IntelliPaat

Capstone Project

Census Income

Kavitha Palraj
kavipalraj.9@gmail.com

Table of Contents

INTRODUCTION 4

a. Use of Census Income Data 4
b. Census Income Data in this report 4

1. PROBLEM STATEMENT 4
2. PROJECT OBJECTIVE 4
3. DATA DESCRIPTION 5

a. Overview 5
b. Categorical attributes 5
c. Continuous attributes 5
d. Data overview 6
e. Insights from the data 6

4. INSPIRATION 6

DATA PRE-PROCESSING STEPS 7

a. Importing the libraries 7
b. Loading the dataset 7
c. Information summary of the dataset 7
d. Handling null values and missing values 8
e. Handling duplicate values 8
f. Removing unique characters and symbols 8
g. Outlier detection 8
h. Dropping columns 9
i. Correlation visualization 9
j. Label encoding 9
k. Multicollinearity 9

DATA VISUALIZATION 10

a. Distribution of ages 10
b. Distribution of people based on their highest level of education 11
c. Number of people who work in each occupation 11
d. Number of people who earn more than $50,000, grouped by their occupation 12
e. Number of people who less than or equal to $50,000, grouped by their occupation 13
i. Conclusion from graphs d and e 13
ii. Percentage Calculations 14
f. Number of people in each workclass grouped by annual income 14
i. Percentage Calculations 15
g. Hours worked per week based on the ages of individuals 15

MACHINE LEARNING MODEL BUILDING 16

1. FEATURE SCALING 16

a. Splitting the dataset 16
b. Feature scaling 16

2. Machine Learning Models 16
3. Choosing the Algorithm for the Project 17
a. Logistic Regression 17
b. Decision Tree 17
c. Random Forest 17
4. Motivation and Reasons For Choosing the Algorithm 18
5. Assumptions 18
a. Overall assumptions 18
b. Logistic Regression assumptions 18
c. Decision Tree assumptions 18
d. Random Forest assumptions 18
6. Model Evaluation and Techniques 19
a. Logistic Regression 19
b. Decision Tree 19
c. Random Forest 20
7. Random Forest Evaluation 21
a. Prediction outcomes 21
b. Accuracy 21
c. Confusion Matrix 21
d. Precision 21
e. Recall 21
f. F1-score 21
g. Overall evaluation 22
8. Inferences 22
CONCLUSION 23
a. Summary 23
b. Future possibilities of the project 23
REFERENCES 24

Introduction

Census income data provides a crucial statistical insight into the distribution of populations dependant on different characteristics such as income and education level. The data provides a holistic overview of a population from an economic, social and demographic standpoint (IAS Gatewayy, 2020).

a. Use of Census Income Data

Using this collected data, machine learning models can be built, trained and used to make informed decisions about specific problems affecting a variety of industries.

While providing immediate insights, this model can also be used in the forecasting and prediction of future outcomes. This enables the anticipation of any economic, business or government issues and provides the opportunity to prepare accordingly. Demographics of the population can be analysed which is pivotal in the allocation of government resources to public services and economic planning (Drishti IAS, 2020).

b. Census Income Data in this report

The division of labour within the economic system today has led to the specialisation of jobs and people focusing on specific skillsets such as data science or medicine (Madiedo, 2019). This has led to a competitive job market where companies are focused on hiring and retaining highly skilled employees (Forbes, 2023). One of the largest factors in achieving this goal is providing the correct salaries.

The dataset provided to us contains many different factors which influence salary including age, education level and the industry in which the employee is working. Using this information, a model can be trained and tested to predict the salary of a specific individual based on these factors.

1. Problem Statement

Perform a predictive task of classification to predict whether an individual makes over $50,000 a year or less by using different machine learning algorithms.

2. Project Objective

The objective of this project is to use supervised learning classification models to predict whether an individual makes over $50,000 a year or less. The different features of the dataset will be investigated and analysed to determine which of these impact the annual salaries of individuals. The features of the provided dataset are analysed below.

3. Data Description

a. Overview

The Census Income dataset that is available has been taken from the UCI Machine Learning Repository. This dataset was collected with the aim of recording and storing characteristics about a given population. It contains information about 32,561 individuals and each entry has 15 features which describe the corresponding person.

b. Categorical attributes

Variable	Description	Data Values
education	Highest form of education completed by the individual	Preschool , 1st-4th , 5th-6th, 7th-8th, 9th, 10th, 11th, 12th,Some-college, Bachelors, HS-grad, Prof-School, Assoc-acdm, Assoc-voc, Masters, Doctorate
education-num	Number corresponding to the type of education	1:Preschool, 2:1st-4th , 3:5th-6th, 4:7th-8th, 5:9th, 6:10th , 7:11th, 8:12th, 9:Hs-grad, 10:Some-college, 11:Assoc-voc, 12:Assoc-acdm, 13:Bachelors, 14:Masters, 15:Prof-school, 16:Doctrate
marital-status	Marital status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
native-country	Where the individual was born	Canada, Netherlands, United-States, Cambodia, Hong, Holand, Trinadad&Tobago, EI-Salvador, Yugoslavia, Thailand, Scotland, Nicaragua, Columbia, Hungary, Guatemala, Nicaragua, Haiti, Taiwan, Educador, Laos, Dominican-Republic, France, Ireland, Portugal, Mexico, Vietnam, Jamaica, Poland, Jamaica, Italy, Philippines, Honduras, Iran, Cuba, China, South, Greece, Japan, India, Outlying-US, German, Puerto-Rico, England
occupation	Job in which the individual is employed	Armed-forces , Prof-Speciality, Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-Fishing, Transport-moving, Privhouse-serv, Proctective-serv
relationship	Relationship status	Husband ,Wife, Own-child, Not-in-family, Other-relative, Unmarried
race	Race of the person	Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black, White
sex	Gender	Male, Female
workclass	Category of work of the individual	Private, Federal-gov, Self-emp-not-inc, Self-emp-inc, Local-gov, State-gov, Without-pay, Never-worked

c. Continuous attributes

Variable	Description
age	Age of an individual
fnlwgt	The final weights of people
capital-gain	Financial profit
capital-loss	Financial loss
hours-per-week	Hours worked each week by the individual

d. Data overview

The column 'annual_income' stores the target variable which describes whether a person makes over $50,000 a year or less. These conditions are represented by the two variables '<=50K' or '>50K'. The columns 'age', 'race', 'sex' and 'marital-status' describe personal aspects of each person while 'workclass', 'education', 'education-num' and 'hours-per-week' contain relevant education and employment details. 'fnlwgt' contains a weight value which emphasize the importance of each individual to the overall dataset.

e. Insights from the data

A few insights can be made from the data before the following pre-processing, EDA and model building steps. As mentioned above, the data contains information about 32,561 individuals. Around 76% of these people (24,720) earn less than or equal to $50,000 a year. 67% of the population are male and 90% are from the United States. 16% of people completed a Bachelors as their highest form of education. Around 5% completed a Masters degree and 1% did a PhD.

4. Inspiration

As briefly mentioned previously, there are a few points which highlight the importance of carrying out this project. Using the census income dataset, a model can be built and trained on the different features provided. An analysis can then be carried out as to which of these features contribute to the annual income of individuals. This model will then be tested using these features to predict whether a person makes above or less than $50,000 annually.

One example of where this model could excel is in the retaining and employment of skilled talent for specialised jobs by employers in a range of industries. Providing the optimal salary in this situation is important to satisfy individual employees. Further to this baseline investigation, additional questions will be asked and visualised such as how annual income varies dependent on the industry, at which age do people tend to earn higher salaries and what is the distribution of ages of people within the dataset.

Data Pre-processing Steps

Data pre-processing is an important step in improving the reliability and interpretability of the dataset (IBM, 2022). **Exploratory Data Analysis (EDA)** helps in showing the initial structure of the data. **Data Cleaning** is included within EDA which ensures the data is well structured and free from issues such as outliers and missing values. This enables us to analyse the data to a higher accuracy and create machine learning models which perform optimally. EDA is also used to create visualizations are such as boxplots and correlation heatmaps which help in the subsequent model building steps.

The individual steps that were carried out and their results are elaborated upon below:

a. Importing the libraries

The first step in EDA is importing all the libraries that will be necessary for the different process carried out.

1. `numpy` is used mainly for scientific computation and mathematical operations. It provides features such as a numpy array which is a powerful N-dimensional array object and functions for linear algebra.

2. `pandas` is used for data cleaning, pre-processing and manipulation. This is achieved using functions such as `.isnull()` and `groupby()` which will be seen later on. Series and DataFrames are used which structure data for manipulation and analysis.

3. `matplotlib` and `seaborn` are both used for data visualization.

b. Loading the dataset

Pandas is used to load and read the dataset from the provided .csv file. This is done using the `pd.read_csv()` function. This .csv file is then converted to a DataFrame which is used for the further pre-processing steps. The DataFrame is stored in the variable `df`.

c. Information summary of the dataset

The `df.info()` function is applied to the DataFrame which provides a breakdown of each column.

There are a total of `32,561` entries. The dataset has 15 different columns which were shown in the Data Description. Each column has `32,561 non-null` values. There are 6 columns with data type `int64` and 9 columns with data type `object`.

`df.dtypes` shows us the data type of each column that was seen with `df.info()`.

d. Handling null values and missing values

Leaving null and missing values within a dataset can compromise the integrity of the dataset which leads to bias in the analysis and a model which makes inaccurate predictions and conclusions. `df.isnull.sum()` and `df.isna.sum()` show the dataset has no null or missing values that need to be removed.

e. Handling duplicate values

Duplicate values can lead to a biased analysis, inaccurate results and a machine learning model which overfits due to learning patterns from duplicated data. Using `df.duplicated.sum()`, the dataset contains 24 duplicated values. When analysing this further, this represents `0.07%` of the overall dataset. Due to this insignificant proportion the duplicated values can be ignored and don't need to be dealt with.

f. Removing unique characters and symbols

Null, missing and duplicated values have been dealt with. Other unique characters and symbols are not supported by machine learning models. Therefore, they need to be identified and replaced with NaN values which can be removed from the dataset.

Using the for loop `(for i in df.columns)`, each column in the dataset is iterated through. An if statement checks which columns contain `'?'` symbols. This investigation reveals the three columns ['workclass', 'occupation', 'native-country'] contain this symbol.

This symbol is replaced with NaN values using `df.replace({'?':np.nan})`. This changes 5.64%, 5.66% and 1.79% of the three respective columns to null values. `df.dropna()` is then used to drop these null values from the columns which changes the DataFrame shape from `(32561, 15)` to `(30162, 15)`. This is 92.6% of the original DataFrame size.

g. Outlier detection

The detection of outliers is a very important step in data pre-processing. It involves spotting values that deviate significantly from the overall pattern and removing them. Outliers can cause distortions in a dataset which lead to a machine learning model making inaccurate predictions.

Box plots are plotted for numerical columns using `sns.boxplot()` within a for loop which iterates over the dataset columns.

Following this visualization, outliers are detected. Upper and lower quantiles are used to calculate an IQR and values which fall outside this IQR are labelled as outliers. These outliers are removed from the DataFrame using `df=df[(df[i]>=LW) & (df[i]<=UW)]`.

h. Dropping columns

The 'capital-gain' and 'capital-loss' columns are removed from the dataset using `df.drop()`. This is because 92% and 95.2% of their respective columns contain the 0 value. These columns are therefore statistically insignificant.

i. Correlation visualization

The `.corr()` and `sns.heatmap()` functions are used to show the correlation between numeric columns. The numerical columns are extracted using `df.select_dtypes(include='number')`. There are weak correlations between all 4 numeric columns.

j. Label encoding

Label encoding is used to transform categorical variables to numerical variables. This process is needed because machine learning models require `int` and `float` as input data types. This is done using the `LabelEncoder()` function. A for loop is used to iterate over all columns with `object` data type. `le.fit_transform()` is used to convert the data in these columns to `int`.

The numerical columns are then visualised in correlation plots as done previously. These plots show very weak correlations between all columns.

k. Multicollinearity

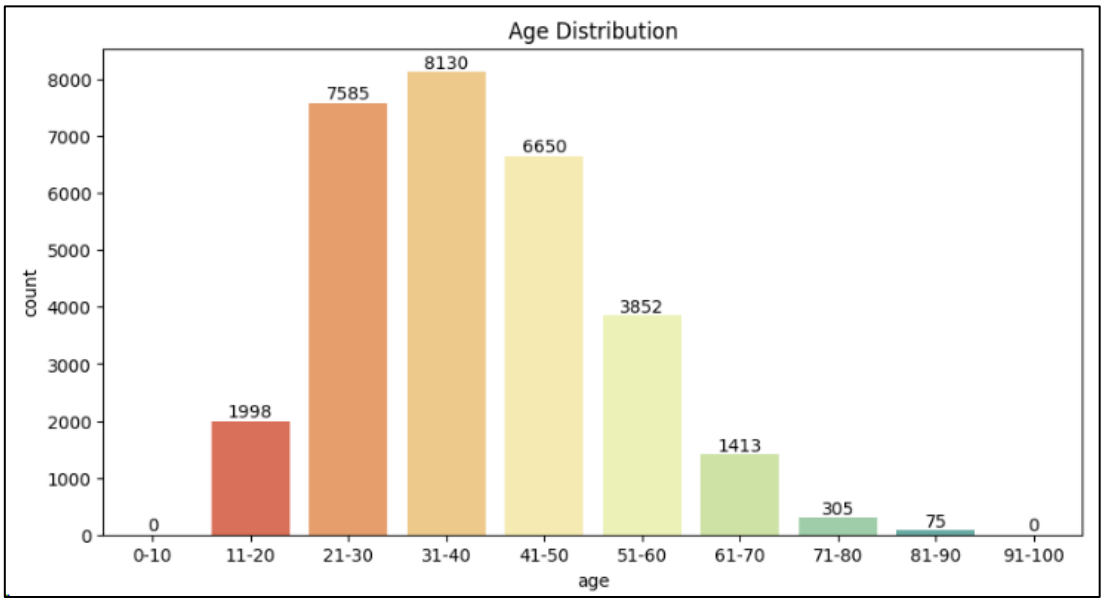
VIF (Variance Inflation Factor) is used to statistically analyse the extent of multicollinearity within a dataset. Multicollinearity is related to the correlation between two or more independent columns in the data. This high correlation leads to skewed results when trying to determine predictions for a dependent variable (Investopedia, 2023).

Using VIF, the column with the highest VIF value is dropped until all columns are below 5. With this process, the columns 'hours-per-week', 'education-num', 'native-country', 'race', 'education' and 'age' are dropped due to their high VIF values.

Data Visualization

Data visualization is a very important step. It transforms information within the dataset into a visual format where it is easier to spot trends, patterns and draw insights from the data. Alongside the given problem statement, these visualizations provide further data exploration to understand different relationships present (TechTarget, 2022). These relationships can be explained and any outliers or anomalies can also be identified.

a. Distribution of ages

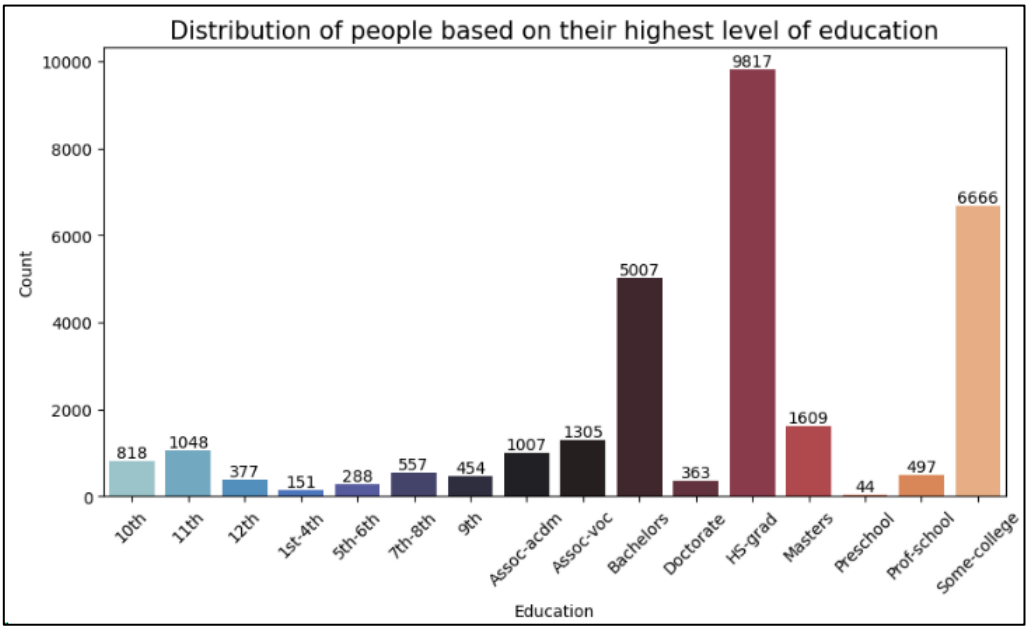


This graph shows the distribution of ages within the dataset. The graph shows most individuals fall within 31 to 40 years of age with 8,130 people. This number gradually declines until the 81 to 90 age range. The second most populous age range is 21 to 30 with 7,858 people.

`sns.countplot()` was used to plot this graph. A count plot is used to compare the counts of observations in each category. This graph shows the distribution of ages. The ages have been categorised into bins which are 10 ages apart. Using `pd.cut()` the 'age' column is segmented into these bins with the corresponding y axis labels.

The count plot was stored as 'graph1'. The `bar_label()` function was applied to 'graph1' which places each value above the respective bar. This enables the distribution of values to be quickly analysed.

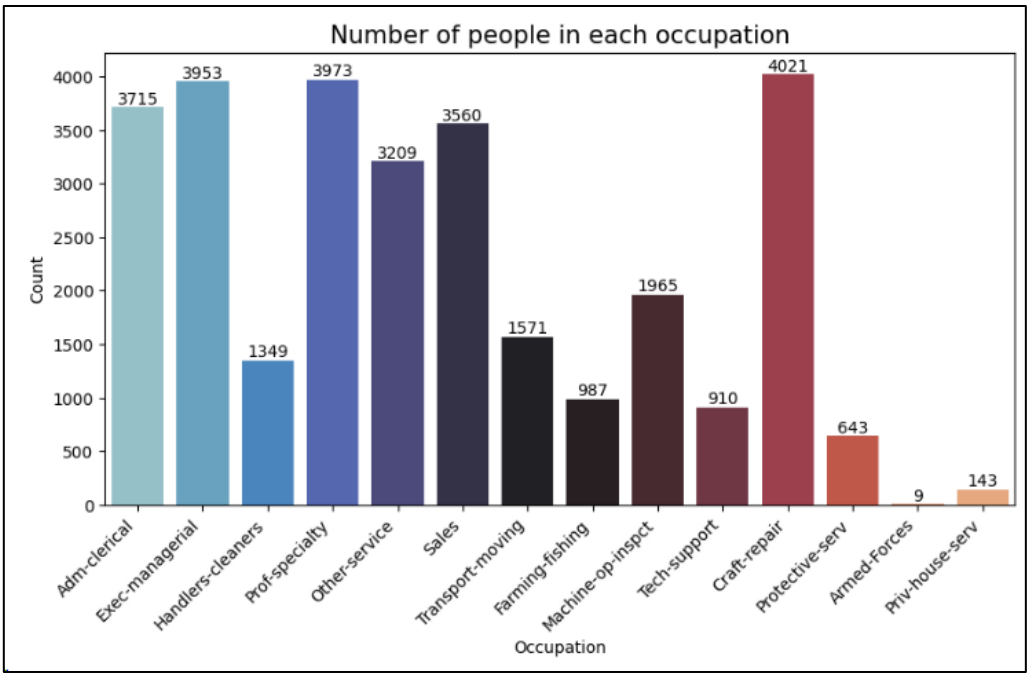
b. Distribution of people based on their highest level of education



This graph demonstrates the individuals within the dataset have a varied background of education levels. 9,817 people have 'HS-grad' as their highest education level which is the highest count. In terms of higher education, 5,007 people completed a 'Bachelors' degree, 1,609 people completed a 'Masters' degree and 363 people completed a 'Doctorate'.

`pd.DataFrame()` was used to create a new DataFrame containing the `value_counts()` of each education level. This was plotted using `sns.barplot()`.

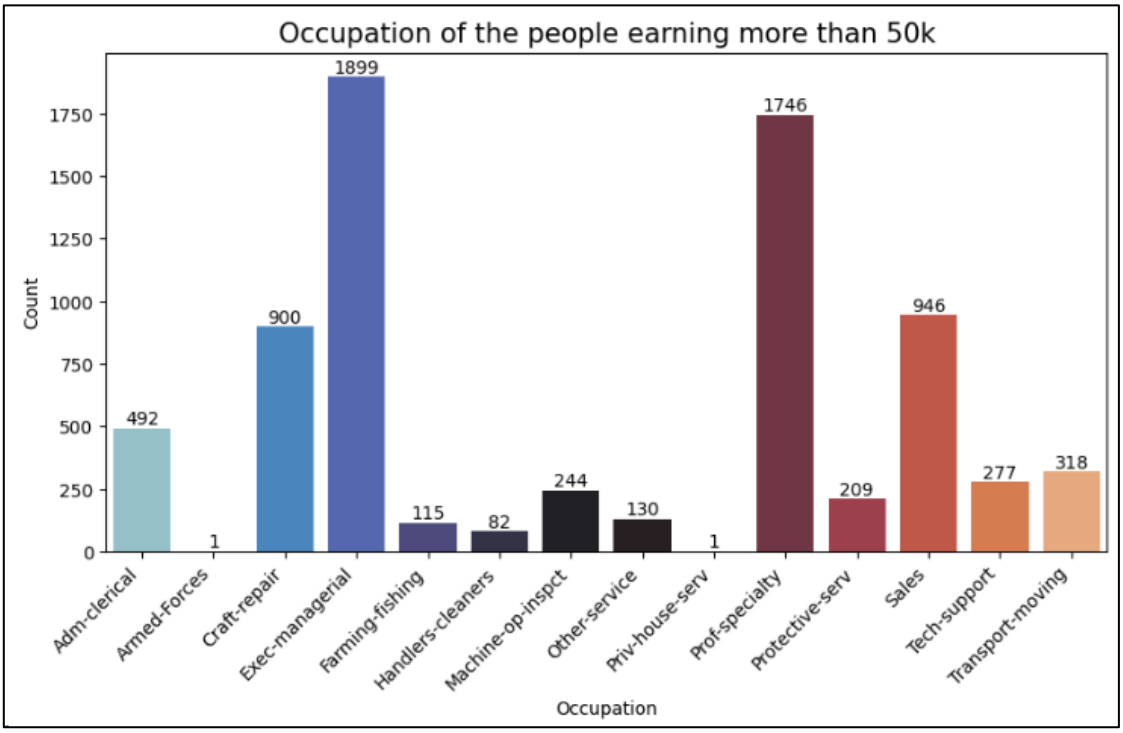
c. Number of people who work in each occupation



This graph shows the 3 occupations with the most employees are ‘Exec-managerial’, ‘Prof-specialty’ and ‘Craft-repair’ with 3,953, 3,973 and 4,021 individuals working in each respectively. The job with the least employees is ‘Armed-Forces’ where only 9 individuals are working.

To plot this graph, `sns.countplot()` was used again with `df['occupation']` accessed as the data source.

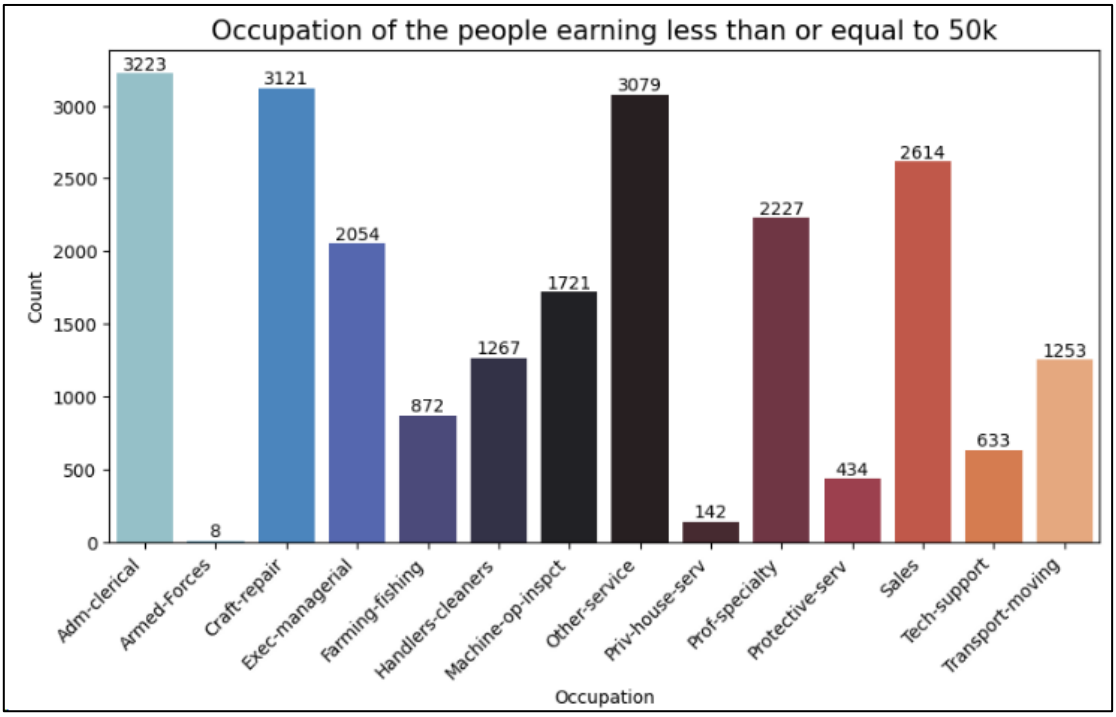
d. Number of people who earn more than $50,000, grouped by their occupation



Following the previous graph plotted, the next step was to analyse which occupations have people earning over $50,000 annually. ‘Exec-managerial’ and ‘Prof-specialty’ have the most individuals earning over $50,000, at 1,899 and 1,746 people respectively. This is 48% and 44% of all people working within these occupations. ‘Exec-managerial’ (48%) has the highest percentage of employees earning a salary greater than $50,000. ‘Craft-repair’ has the highest number of total employees at 4,021 but only 22% of these people earn a salary greater than $50,000.

To create this graph a new DataFrame was created with the count of occupation filtered by annual_income that is equal to ‘>50K’.

e. Number of people who less than or equal to $50,000, grouped by their occupation



'Adm-clerical' have the most employees earning less than or equal to $50,000 at 3,223 people. This is 87% of their work force. 'Priv-house-serv' has the largest proportion of employees earning this salary at 98%. 'Other-service' has the second highest proportion at 96% (3,079 out of 3,209). Although the most people earning more than $50,000 were 'Exec-managerial' (1,899), the majority of their employees (52%) earn less than or equal to $50,000.

i. Conclusion from graphs d and e

The two previous graphs have highlighted how annual income is highly dependent on the occupation of the individual.

The most people earning above $50,000 work in 'Exec-managerial'. They have the highest proportion of people earning this salary at 48%. 'Priv-house-serve' has the lowest proportion at 0.7% with 'Other-service' having only 4% (130 out of 3,079 people).

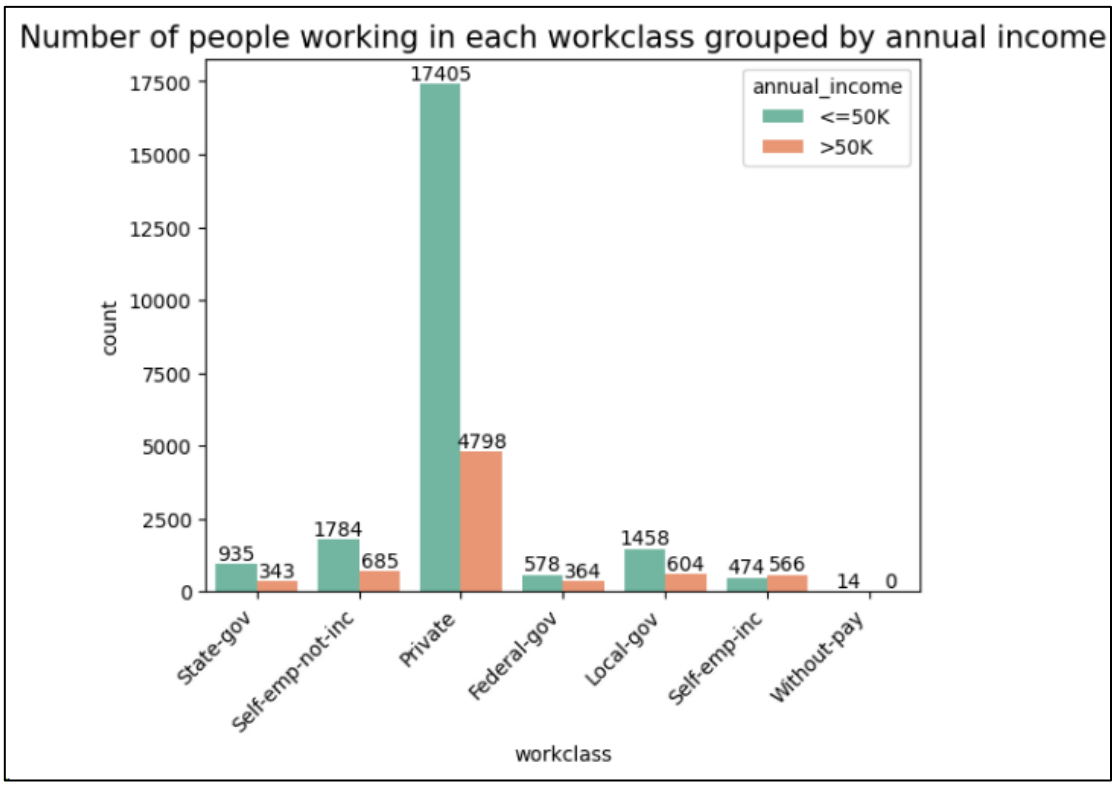
The occupation with the highest proportion of employees earning less than or equal to $50,000 is 'Priv-house-serve' with 99.3% followed by 'Other-services' with 95%. 'Adm-clerical' has the most employees earning this salary at 3,223.

ii. Percentage Calculations

To help with analysing the statistics, the percentages were calculated occupation wise for the number of people earning greater than or less than $50,000 out of the total number of people. The breakdown of these calculations can be seen in the table below.

Occupation	Total No. People	No. People With Salary <= 50K	% of Total No. People	No. People With Salary > 50K	% of Total No. People
Adm-clerical	3,715	3,223	**87%**	492	**13%**
Armed-Forces	9	8	**89%**	1	**11%**
Craft-repair	4,021	3,121	**78%**	900	**22%**
Exec-managerial	3,953	2,054	**52%**	1,899	**48%**
Farming-fishing	987	872	**88%**	115	**12%**
Handlers-cleaners	1,349	1,267	**94%**	82	**6%**
Machine-op-inspect	1,965	1,721	**88%**	244	**12%**
Other-service	3,209	3,079	**96%**	130	**4%**
Priv-house-serv	143	142	**99.3%**	1	**0.7%**
Prof-specialty	3,973	2,227	**56%**	1,746	**44%**
Protective-serv	643	434	**68%**	209	**32%**
Sales	3,560	2,614	**73%**	946	**27%**
Tech-support	910	633	**70%**	277	**30%**

f. Number of people in each workclass grouped by annual income



The 'Private' workclass has the largest number of employees at 22,203. 17,405 (78%) of these people earn a salary less than or equal to $50,000. Excluding 'Without-pay', 'Private' is the highest proportion of a workclass earning less than or equal to $50,000. 'State-gov', 'Self-emp-not-inc' and 'Local-gov' have 73%, 72% and 71% of their employees earning this salary respectively. The 'Private' workclass has the highest number of employees (4,798) earning more than $50,000 but this is only 22% of their workforce. Self-emp-inc has the highest proportion of employees earning this salary at 54%.

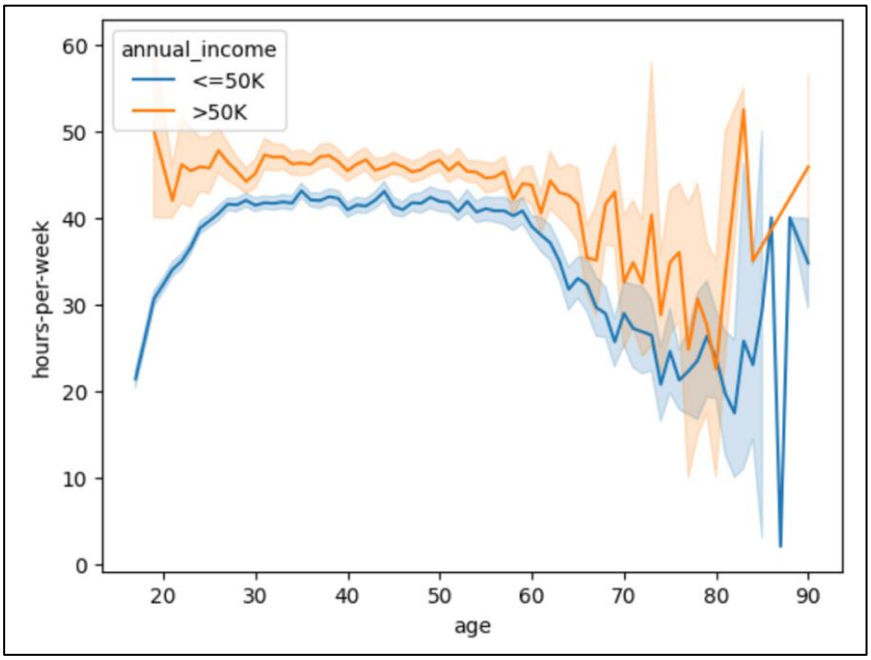
A countplot was used to plot this graph with the workclass on the x axis and a hue of annual_income.

i. Percentage Calculations

The proportion of people earning each salary to the total number of people for each 'workclass' has been calculated in the table below.

Workclass	Total No. People	No. People With Salary <= 50K	% of Total No. People	No. People With Salary > 50K	% of Total No. People
State-gov	1,278	935	**73%**	343	**27%**
Self-emp-not-inc	2,469	1,784	**72%**	685	**28%**
Private	22,203	17,405	**78%**	4,798	**22%**
Federal-gov	942	578	**61%**	364	**39%**
Local-gov	2,062	1,458	**71%**	604	**29%**
Self-emp-inc	1,040	474	**46%**	566	**54%**
Without-pay	14	14	**100%**	0	**0%**

g. Hours worked per week based on the ages of individuals



This lineplot shows the average 'hours-per-week' value for each age separated based on their annual salary. The graph clearly shows the longer hours people work a week, the higher salary they make.

Machine Learning Model Building

1. Feature Scaling

Before data can be used to train the machine learning model, a final step of pre-processing is required. Feature scaling is an important step of the model building process. It involves the normalisation and standardisation of the data which makes sure all the data is on the same scale and improves the performance of the model (Analytics Vidhya, 2024). Since the features are all in the same scale, they will now contribute equally in the machine learning model.

The 2 steps in the feature scaling process are explained below:

a. Splitting the dataset

The dataset is first divided into dependent and independent variables. The dependent variable in this case is 'annual_income' as we are trying to predict whether an individual makes more than or less than $50,000. This is stored in `x` while the independent variables are stored within `y`.

This data is then split into the test and train datasets using the `train_test_split()` function with a test size of 20%.

b. Feature scaling

Feature scaling is the next step to perform. To carry out this step, the `StandardScalar()` object is imported. `sc.fit_transform()` is applied to the x_train dataset and `sc.transform()` is applied to the x_test dataset. These datasets are now ready to be used to train the machine learning models

2. Machine Learning Models

The main step is the building the machine learning models, fitting and training these models and then using these models to predict the desired outputs.

Problem Statement: After having sufficient knowledge about the attributes, you will perform a predictive task of classification to predict whether an individual makes over $50,000 a year or less by using different machine learning algorithms.

3. Choosing the Algorithm for the Project

Since this is a classification task, different classification models will be used and their accuracies compared to determined which model produces the most accurate output to address the problem statement.

The 3 chosen models are explained below and their outputs analysed:

a. Logistic Regression

A logistic regression model is a multivariate model which is used to investigate the relationship between one dependent variable and multiple independent variables (Boateng, 2019). The dependent variable is categorical and should fit into one of two categories while the independent variables can be continuous or discrete.

This algorithm is simple to implement and can process large amounts of data efficiently. It outputs a single probability score which assesses the accuracy of the model.

b. Decision Tree

Decision tree models are used for classification and regression tasks. The algorithm works by splitting the dataset into subsets of data based on decisions at each node related to a chosen feature (Xu, 2023).

The model can easily handle nonlinear patterns within the data and can process numerical and categorical data. The process is easy to interpret and it performs efficiently with large datasets. A problem with this algorithm is it is prone to overfitting when the tree is too deep and contains too many nodes. This can be rectified using techniques such as limiting the depth based on hyper-parameters and pruning the tree (Scikit-learn, 2020).

c. Random Forest

Random forest extends the decision tree model by training multiple decision trees which are used together to reach a single prediction. The most common opinion among the ensemble of decision tree models will be chosen. The input data is randomised into different subsets which are used to train different trees. This leads to a more accurate and robust prediction (Career Foundry, 2023).

While overfitting was an issue with the decision tree algorithm, the ensemble of trees and use of feature masking reduces overtraining. This model is also able to handle large datasets with a large number of features (Built In, 2023). However, due to the size of the model it may be less interpretable and use more memory and computational power.

4. Motivation and Reasons For Choosing the Algorithm

From the analysis carried out above, **random forest** is the chosen algorithm because it provides the most accurate and robust prediction.

5. Assumptions

Defining assumptions is an important step before building machine learning models. It enables you to understand the performance and interpretation of the model and define parameters to ensure the predictions are accurate and robust (TowardsDataScience, 2018).

a. Overall assumptions
   - Data cleaning and pre-processing has taken place so the dataset is free from duplicated and null values, outliers and multicollinearity.
   - The training data accurately represents the dataset distribution.
   - The sample size is sufficient to ensure a reliable prediction is reached.

b. Logistic Regression assumptions
   - The dependent variable is binary and categorical.
   - There should be a linear relationship between the independent variables.

c. Decision Tree assumptions
   - The data is repeatedly split into further subsets of data to reach distinct categories.
   - A linear relationship between the variables is not required.
   - Both numerical and categorical features can be used without a need for feature scaling or label encoding.
   - Hyper-parameters are tuned and pruning and early stopping are used to limit overfitting.

d. Random Forest assumptions
   - An ensemble of decision trees provides a more robust prediction.
   - Like with decision trees, numerical and categorical features can be used.
   - Using an ensemble of decision trees and feature masking helps reduce overfitting.
   - Hyper-parameter tuning has less impact than with decision trees but it can improve the prediction accuracy.

6. Model Evaluation and Techniques

The summary of the accuracy of the 3 models can be seen in the following table:

Algorithm	Accuracy
Random Forest	82.3%
Decision Tree	81.3%
Logistic Regression	79.9%

a. Logistic Regression

The `.fit()` function from the `LogisticRegression()` class was used to train and test the model. Following this, the model was able to provide predictions with a **75% accuracy**.

```
# Printing the classification report
print(classification_report(y_test,y_pred_lo))
```

```
              precision    recall  f1-score   support

           0       0.75      1.00      0.86      4519
           1       0.24      0.00      0.01      1483

    accuracy                           0.75      6002
   macro avg       0.50      0.50      0.43      6002
weighted avg       0.63      0.75      0.65      6002
```

In the classification report above, the model correctly predicted salary '<=50K' which is represented by 0 to a **75% precision** and '>50K' salaries to a **24% precision**.

b. Decision Tree

The Decision Tree model gave an **initial 75.9% accuracy** after being trained and tested using the `.fit()` function from the `DecisionTreeClassifier()` class. A for loop was then used to compare the accuracy of the Decision Tree model using different tree depths. Tree depth of 9 had the highest accuracy at 81.6%.

Next, `GridSearchCV()` was used with the following parameters:

Parameter	Description	Values
max_depth	Maximum node depth of the tree	[none, 5, 10, 15]
min_samples_split	Minimum number of splits	[2, 5, 10]
min_samples_leaf	Minimum number of leaves	[1, 2, 4]
critereon	Method to determine node splits	['gini', 'entropy']

Grid search identified the optimal hyperparameters from these values provided. Another Decision Tree model was then trained and tested using these new hyperparameters. This model gave a **81.2% accuracy** which is 5.3% higher than previously.

```
# Printing the classification report
print(classification_report(y_test,dt_pred))
```

```
              precision    recall  f1-score   support

           0       0.83      0.94      0.88      4519
           1       0.69      0.43      0.53      1483

    accuracy                           0.81      6002
   macro avg       0.76      0.68      0.71      6002
weighted avg       0.80      0.81      0.79      6002
```

The classification report shows '<=50K' values were identified with **83% precision** and '>50K' values with **69% precision**.

c. <u>Random Forest</u>

The dataset was trained and tested with the Random Forest model using the `RandomForestClassifier()` class. The initial **accuracy score was 78%**. A for loop was then used to train and test Random Forest models against an array containing different numbers of trees. 600 trees produced the highest accuracy at 78.9%.

As with the Decision Tree model, the optimal hyperparameters were then found using GridSearchCV using the values below.

Parameter	Description	Values
n_estimators	Number of trees	[100, 300, 500, 700]
max_depth	Maximum node depth of the tree	[none, 5, 10, 15]
min_samples_split	Minimum number of splits	[2, 5, 10]
min_samples_leaf	Minimum number of leaves	[1, 2, 4]
critereon	Method to determine node splits	['gini', 'entropy']

Using the optimal hyperparameters, the new model gave an **accuracy of 82.6%**.

```
# Printing the classification report
print(classification_report(y_test,new_pred))
```

```
              precision    recall  f1-score   support

           0       0.86      0.92      0.89      4519
           1       0.69      0.53      0.60      1483

    accuracy                           0.83      6002
   macro avg       0.78      0.73      0.75      6002
weighted avg       0.82      0.83      0.82      6002
```

The classification report shows '<=50K' values were identified with **86% precision** and '>50K' values with **69% precision**.

7. Random Forest Evaluation

The following techniques and steps were involved in the evaluation of the Random Forest model.

a. Prediction outcomes
There are four values provided in the confusion matrix which are used to judge the correctness of predictions.

Outcome	Description	Values
TP (True Positive)	Outcome was positive and predicted positive	4,168
TN (True Negative)	Outcome was negative and predicted negative	792
FP (False Positive)	Outcome was negative but predicted positive	351
FN (False Negative)	Outcome was positive but predicted negative	691

b. Accuracy
Accuracy gives the ratio of correct predictions to all predictions. The overall accuracy is 82.6%.

c. Confusion Matrix
Confusion Matrix provides calculation of correct and incorrect classifications for each class. From the confusion matrix observed there is a 83% for correct predictions.

d. Precision
Precision is used to assess how well the model correctly classified positive instances: TP/(TP+FN). Precision scores for '<=50K' (class 0) and '>50K' (class 1) are 86% and 69% correspondingly.

e. Recall
Recall provides correct positive results divided by all actual positive instances considering both true positive and false negative instances: TP/(TP+FN). Class 0 has recall 92% and class 1 has recall 53%.

f. F1-score
It is a measure of test accuracy that considers both the precision and recall of the test to compute the score. The f1-score for this model for '<=50K' (class 0) is 89% and '>50K' (class 1) is 60%.

g. Overall evaluation

Problem Statement: After having sufficient knowledge about the attributes, you will perform a predictive task of classification to predict whether an individual makes over $50,000 a year or less by using different machine learning algorithms.

The evaluation of Random Forest in correctly predicting whether an individual made over $50,000

Evaluation	Score
Accuracy	83%
Precision	69%
Recall	53%
F1-score	60%

The evaluation of Random Forest in correctly predicting whether an individual made less than or equal to $50,000

Evaluation	Score
Accuracy	83%
Precision	86%
Recall	92%
F1-score	89%

8. Inferences

- The model performs better in predicting whether an individual makes less than or equal to $50,000 (class 0) instead of more than $50,000 (class 1). This was seen through higher precision, recall and F1-score values.
- Although the Random Forest model performs worse with class 1 results, these results are higher than those with the other machine learning models.
- The weighted average suggest that the model is relatively balanced.

Conclusion

a. Summary

The aim of this project was to address the problem statement and make predictions about whether a person makes less, equal to or more than $50,000 annually dependent on other variables. This information about individuals was retrieved from a census income dataset.

The 3 supervised learning models Logistic Regression, Decision Tree and Random Forest were trained, tested and evaluated to understand which produces the most robust and accurate classification predictions. From the steps carried out, the hyperparameter tuned Random Forest model performed the best with an overall accuracy of 82.6%. It also had the overall highest precision, recall and f1-score values.

b. Future possibilities of the project

If this project were to be carried out again, it would be beneficial to use a more recent census income dataset which is representative of a current population. The dataset provided was from the 1994 Census database. Furthermore, an overall better set of results could be achieved by using a combination of machine learning and deep learning models.

References

Analytics Vidhya, 2024. *Feature Scaling: Engineering, Normalization and Standardization.* [Online] Available at: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/ [Accessed January 2024]

Boateng, Ernest Yeboah & Abaye, Daniel. (2019). A Review of the Logistic Regression Model with Emphasis on Medical Research. Journal of Data Analysis and Information Processing. 07. 190-207. 10.4236/jdaip.2019.74012.

Built In, 2023. *Random Forest: A Complete Guide for Machine Learning.* [Online] Available at: https://builtin.com/data-science/random-forest-algorithm#procon [Accessed January 2024]

Career Foundry, 2023. *What is Random Forest.* [Online] Available at: https://careerfoundry.com/en/blog/data-analytics/what-is-random-forest/ [Accessed January 2024]

Drishti IAS 2020. *The Big Picture Census Challenges.* [Online] Available at: https://www.drishtiias.com/loksabha-rajyasabha-discussions/the-big-picture-census-challenges-importance [Accessed January 2024]

Forbes, 2023. *Why Skills Based Hiring Is On The Rise.* [Online] Available at: https://www.forbes.com/sites/carolinecastrillon/2023/02/12/why-skills-based-hiring-is-on-the-rise/ [Accessed January 2024]

IAS Gatewayy, 2020. *Census – Challenges & Importance.* [Online] Available at: https://iasgatewayy.com/census-challenges-importance/ [Accessed December 2024]

IBM, 2022. *What is Exploratory Data Analysis.* [Online] Available at: https://www.ibm.com/topics/exploratory-data-analysis [Accessed January 2024]

Investopedia, 2023. *Multicollinearity.* [Online] Available at: https://www.investopedia.com/terms/m/multicollinearity.asp [Accessed January 2024]

Madiedo, Juan & Chandrasekaran, Aravind & Salvador, Fabrizio [2019]. Capturing the Benefits of Worker Specialization: Effects of Managerial and Organizational Task Experience. Production and Operations Management.

Scikit-learn, 2020. *Decision Trees.* [Online] Available at: https://scikit-learn.org/stable/modules/tree.html [Accessed January 2024]

TechTarget, 2022. *Data Visualization.* [Online] Available at: https://www.techtarget.com/searchbusinessanalytics/definition/data-visualization [Accessed January 2024]

TowardsDataScience, 2018. *Accuracy, Precision, Recall or F1?* [Online] Available at: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 [Accessed January 2024]

Xu, JunYi. (2023). Systematic Analysis and Application Prospect of Decision Tree. Highlights in Science, Engineering and Technology. 71. 163-170. 10.54097/hset.v71i.12687.