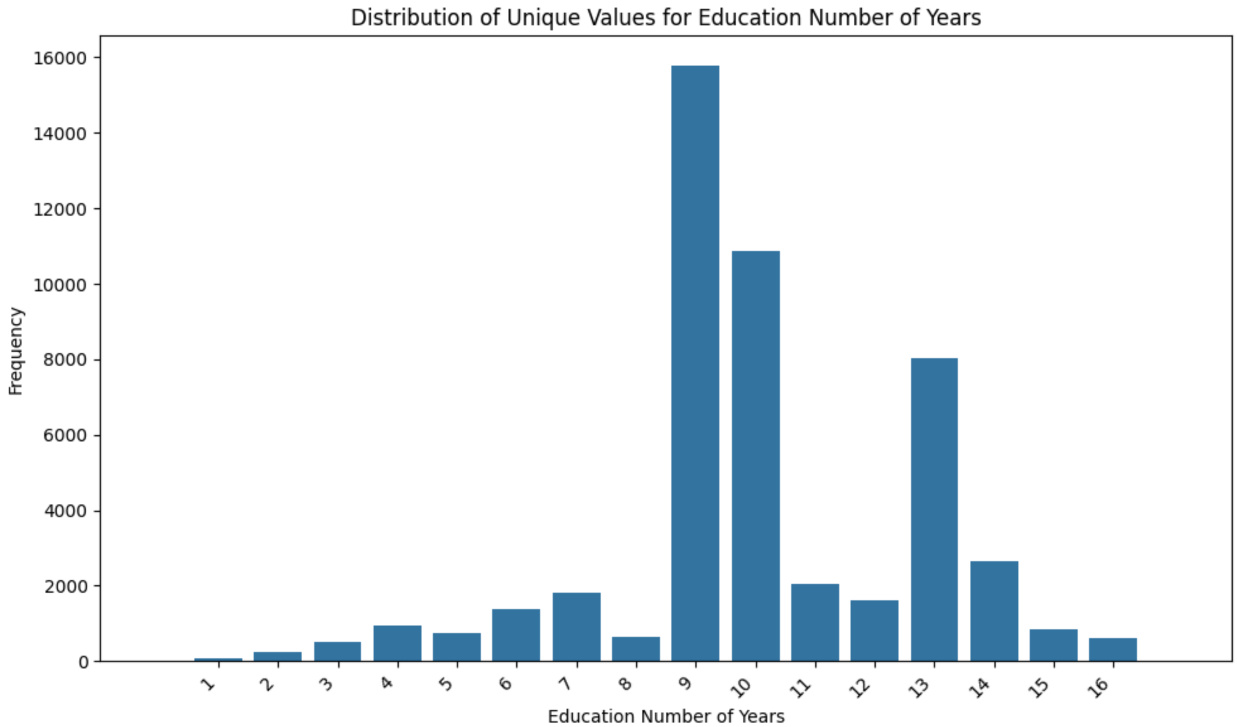
Fig: Summary of Dataset and Features

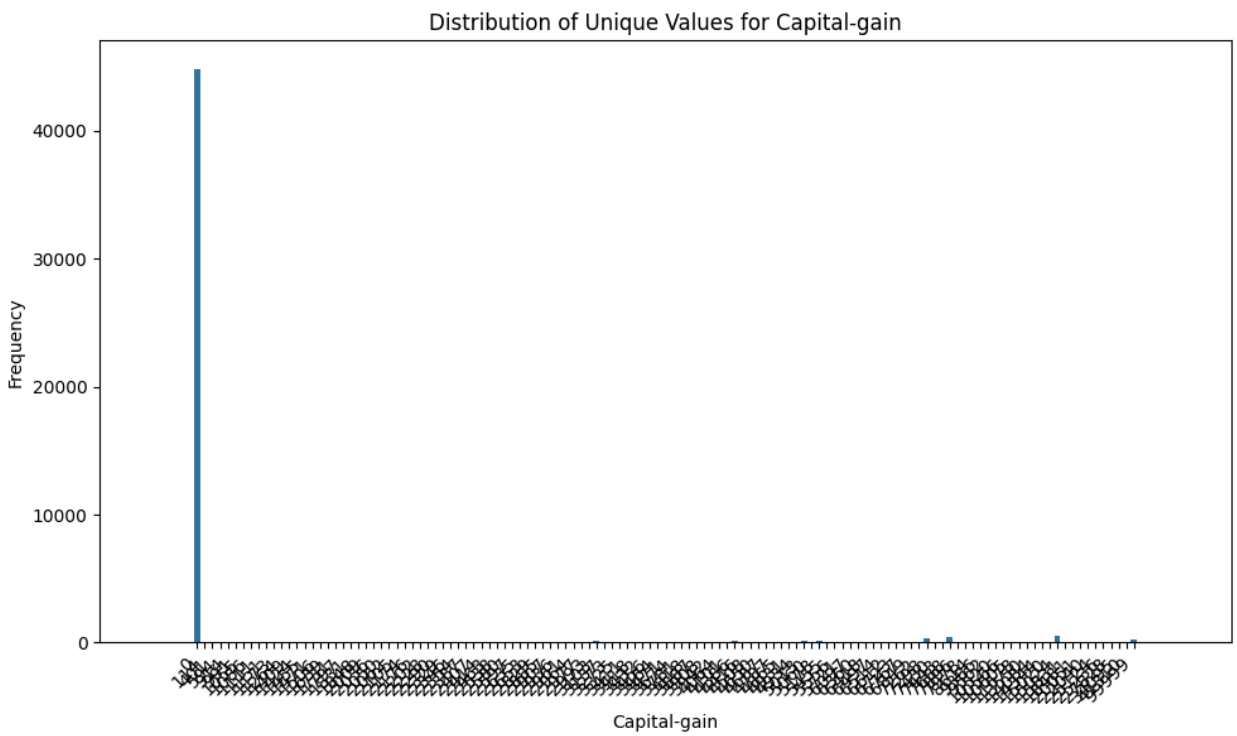


Distribution of Unique Values for Age
Frequency
Age



Distribution of Unique Values for Final Weight
Frequency
Final Weight


Distribution of Unique Values for Education Number of Years
Frequency
0
2000
4000
6000
8000
10000
12000
14000
16000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
Education Number of Years


Distribution of Unique Values for Capital-gain
Frequency
0
10000
20000
30000
40000
Capital-gain

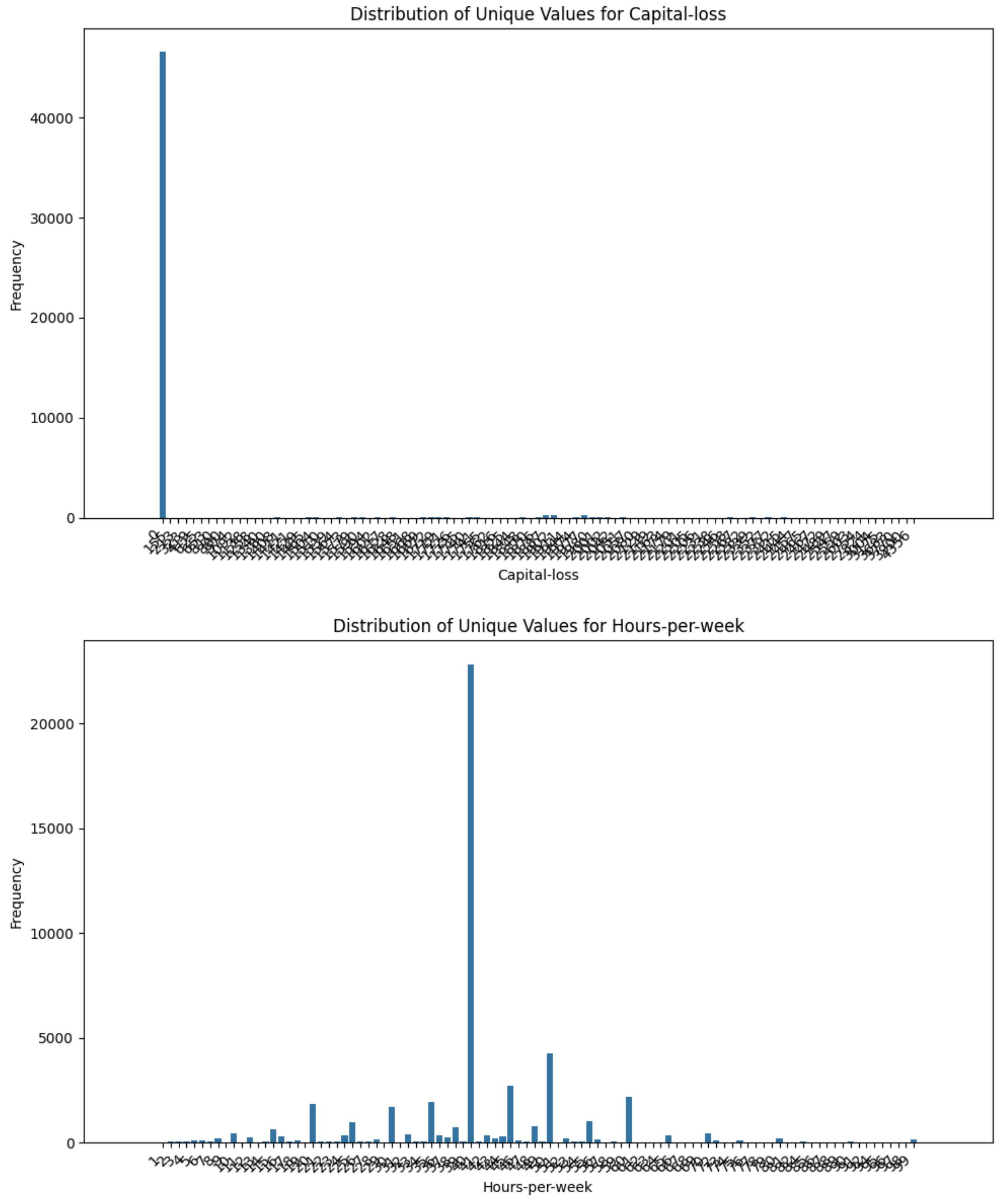


Fig: Distribution of some of the features of the dataset

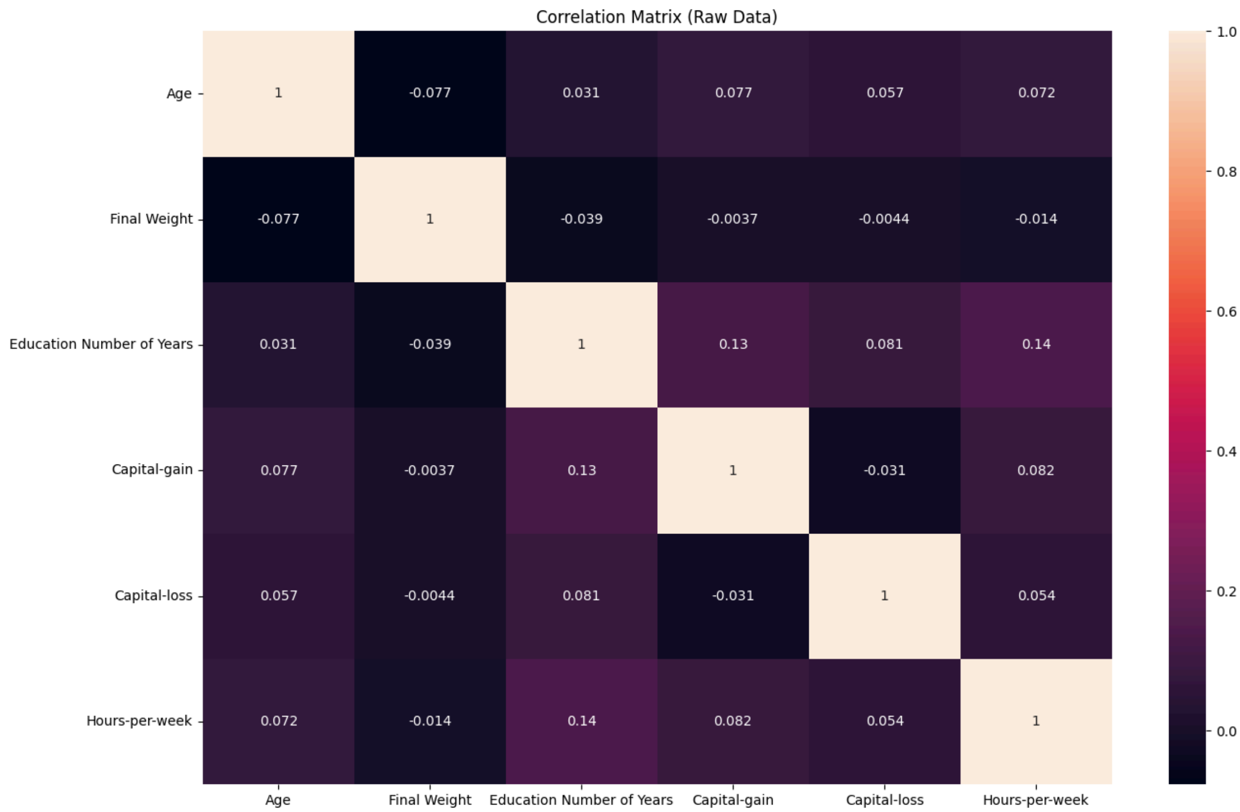


Fig: Heatmap

From the Correlation Matrix above, we can notice the correlation between different features of the dataset. Here, each row variable is associated with each of the column variables. The numbers of the cells represent their correlation. Any value greater than 0 means that the two variables are positively correlated, meaning one increasing or decreasing leads the other to also increase or decrease.Any value less than 0 means that the two variables are negatively correlated, meaning one increasing or decreasing leads the other to also decrease or increase. Zero means that the variables are not correlated.

1 would mean a perfect positive correlation and -1 would mean a perfect negative correlation. We can notice that the diagonal values are 1, because each variable is perfectly correlated with itself.

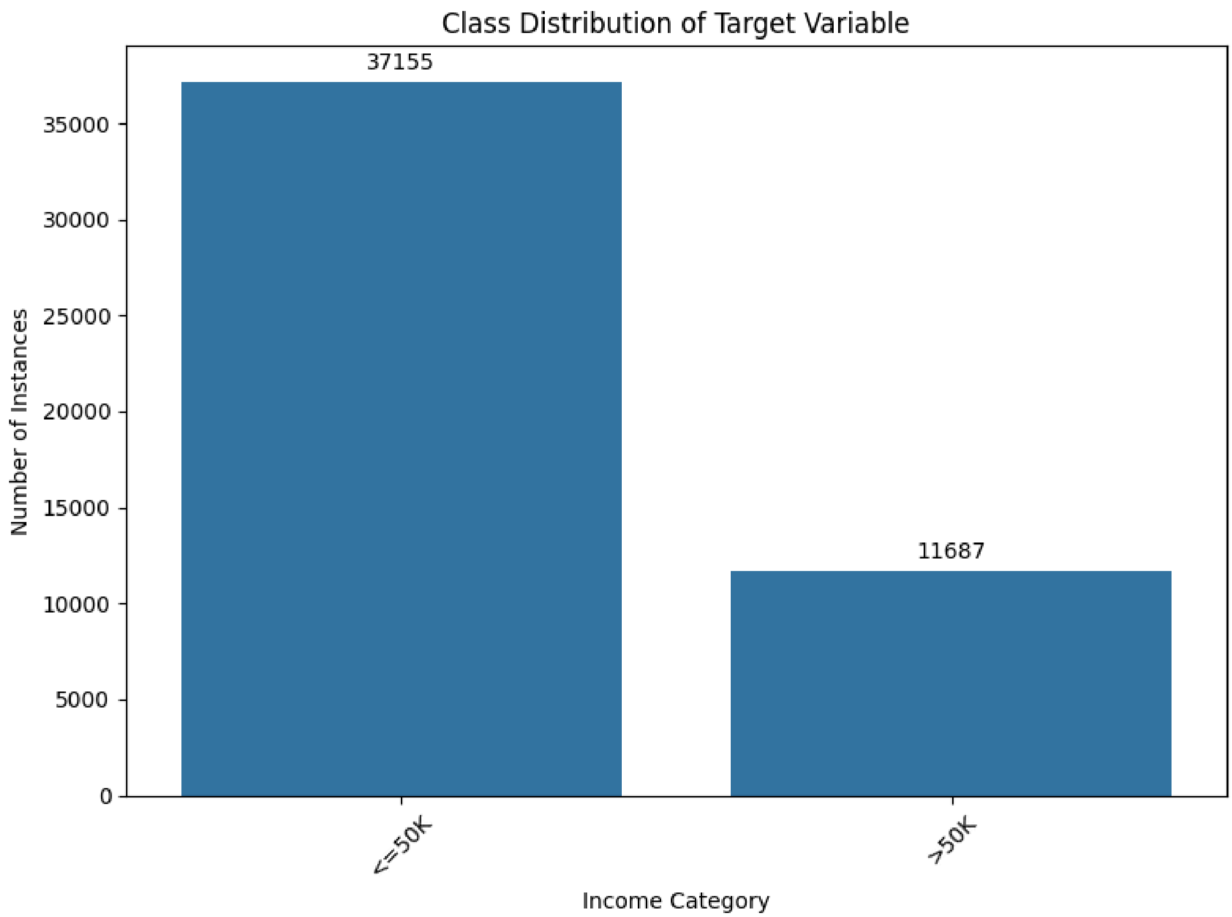


Fig: Class distribution of Target Variable

The output feature is not balanced, which is very natural for a dataset based on income.

Dataset Pre-processing

Dataset preprocessing involved replacing erroneous values in several columns and imputing with the mode values for the necessary columns.

Problem 1: Many columns in the dataset had null values, denoted by "?". These null values made it difficult to feed the dataset to the model.

Solution: The null values were imputed with the mode values. The reason for choosing mode was that it was the most frequent value in the column.

Problem 2: Columns such as ‘Education’, ‘Race’ and ‘Sex’ are either redundant or could have introduced racial and gender bias, which would challenge the ethics of Machine Learning.

Solution: They were dropped from the dataset.

Problem 3: There were many categorical features such as ‘Marital Status’, ‘Occupation’ and ‘Native Country’ that did not have numerical values that could be used in the model.

Solution: They were encoded using One-Hot encoding to ensure there were binary columns for all the values of the respective columns.

Problem 4: The quantitative features have different ranges of values, which makes it difficult to optimize.

Solution: They were normalized using standard scaling.

Dataset Splitting

The dataset was split into 70% as training data and 30% for testing data using the stratified method. This was done to ensure that the model had enough data to learn and also enough data to test for accuracy. The split was stratified to ensure the same distribution across the subsets. For Neural Network, an additional 10% data was split from the training data as the cross validation data for evaluating in between epochs.

Model Training and Testing

Three models were used in our project:

a) Logistic Regression
b) Decision Tree
c) Neural Network

Logistic Regression

```
--- Performance Metrics for: Logistic Regression ---
Accuracy: 0.8506
AUC Score: 0.9032
Classification Report:
              precision    recall  f1-score   support

       <=50K       0.88      0.93      0.90     11147
        >50K       0.73      0.59      0.65      3506

    accuracy                           0.85     14653
   macro avg       0.81      0.76      0.78     14653
weighted avg       0.84      0.85      0.84     14653

Confusion Matrix:
[[10399   748]
 [ 1441  2065]]
```

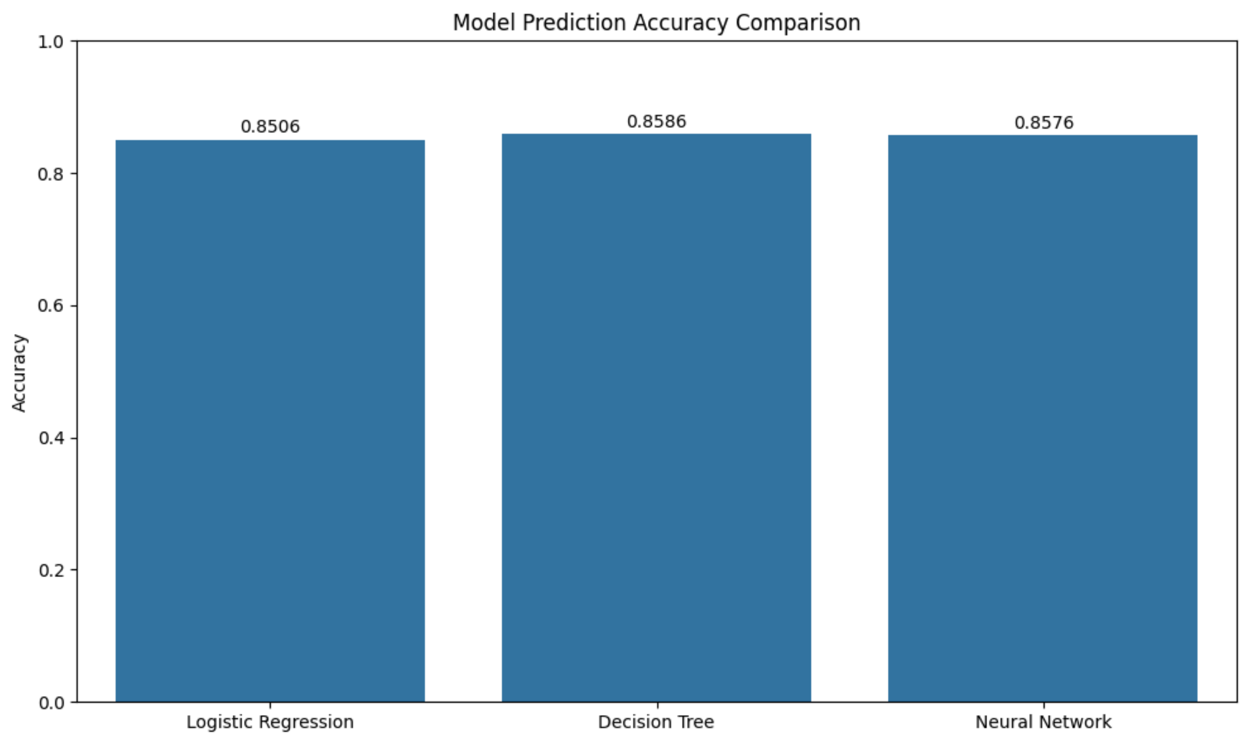
Decision Tree

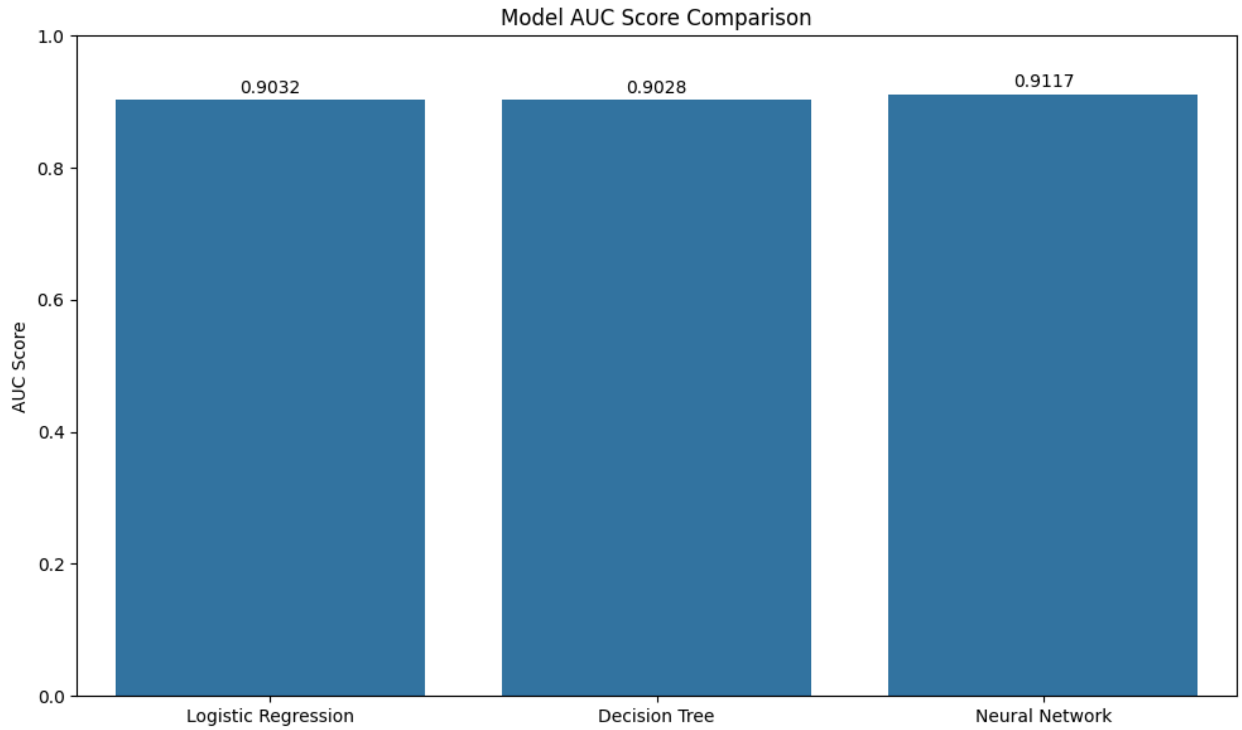
```
--- Performance Metrics for: Decision Tree ---
Accuracy: 0.8586
AUC Score: 0.9028
Classification Report:
              precision    recall  f1-score   support

       <=50K       0.88      0.95      0.91     11147
        >50K       0.77      0.58      0.66      3506

    accuracy                           0.86     14653
   macro avg       0.82      0.76      0.79     14653
weighted avg       0.85      0.86      0.85     14653

Confusion Matrix:
[[10540   607]
 [ 1465  2041]]
```

Neural Network

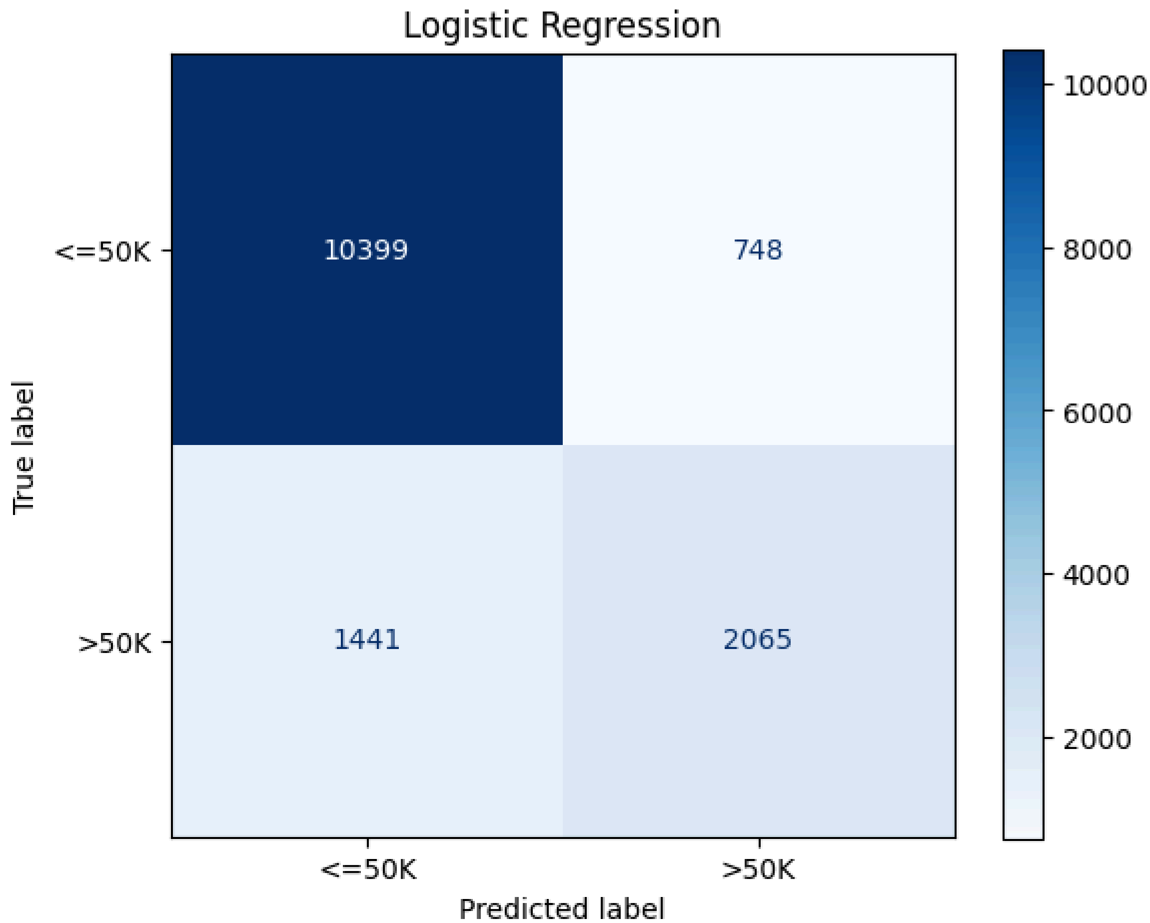
```
--- Performance Metrics for: Neural Network ---
Accuracy: 0.8576
AUC Score: 0.9117
Classification Report:
              precision    recall  f1-score   support

       <=50K       0.89      0.92      0.91     11147
        >50K       0.73      0.65      0.69      3506

    accuracy                           0.86     14653
   macro avg       0.81      0.79      0.80     14653
weighted avg       0.85      0.86      0.85     14653

Confusion Matrix:
[[10288   859]
 [ 1228  2278]]
```

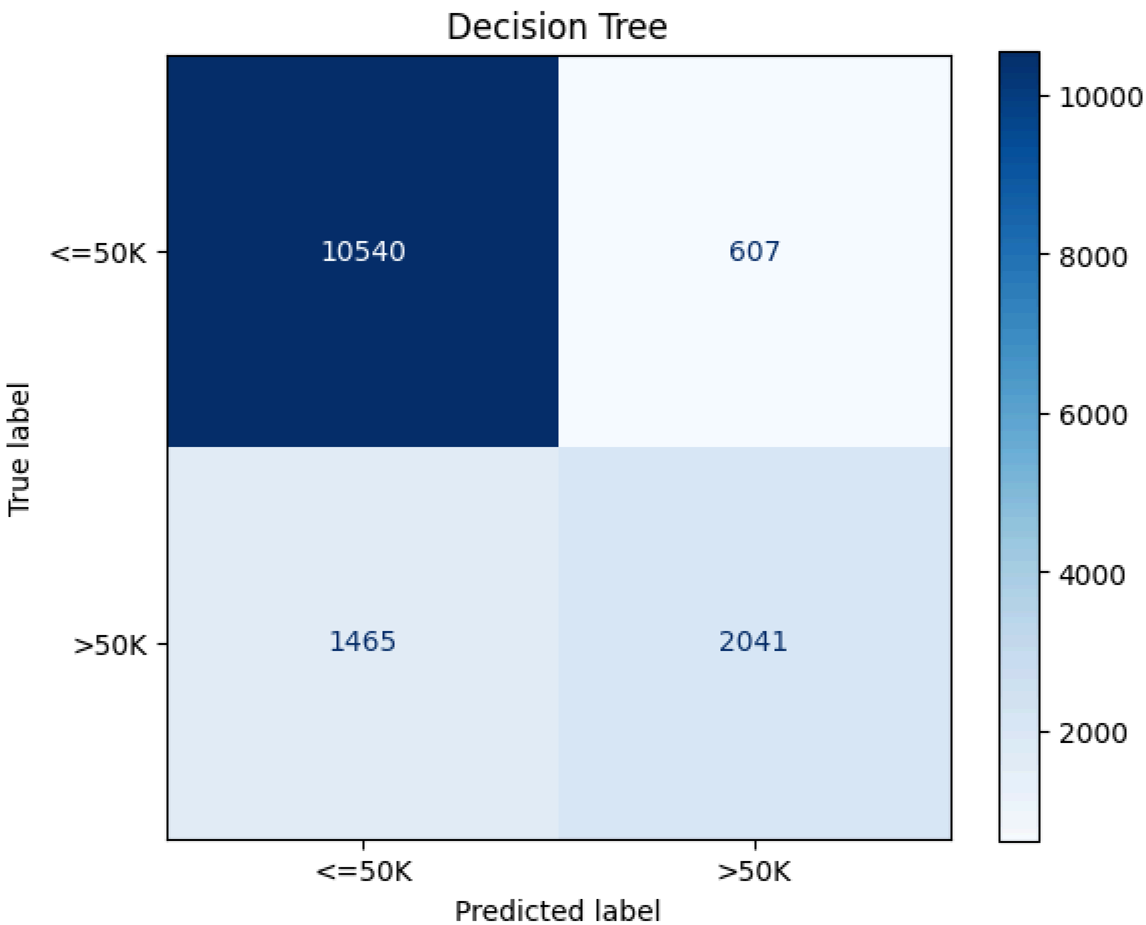
Model Selection and Comparison Analysis

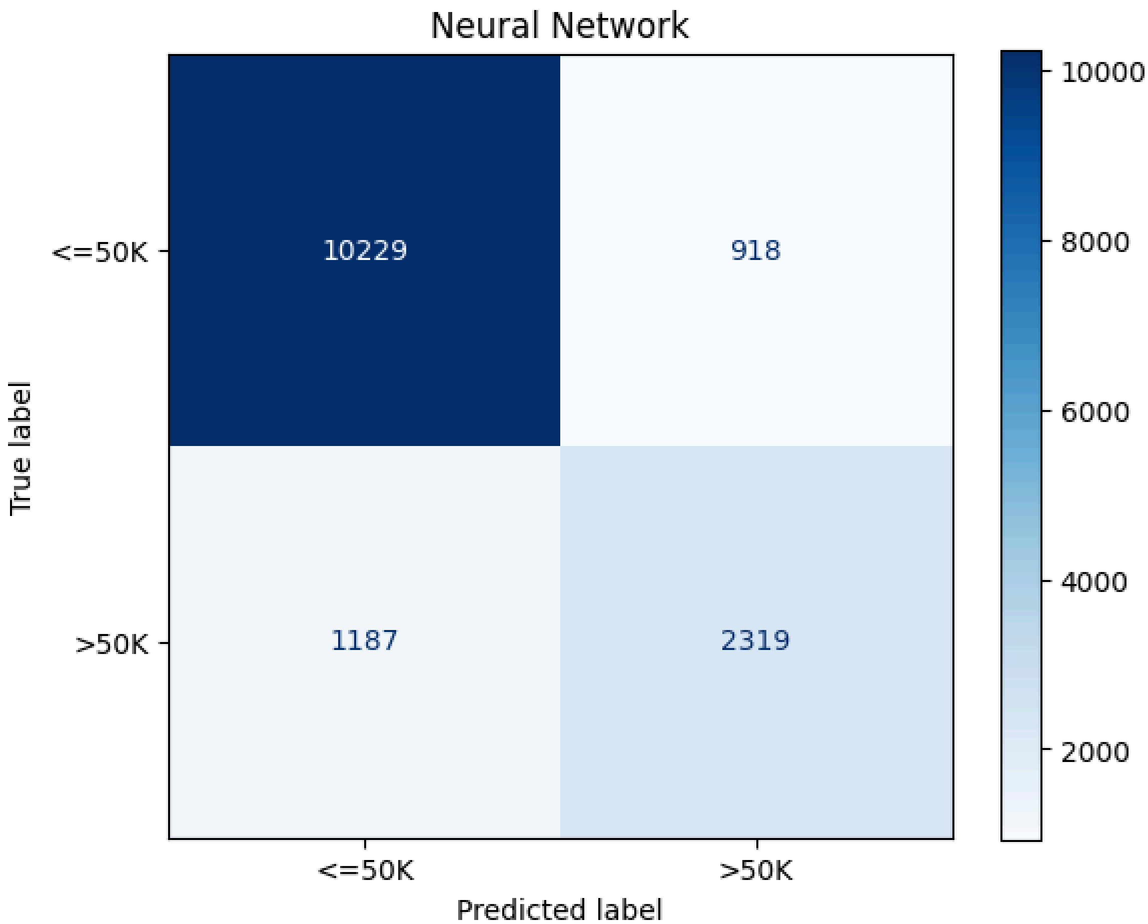


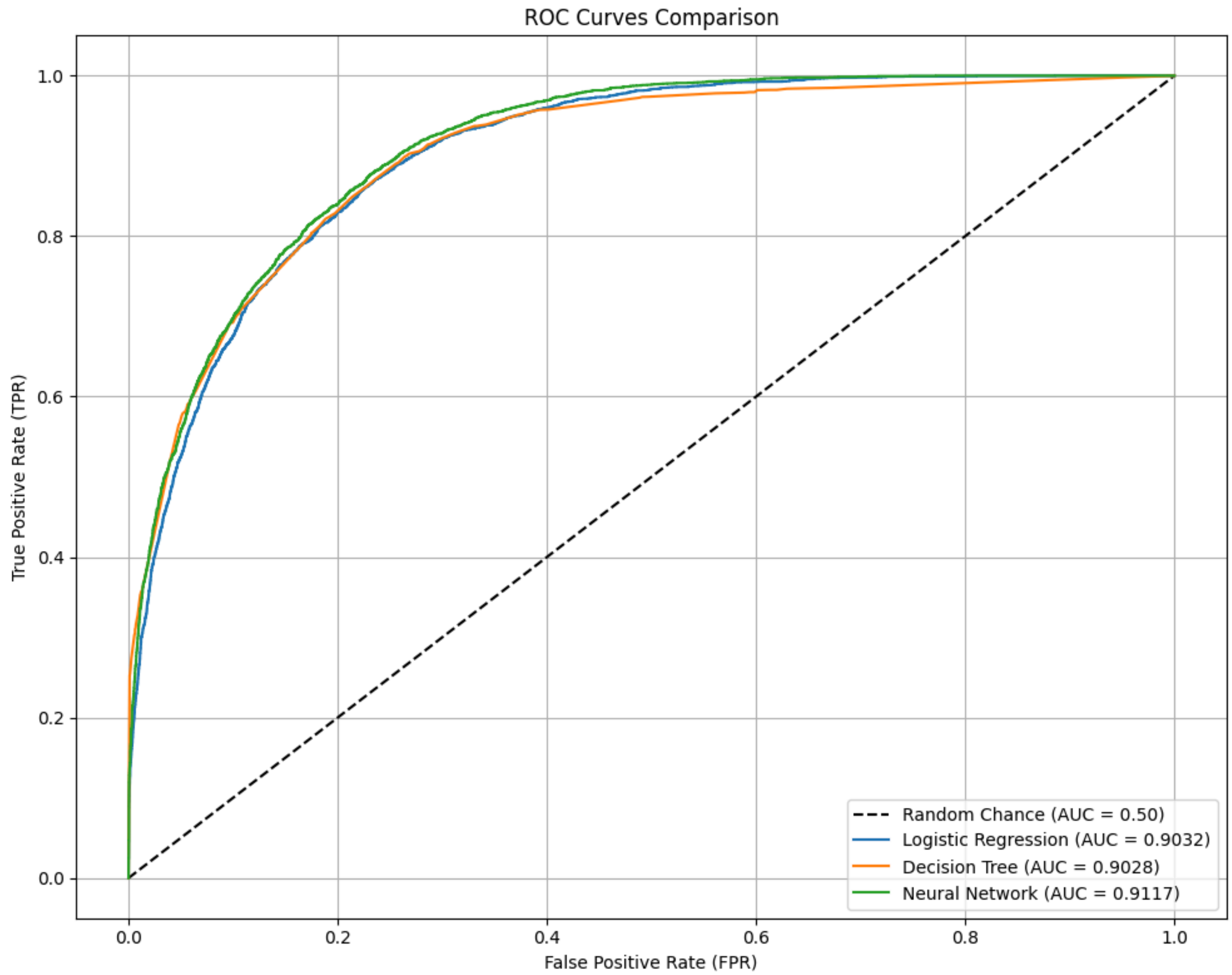

Model AUC Score Comparison
AUC Score
1.0
0.8
0.6
0.4
0.2
0.0
0.9032
0.9028
0.9117
Logistic Regression
Decision Tree
Neural Network

Confusion Matrices for Different Models




Decision Tree
<=50K
>50K
True label
10540
607
1465
2041
<=50K
>50K
Predicted label
10000
8000
6000
4000
2000


Neural Network
<=50K
>50K
True label
10229
918
1187
2319
<=50K
>50K
Predicted label
10000
8000
6000
4000
2000



As it can be seen, all three models score very closely in terms of accuracy and AUC score. The Confusion Matrix for all three models are nearly identical. However, amongst them Decision Tree gives a higher Accuracy, while the Neural Network slightly outperforms in the AUC score. Logistic regression falls behind both, however with a very small difference.

Conclusion

	Model	Accuracy	AUC	Precision (>50K)	Recall (>50K)	F1-score (>50K)
0	Logistic Regression	0.850611	0.903215	0.734092	0.588990	0.653584
1	Decision Tree	0.858596	0.902770	0.770770	0.582145	0.663308
2	Neural Network	0.857572	0.911699	0.726172	0.649743	0.685835

Since all three models gave very very similar scores, this may speak to the simplicity of the dataset and the target value. Moreover, classification models used to treat this dataset may not be the best method, as regression can be explored to estimate the salary directly. However, this still provides a decent model to estimate the annual income of a person which can help the government to set policies for taxation for example and many uses. Challenges regarding this dataset laid in the structuring and preprocessing of the dataset, including fixing null values and deciding on which features to drop from the dataset. Overall, while all models have very similar scores on Accuracy and AUC, we believe that Neural Network is the best model out of the three, given its F1 Score, which is significantly higher than the others, which is also observed in its Recall score. Out of five measures, Neural Networks outform the rest in three measures. Since there is no baseline to compare the model to, it is not possible to comment on how well the models performed.