INCOME PREDICTION USING MACHINE LEARNING TECHNIQUES

Group Members

- Rehul Ghag
- Siddhant Dube
- Soham Bhogale

Abstract

A deep knowledge of machine learning techniques is key when it comes to developing more efficient and effective approaches, and methods for predicting an output after training a set of data. The aim of this project is to analyze the adult dataset and classify the annual income of an individual by gathering the data and fitting the data into different machine learning models. We aim to predict whether an individual's income will be greater than $50,000 or less than $50,000 per year based on several attributes which are defined as the input. The results showed that several factors affect the accuracy for predicting the outcome of a dataset. For the project we will examine the US Adult Census dataset which is a repository of 48,842 entries extracted from the 1994 US Census database.
Data obtained from : https://archive.ics.uci.edu/ml/datasets/Adult

Introduction

This machine learning project examines the potentially significant effects that the census income data has on the yearly income of an individual. Census data is collected at regular intervals using various methodologies such as total counts, sample surveys, and administrative records. After it is collected or generated, census data is summarized to represent counts or estimates of groups of people for different geographic areas [1].

This project consists of two primary components, performing data cleaning techniques on the dataset which is the entire process of data preprocessing, and applying machine learning models on the cleaned data. The data is preprocessed by handling missing values and reconstructing the data into the most suitable format to ensure successful fitting of the data on the different machine learning models. By performing machine learning techniques on the dataset, we aim to successfully predict that an income exceeds the value of $50,000 annually or not. An individual's annual income results from several factors which are discussed in further detail in the data description section. Intuitively, it is influenced by the individual's education level, age, gender, occupation, and etc.

We aimed to assess the effectiveness of the machine learning models on the preprocessed dataset. A multi-step process was implemented to fit the data into the machine learning models, the machine learning models we aim to use for the dataset are, Logistic Regression, Support Vector Machine, and Neural Network.

It was concluded that Logistic Regression is the bit fit for the dataset hence it is the base model for our project and the other two models are compared to the Logistic Regression model. After the results from fitting the dataset into the models were obtained, they were studied to examine the performance across different machine learning models. The bias variance tradeoff property was also studied to determine if our model was under fitting or overfitting the data fitted into it [2].

Data description

The dataset was obtained from the University of California Irvine machine learning repository. This dataset which is named as the adult dataset was extracted by Barry Becker from the 1994 census database. A set of pre-conditions were applied to obtain the data to retrieve clean records. The data description is as follows:

Dataset Characteristics: Multivariate, the dataset consists of two or more variable quantities which are observed

Number of instances: 48842, the number of rows in the dataset

Area: Social, the domain of the dataset comes from a social background perspective

Attribute Characteristics: Categorical, Integer. The attributes of the dataset take the form of a categorical data and integer based data.

Number of attributes: 15, the number of columns in the dataset

Date Donated: 1996-05-01

Associated Tasks: Classification, the machine learning model to be applied on this dataset is the classification model for optimal results

Missing Values: Yes, the dataset consists of missing values which needs to be cleaned

Number of web hits: 2183850

Attribute Information

The attributes of the dataset are defined as the columns in the dataset. Below is an explanation of each attribute by first stating the type of the attribute and what it signifies.

- age: The age of an individual
    - Continuous variable with an integer value of above 0
- workclass: The employment status of an individual
    - Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
- fnlwgt: The number of people an entry represents
- education: The level of education of an individual
    - Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
- education_num: The level of education in numerical form
    - Continuous variable
- marital_status: Marital status of an individual
    - Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
- occupation: Occupation of an individual

- Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspect, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces

- relationship: What the relationship status of an individual is
  - Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
- race: Race of an individual ○ White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
- sex: Biological sex of an individual
  - Female, Male
- capital_gain: The capital gain of an individual
  - Continuous variable with an integer value of 0 or above
- capital_loss: The capital loss of an individual
  - Continuous variable with an integer value of 0 or above
- hours_per_week: The number of hours an individual has worked in a week
  - Continuous
- native_country: Country of origin for an individual
  - United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad Tobago, Peru, Hong, Holland-Netherlands
- income: whether or not an individual makes more than $50,000 annually.
  - <=50k, >50k

Methods (description of models you have used)

In this project we have used different types of classification algorithms to classify our output in order to minimize the error and improve the accuracy. Once the results are obtained we will then compare the results from the different classification algorithms to determine the most accurate classification technique for predicting the yearly income of an individual. For implementation, we will be dividing our data into two sets, training data and testing data. Eighty percent of the original data is divided into the training data and the remaining twenty percent is divided into the testing data, this is essential to ensure that enough data has been passed through the models for a better accuracy when the model is tested on the testing data. This will be done to train the model and validate the trained model against the test data to check our model. We will be using RMSE (Root mean squared error) and other testing algorithms to check the accuracy of our model.

Data Preprocessing

The data obtained from its original source consisted of several missing values which were handled by first locating all the instances where a question mark was present that suggested a missing value, this question mark was converted into a nan value. The reason for this was that it is efficient to detect all the nan values and drop the entire row where those values occur. The reason we decided to eliminate an entire row rather than replacing the missing value with the mean of that column was that we observed that after dropping all the rows consisting of missing values, we only got rid of 7 percent of the total rows which showed we are not losing too much of data, therefore the entire rows consisting of missing values were dropped from the dataset.
Our dataset consisted of several columns, some were numerical columns and some were categorical, therefore to ensure consistency of the data types in our dataset we encoded the categorical columns into numerical columns which converted our entire dataset into numerical values, this also allowed us to have a cleaner and more efficient dataset.

After the data was encoded, some columns had a significantly higher variance compared to other columns. To solve this problem, we performed scaling on all the columns in the dataset using the standard scaler, this minimized the variance of the data between different columns from our dataset and hence would give us more accurate results.

It is important to mention that three columns were excluded from the dataset after evaluating the precision of the logistic regression model. The columns which were excluded from the dataset were, education, fnlwgt, and relationship. The reason is that fnlwgt measures the total number of instances of the population that falls under that particular category which was not helpful in the model prediction. Education included the value of the highest level of education of an individual which is the same as the education_num column therefore it was not efficient to have two similar columns in the dataset and education was eliminated. Lastly, the relationship column could be determined from the marital status and sex column so it was determined that the relationship did not hold any significance in the dataset and it was dropped. The final dataset consisted of 11 input columns and 1 out column.

Logistic Regression

Logistic Regression is one of the easiest and most commonly used supervised Machine learning algorithms for categorical classification. The basic fundamental concepts of Logistic Regression are easy to understand and can be used as a baseline algorithm for any binary (0 or 1) classification problem. For this use-case, we are going to choose Logistic Regression as our classification model as it would be a good start for any beginner to start with a simple yet popular algorithm. Logistic Regression is a Statistical predicting model that can predict either a ‘Yes’ (1) or ‘No’ (0). It is based on a Logit or Sigmoid function which ranges between 0 and 1.

Our logistic regression model was initialized with a learning rate of 0.0000001, a tolerance rate of 0.0001 and the maximum iterations were set to 5. The maximum iterations were set to 5 by considering the size of our dataset which was fairly large, therefore to avoid computational drawbacks we fixed the maximum iterations

to 5. The dataset was converted into a float data type when it was passed into the Logistic Regression class to avoid any data type clashes, also after scaling was implemented on the dataset, the values were best suited for a float data type.

The model was then initialized and the training datasets were passed into the model, after the data was trained by the logistic regression model.

Support vector machines (SVMs)

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier detection.
The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoiding over-fitting in choosing Kernel functions and regularization terms are crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

Support Vector Machine work by marking each data point to a category and a total of two categories are present. These categories are divided by a hyperplane and the distance between the support vectors is maximized to ensure there is minimal fault in the classification of the data points.
The Support Vector Machine model is initialized with a learning rate of 0.00001, lambdaParam is set to 0.001 and the iterations are initialized to 100. It was necessary to keep the iterations to 100 to make sure that the loop executes enough times to accurately label all the vector data points that are fitted into the model. The testing data is then passed into the predict function that was defined in the Support Vector Machine class to output the predicted values.

Neural Network

A neural network works by combining several neurons. A neuron is a basic unit of the neural network which accepts an input, performs calculations on the input and produces an output. When several of these neurons are merged together a neural network model is obtained. Because of the existence of a network, there are layers of neurons in the network, this means that there will be an input layer, an output layer and several hidden layers in between the input and output layers.

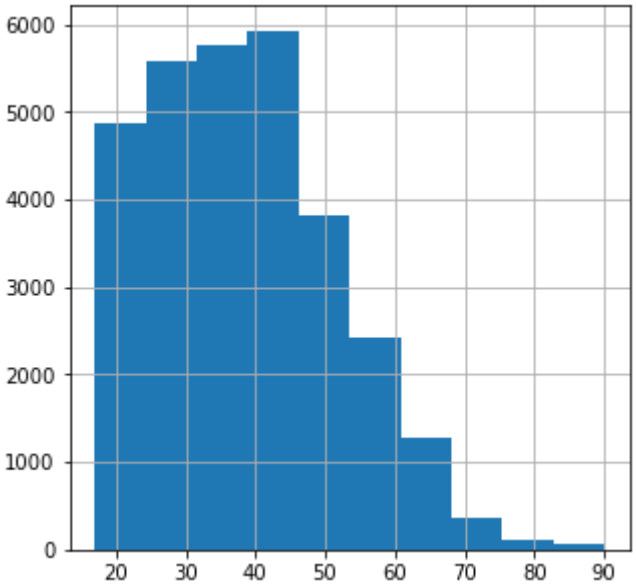
Since our output was in a binary format, the sigmoid activation function was implemented for the neural network in the output layer. We built our neural network model using the keras framework, the model was initialized as a sequential model. The reason behind using a sequential model was that sequential groups a linear stack of layers into the model, sequential provides training and inference features on the model, also because we are not implementing a non-linear topology it was ideal to use the sequential model so that we can build multiple layers with each of them having an input and output tensor. We added four layers to our model, with the first layer holding a dimensionality of 128 and the remaining three layers held a dimensionality of 256. The input dimension was set to 14 because those are the number of inputs our dataset has and the output dimension was 1 corresponding to the binary output we needed to obtain. The model was initialized by fitting the training data using a batch size of 64 because we needed to make sure our input size does not exceed the batch size. Finally, the model was evaluated on the testing data using a batch size of 128 to ensure all the necessary parameters are being considered and displayed as a result.

Exploratory data analysis (EDA)

During the Exploratory data analysis (EDA) our objective was to identify patterns in our data, understand the data, check for the distribution of the variables and try to predict which ones will be good estimators of the variable we want to predict.

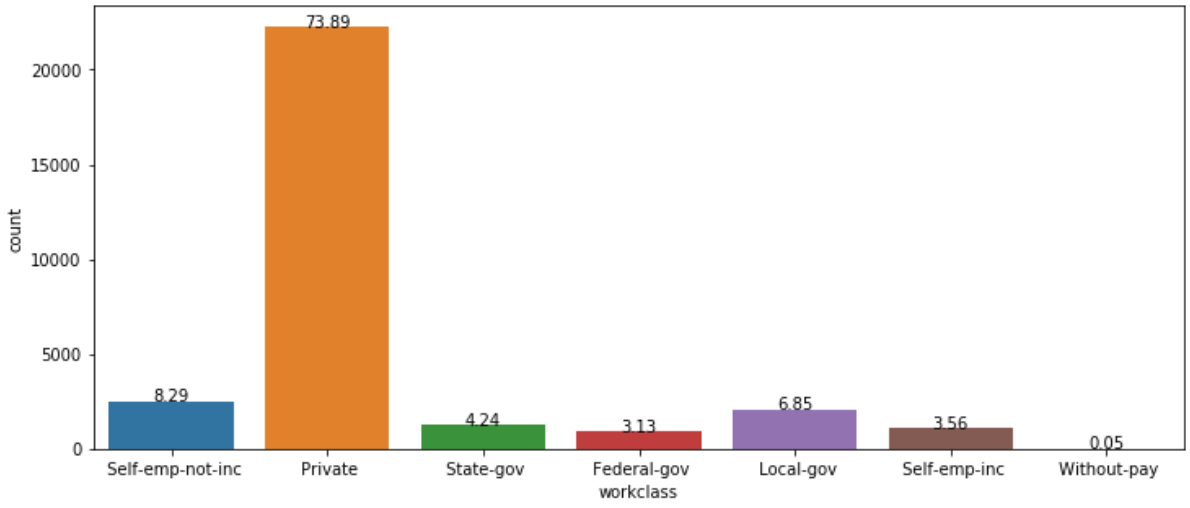
To facilitate the analysis, we will first understand the data to identify trends

1. Age: Discrete (from 17 to 90)



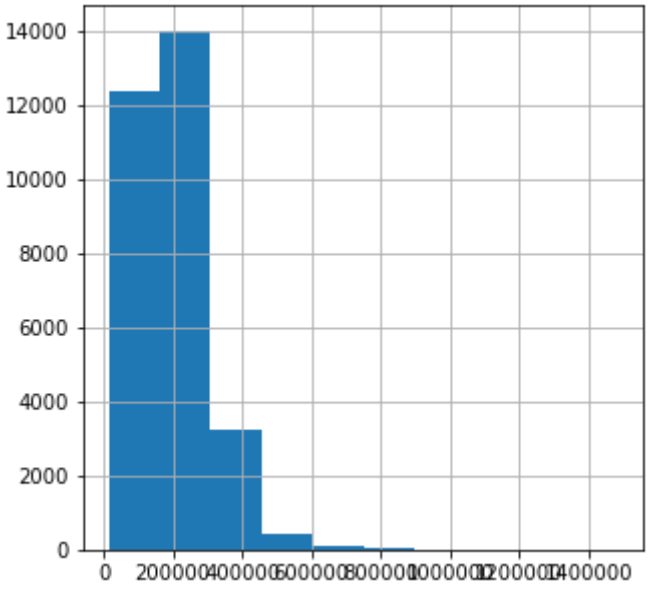
The distribution of 'Age' is right skew as most of the individuals are below age group 55.

2. Work class (Private, Federal-Government, etc): Nominal (9 categories)

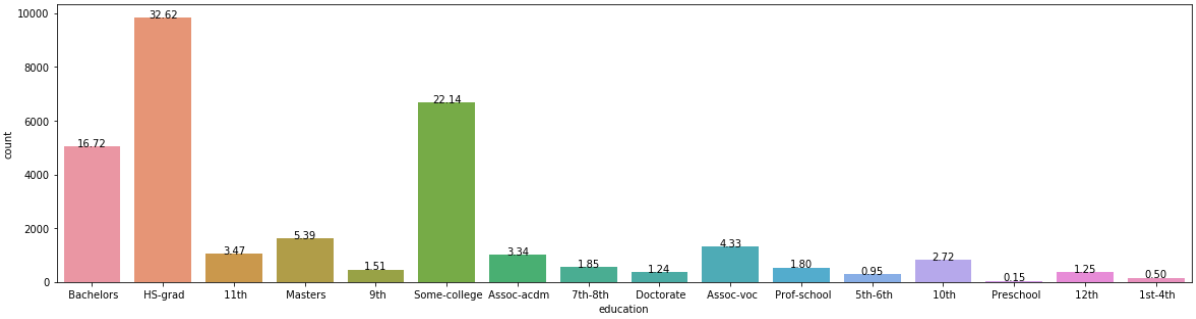


Majority of the sample population works in the Private Sector

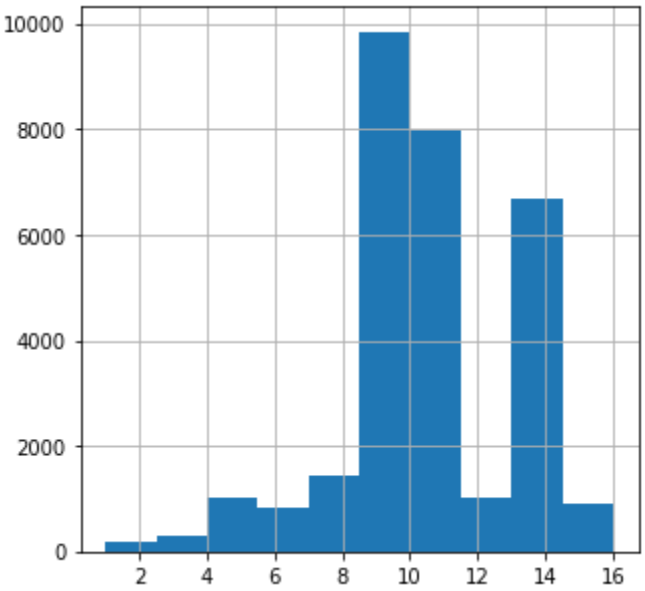
3. Final Weight (the number of people the census believes the entry represents): Discrete



4. Education (the highest level of education obtained): Ordinal (16 categories)

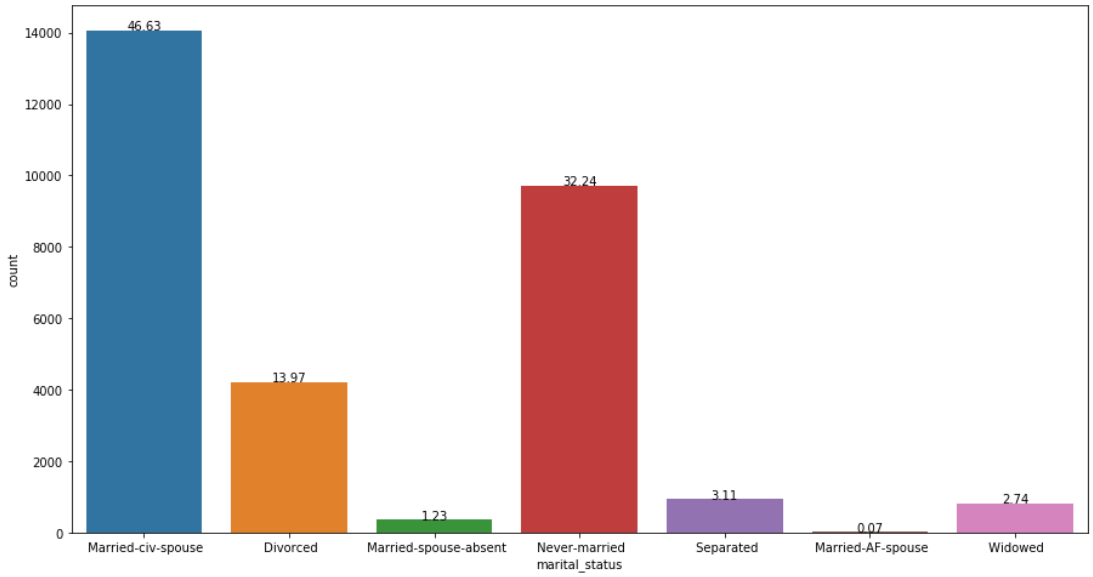


5. Education Number (the number of years of education): Discrete (from 1 to 16)

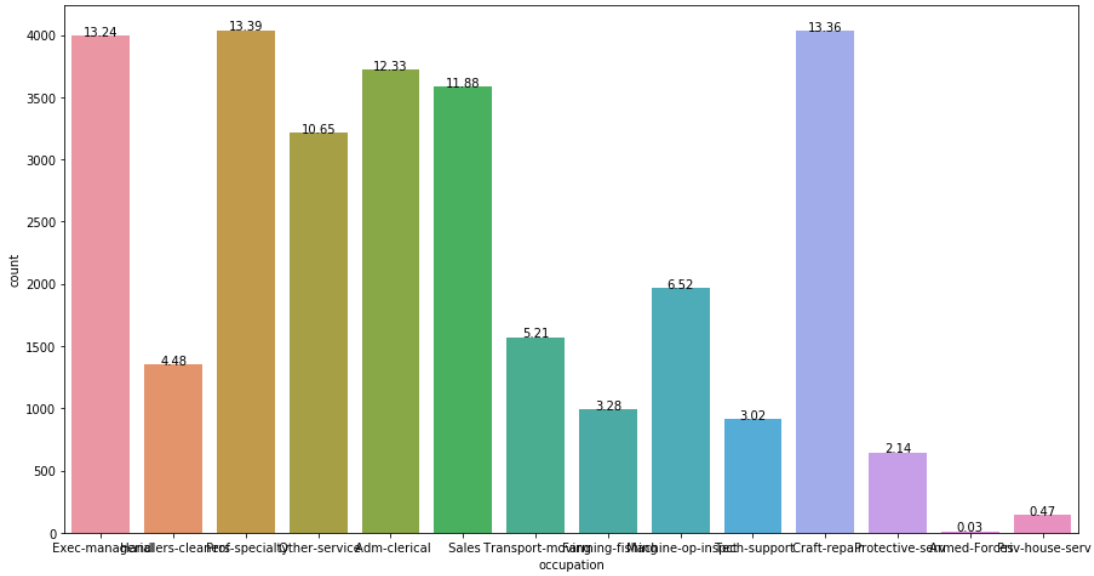


The average number of years of education for the given sample population is between 8-10 years.

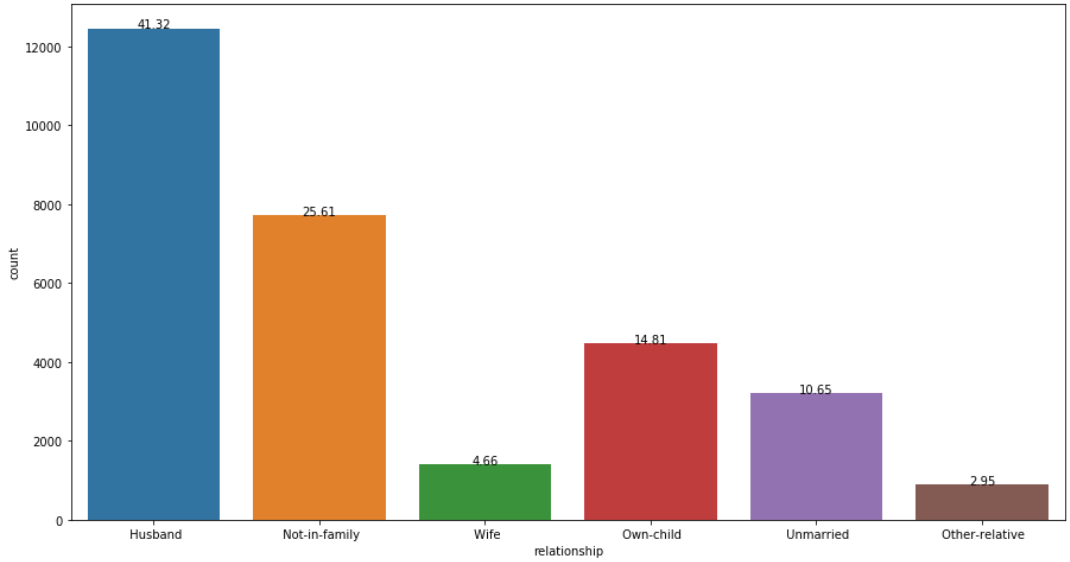
6. Marital Status: Nominal (7 categories)



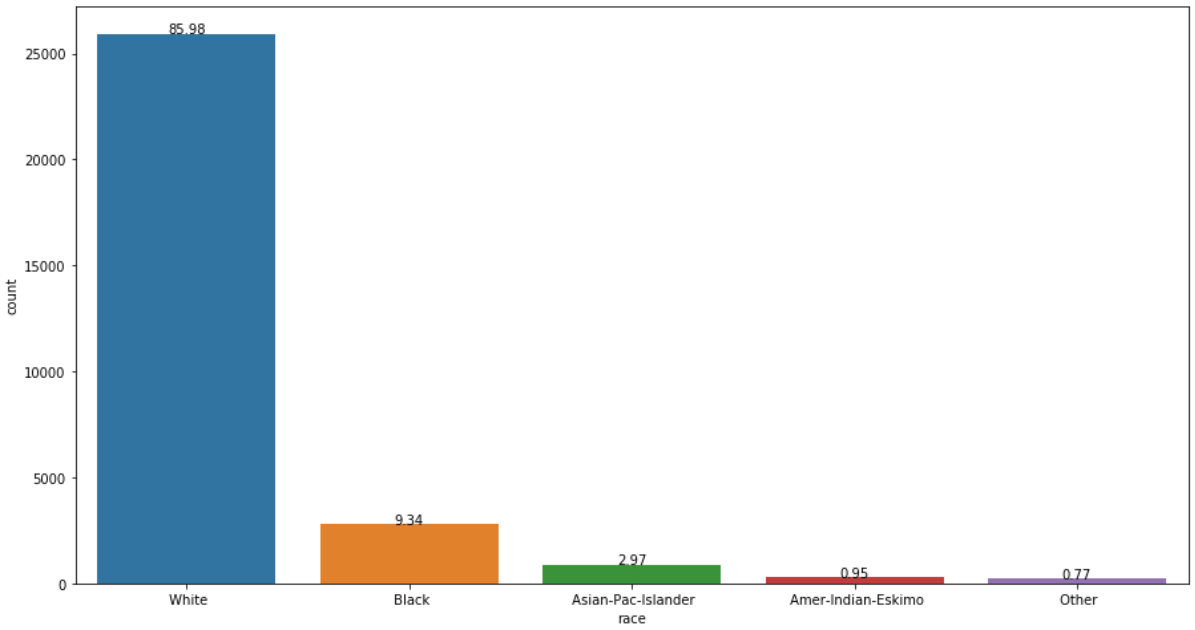
7. Occupation (Transport-Moving, Craft-Repair, etc): Nominal (15 categories)



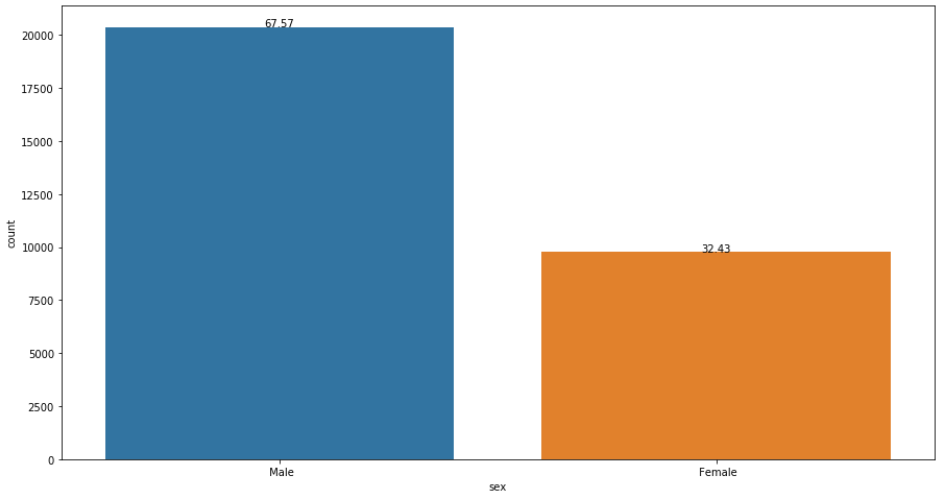
8. Relationship in family (unmarried, not in the family, etc): Nominal (6 categories)



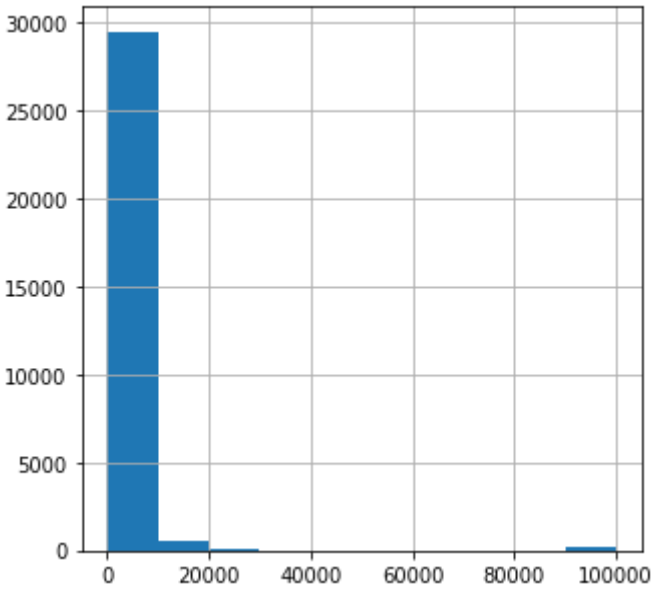
9. Race: Nominal (5 categories)



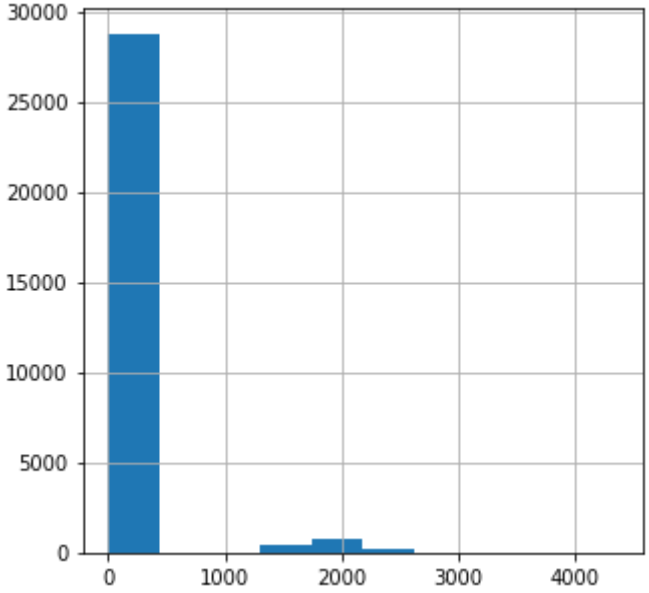
10. Sex: Nominal (2 categories)



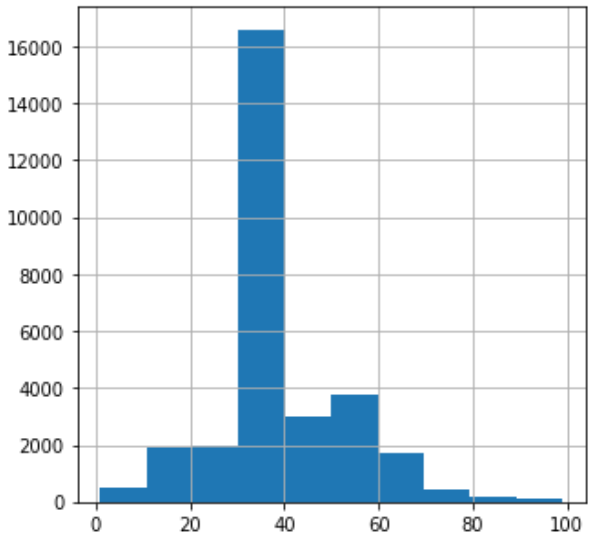
11. Capital Gain: Continuous



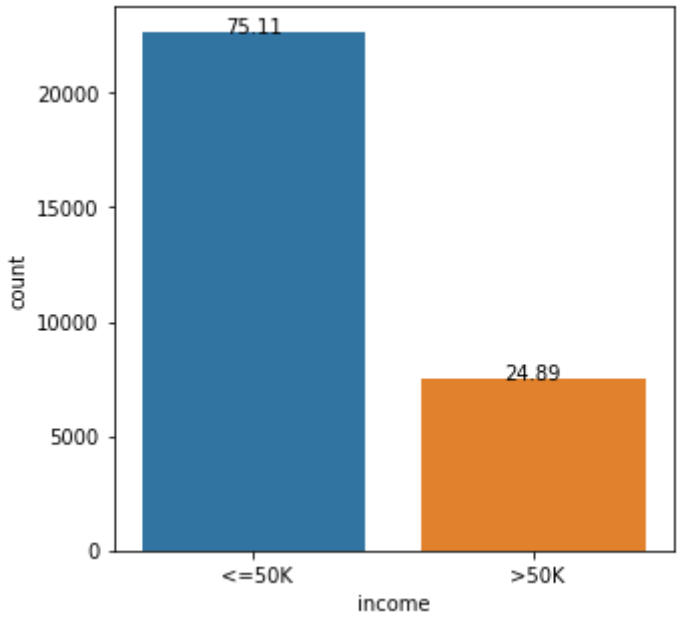
12. Capital Loss: Continuous



13. Hours (worked) per week: Discrete (from 1 to 99)



14. Income (whether or not an individual makes more than $50,000 annually): Boolean (≤$50k, >$50k)



RESULTS

After performing machine learning techniques on the dataset, the results obtained were analyzed and compared to conclude which models best fit the data. There were several metrics which were considered to evaluate the efficiency of a machine learning model, the most important metric being the accuracy of the model. Other metrics were also considered such as the precision value, recall value and the accuracy loss percentage. Below we will discuss the results obtained from all the three machine learning models and compare the results.

Logistic Regression

In the logistic regression model the metrics of evolution are the accuracy, the precision and the recall. Before analyzing the metrics of logistic regression, it is important to realize that factors such as the specific columns chosen as the input and the size of the dataset may have an impact on the performance of the logistic regression model.

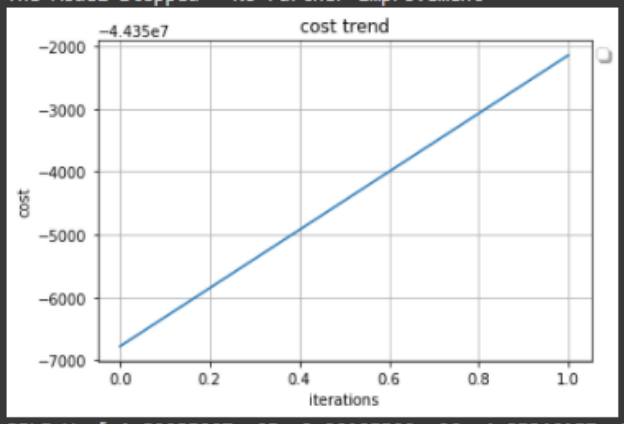
The bias-variance tradeoff is a phenomenon that occurs when there is a high bias in the dataset which signifies the under fitting of the dataset. When the logistic regression model was fitted on our dataset, it was noticed that the training dataset accuracy was lower than the testing dataset accuracy which suggested a high variance in the dataset. In order to prevent the under fitting problem, we have used a regularization technique. We have performed feature engineering to increase the number of features from 1000 to 10000 so that we can increase the model complexity.

Logistic Regression Technique 1

The following model consists of 10,000 random rows with all attributes. We tried to check if the size of the dataset has any impact on the performance of the model, therefore we fitted the model with 1000 instances and 10000 instances but both the dataset sizes resulted in a similar occurrence percentage. Therefore, we decided to use 10000 instances from a total of 30161 instances because firstly using all of the instances was heavily impacting the computer processor performance resulting in the kernel to shut down, and secondly there was no significant difference in the occurrence percentage.

```
1 # The model is executed here on the training set and the output is given from the testing set
2 lr.runModel(train_df,test_df)
```

```
(8000, 14)
  0%|          | 0/5 [00:00<?, ?it/s][ 6.61548774e-06 -4.95224266e-06  2.42748986e-05  1.13658907e-04
 -6.31238031e-05  2.25087777e-05 -8.42704257e-05  2.71100174e-05
  7.79905014e-05  8.12262755e-05  5.18777403e-05  7.88086933e-05
  6.74897422e-06  1.02350000e-04]
 20%|          | 1/5 [00:02<00:08,  2.20s/it]-44356779.80773362
[ 1.32235207e-05 -9.90105500e-06  4.85346157e-05  2.27281981e-04
 -1.26222125e-04  4.50085669e-05 -1.68501505e-04  5.42074644e-05
  1.55945988e-04  1.62424827e-04  1.03738762e-04  1.57582987e-04
  1.34932134e-05  2.04682479e-04]
 20%|          | 1/5 [00:04<00:17,  4.35s/it]
No handles with labels found to put in legend.
-44352143.316181295
The Model Stopped - No Further Improvement
```




```
SELF.W: [ 1.32235207e-05 -9.90105500e-06  4.85346157e-05  2.27281981e-04
 -1.26222125e-04  4.50085669e-05 -1.68501505e-04  5.42074644e-05
  1.55945988e-04  1.62424827e-04  1.03738762e-04  1.57582987e-04
  1.34932134e-05  2.04682479e-04]
[[3532 2421]
 [ 204 1843]]
Training Accuracy:  0.671875
Training Precision:  0.4322232645403377
Training Recall:  0.9003419638495359
```

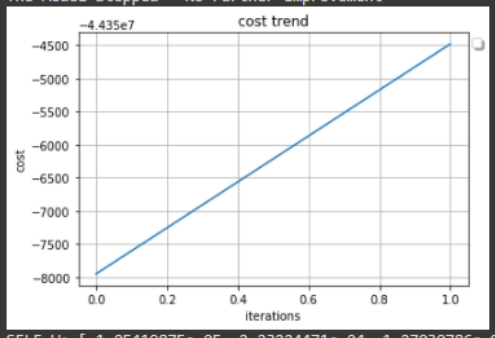
Training Accuracy: 0.671875
Training Precision: 0.4322232645403377
Training Recall: 0.9003419638495359

Logistic Regression Technique 2

The following model consists of 10,000 random rows with 'age', 'workclass', 'education_num', 'marital', 'occupation', 'race', 'capital gain', 'capital loss', 'hours per week', 'country' and 'income' columns. We have removed Final Weight (the number of people the census believes the entry represents), 'education_num' and relationships as they were not having a significant impact on the model as explained in our methodology section. We noticed here that there was a slight increase in the accuracy of the model which suggested that the additional columns did not hold much significance in our original dataset.

```
#removing the unnecessary columns
df = df[['age','workclass','education_num','marital','occupation','race','capital gain', 'capital loss', 'hours per week','country','income']]
train_df,test_df = train_test_split(df,train_size=0.8)
lr.runModel(train_df,test_df)
```

```
(8000, 10)
  0%|          | 0/5 [00:00<?, ?it/s][ 5.27298685e-06  1.11628558e-04 -6.35256393e-05  1.70305773e-05
  2.74180953e-05  7.97780418e-05  5.46954023e-05  7.72025776e-05
  8.32619487e-06  1.02050000e-04]
 20%|          | 1/5 [00:02<00:08,  2.14s/it]-44357950.749323905
[ 1.05419875e-05  2.23224471e-04 -1.27030786e-04  3.40544070e-05
  5.48265317e-05  1.59531226e-04  1.09375140e-04  1.54378780e-04
  1.66477061e-05  2.04085979e-04]
 20%|          | 1/5 [00:04<00:17,  4.26s/it]
No handles with labels found to put in legend.
-44354483.81308432
The Model Stopped - No Further Improvement
```




```
SELF.W: [ 1.05419875e-05  2.23224471e-04 -1.27030786e-04  3.40544070e-05
  5.48265317e-05  1.59531226e-04  1.09375140e-04  1.54378780e-04
  1.66477061e-05  2.04085979e-04]
[[3655 2304]
 [  92 1949]]
Training Accuracy:  0.7005
Training Precision:  0.4582647542910886 5
Training Recall:  0.9549240568348849
```

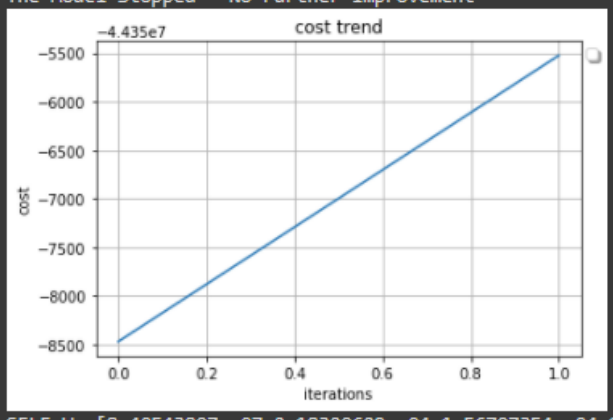
Training Accuracy: 0.7005
Training Precision: 0.45826475429108865
Training Recall: 0.9549240568348849

Logistic Regression Technique 3

The following model consists of 10,000 random rows with only numeric columns. The reason behind this fitting was to only consider the columns that were originally in a numerical format and discard the columns that had to be later converted into a categorical then a numerical format. This dataset when inputted into the logistic regression model gave the highest accuracy out of all the three techniques we performed on logistic regression. However, we cannot consider this technique as the best technique because the selection of columns from the dataset were only based on the data type and not on any other factors.

```
#numerical columns
df1 = df[['age','fnlwgt','education_num','capital gain', 'capital loss', 'hours per week','income']]
train_df,test_df = train_test_split(df1,train_size=0.8)
lr.runModel(train_df,test_df)
```

```
(8000, 6)
  0%|          | 0/5 [00:00<?, ?it/s][4.19631686e-07 1.09175152e-04 7.83648767e-05 5.20528952e-05
 7.85559365e-05 9.96000000e-05]
 20%|██        | 1/5 [00:02<00:08,  2.14s/it]-44358471.01306933
[8.40543807e-07 2.18320629e-04 1.56707354e-04 1.04092018e-04
 1.57088784e-04 1.99188039e-04]
 20%|██        | 1/5 [00:04<00:17,  4.26s/it]
No handles with labels found to put in legend.
-44355523.86458715
The Model Stopped - No Further Improvement
```




```
SELF.W: [8.40543807e-07 2.18320629e-04 1.56707354e-04 1.04092018e-04
 1.57088784e-04 1.99188039e-04]
[[4092 1916]
 [  86 1906]]
Training Accuracy:  0.74975
Training Precision:  0.4986917844060701
Training Recall:  0.9568273092369478
```

Training Accuracy: 0.74975
Training Precision: 0.4986917844060701
Training Recall: 0.9568273092369478

Therefore, we conclude that technique 3 gives the most efficient results when the logistic regression model is fitted on our dataset. But we will consider technique 2 to be the best technique because of the consideration of more columns which hold a significance to the prediction of the income of an individual.

Support Vector Machine

In the support vector machine model, the performance metric was the output given using the predict function that was included in the support vector machine model class. There were two functions in the support vector machine model, one was the fitting function and the other was the predict function. The training dataset was passed into the model to fit it, after which the testing dataset was passed into the prediction function. This gave the output which is shown below, there were a mixture of +1 and -1 values output which is what was expected because in the support vector machine model each data point is classified into a +1 or a -1. When these values were matched with the actual output some of them were correctly classified which was the +1 was corresponding to an income greater than $50k and a value of -1 corresponding to an income lesser than $50k. This model was found to output the least accurate prediction of the income among all three machine learning models.

```
[231] 1 print(clf.w)

[ 0.20190707 -0.00667047  0.21167526 -0.21065057  0.01398886  0.04997774
  0.09254809  0.05273568  0.18636243  0.0163812 ]

      1 print(clf.b)

[0.96529]

[233] 1 # Printing the predicted values using the X_test data
      2 print(clf.predict(X_test))

[1. 1. 1. ... 1. 1. 1.]
```

NEURAL NETWORK

The neural network machine learning model was implemented by including 4 layers of neuron layers creating a network. Once the sequential neural network model was built, the summer function was used to display all the information about the model and the layers the model consists of. After the summary was checked to contain the correct information, the model was recompiled and the training dataset was fitted in the neural network model. The neural network model resulted in an accuracy of 83% which is the highest accuracy compared to the other models and also giving a higher occurrence than the base model which was the logistic regression.

```
1 model.fit(X_train, y_train, batch_size=64, epochs=2)

Epoch 1/2
125/125 [==============================] - 1s 8ms/step - loss: 0.4081 - accuracy: 0.8073
Epoch 2/2
125/125 [==============================] - 1s 7ms/step - loss: 0.3455 - accuracy: 0.8374
<tensorflow.python.keras.callbacks.History at 0x7f11d6e4a750>

[240] 1 results = model.evaluate(X_test, y_test, batch_size=128)
      2 results

16/16 [==============================] - 0s 5ms/step - loss: 0.3560 - accuracy: 0.8315
[0.35595622658729553, 0.8314999938011169]
```

REFERENCES

[1] Donnelly, F. (2020, January 27). What is census data? Social Science Space. https://www.socialsciencespace.com/2020/01/what-is-census-data/.

[2] Wikimedia Foundation. (2021, July 23). Bias–variance tradeoff. Wikipedia. https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff.