CM2604 Machine Learning CW

Income predictions using census data

March 2024

Documented by: Abdul Shahid Mohamed Afham

IIT no: 20221246

RGU no: 2237030

Table of Contents

Project URL

https://github.com/Abdul-Shahid-2002/Income-prediction

Corpus preparation

The first step was to load the data

- The import using Python code was used for this.
- I preferred using the data in CSV format
- The adult metadata contained a link to the csv file
- I used the pandas library to load in the adult dataset in CSV format

Now that I had the data, the next step was to prepare the data for analysis and predictions

The first step in data preparation was to handle duplicates

- This is done to ensure data integrity so as to obtain trustworthy insights from the data
- The first step here was to make sure the dataset had duplicates and to get the count
- Once it was confirmed to have duplicated the next step was to remove all the duplicate rows from the dataset.

Check for structural errors in categorical variables

- It is natural to strange naming conventions, misspelled words or incorrect capitalizations when recording a dataset. This causes inconsistencies in the data, mainly this occurs in the categorical variables
- This was accomplished by grouping all the values in the categorical variables and getting their value counts to check if there were any unusual words.
- Also, replace the question mark values with nan since they both meant that the values were missing

Check for outliers

- Usually there are some off values in a data set that need to be validated in order to preserve the data integrity
- This was done by analyzing the numerical variables based on their distribution, and standard deviation and also visually by using some histogram plots

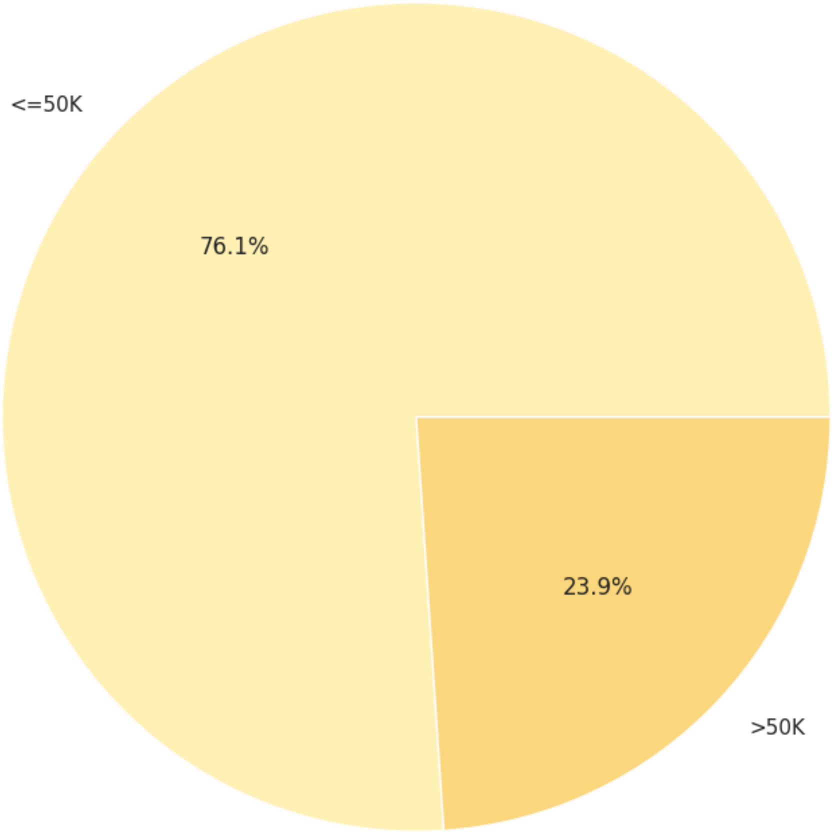
Handle missing values

- Missing values are common for any data set and there are countless ways to mitigate these errors
- The first step was to identify if missing value existed and then find a way to fix this issue
- The one followed here was by the nan or missing values with their mode or the most frequently occurring variable.

Solution methodology

Once the data is properly prepped it's time to get into analyzing and getting comfortable with the data.
The first step here was to understand the target variable. This was the “income” column in the table which contained four different valuers. Once cleaned it looked like this,

The value counts here were observed, visually using a pie chart results as follows

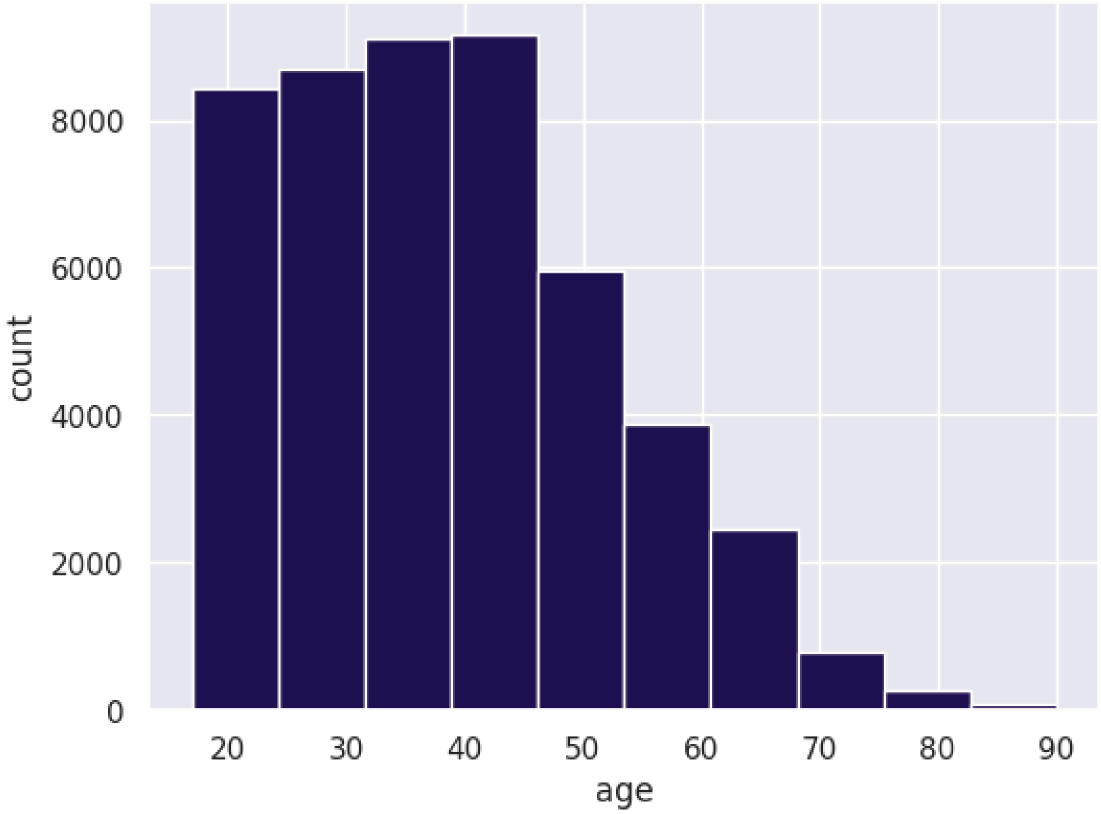


We can see that this is not a very good distribution of data for the model to learn patterns from.

Next univariates analysis was carried out on each of the columns on the table.

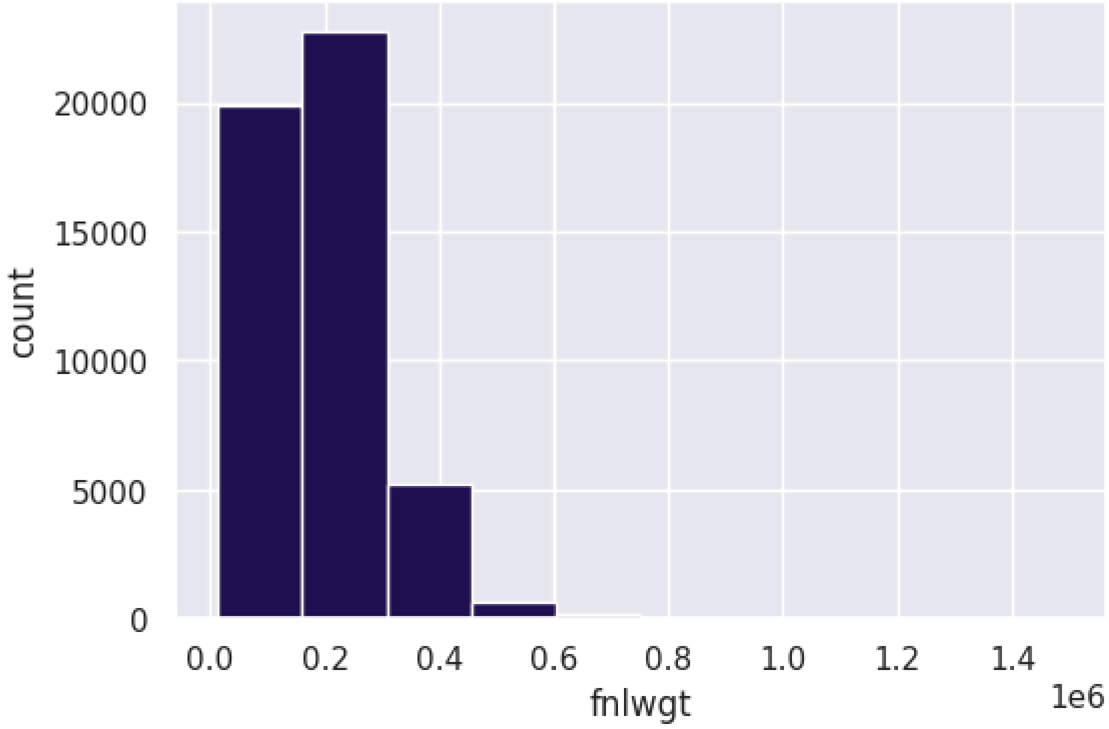
Univariate analysis

Age distribution



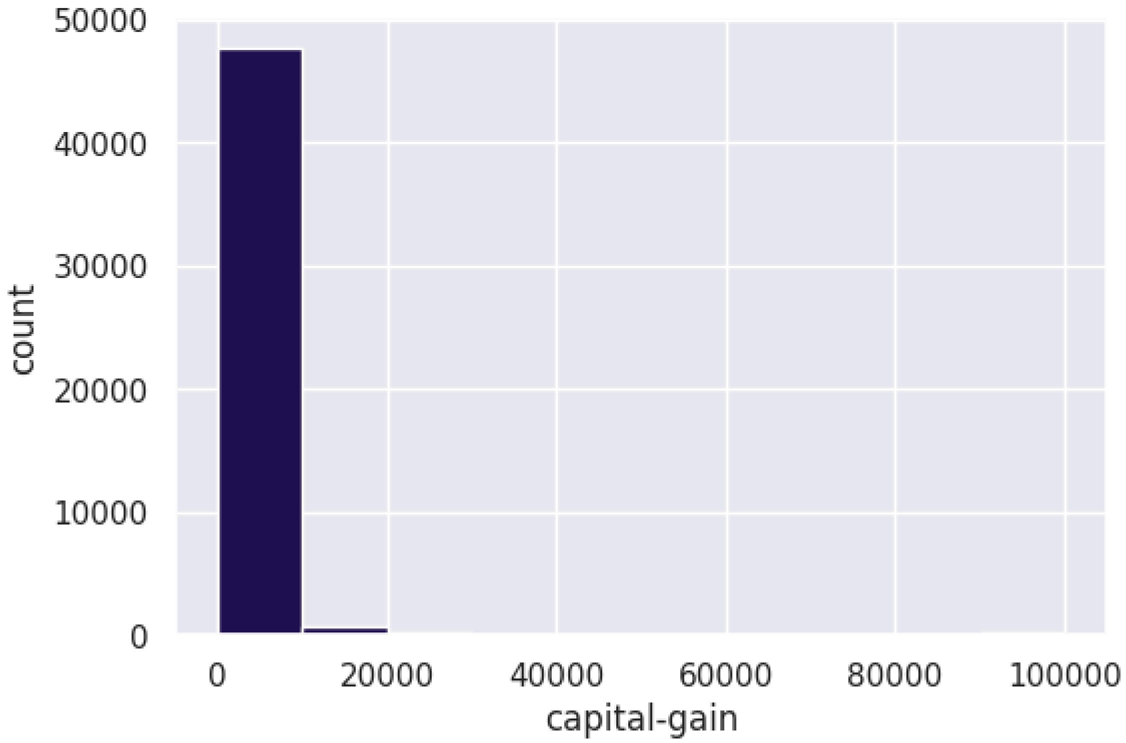
Rightly skewed not symmetric. The majority of the people are less than 50 years of age.

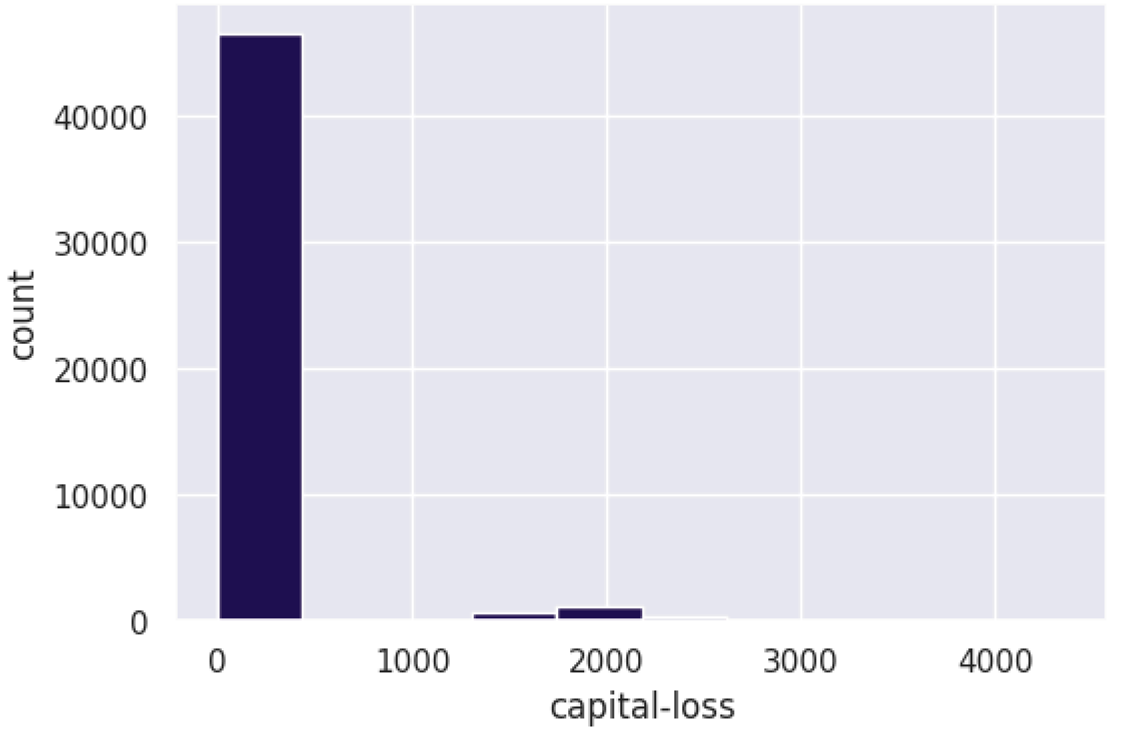
Final weight distribution



Really rightly skewed. Most of the values range from 0.0 to 0.4

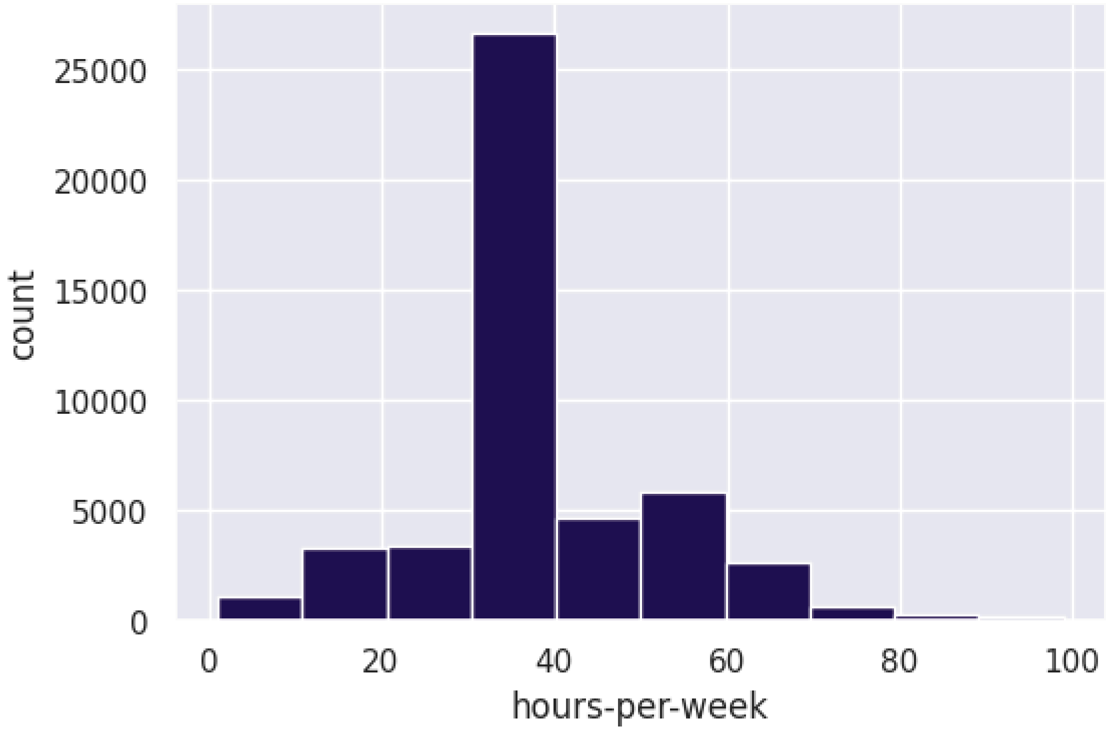
Capital gain and Capital loss distribution





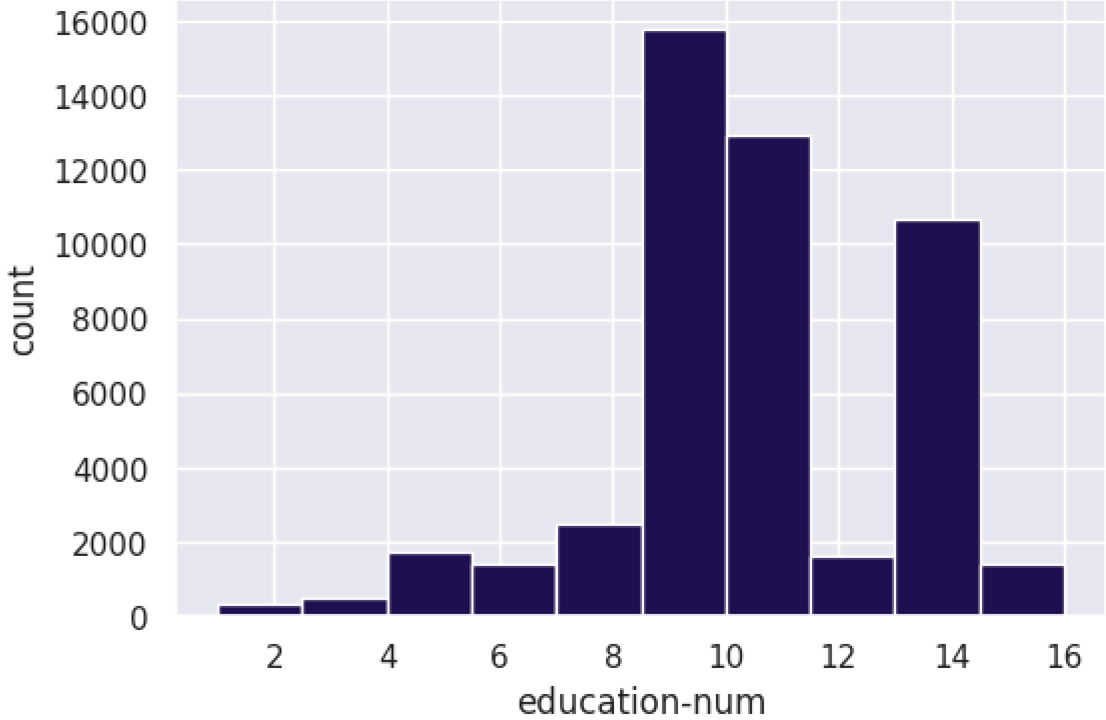
For both of these two features, more than 75% of the data they contain is zeros

Hours-per-week distribution



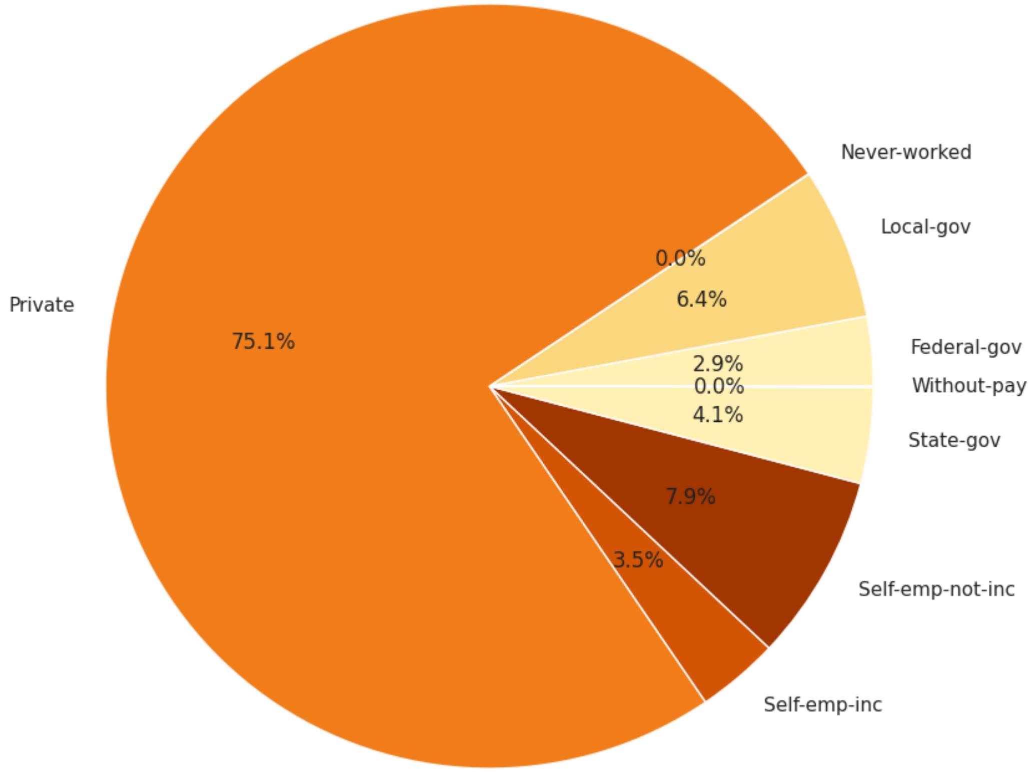
Slightly normally distributed.

Education number



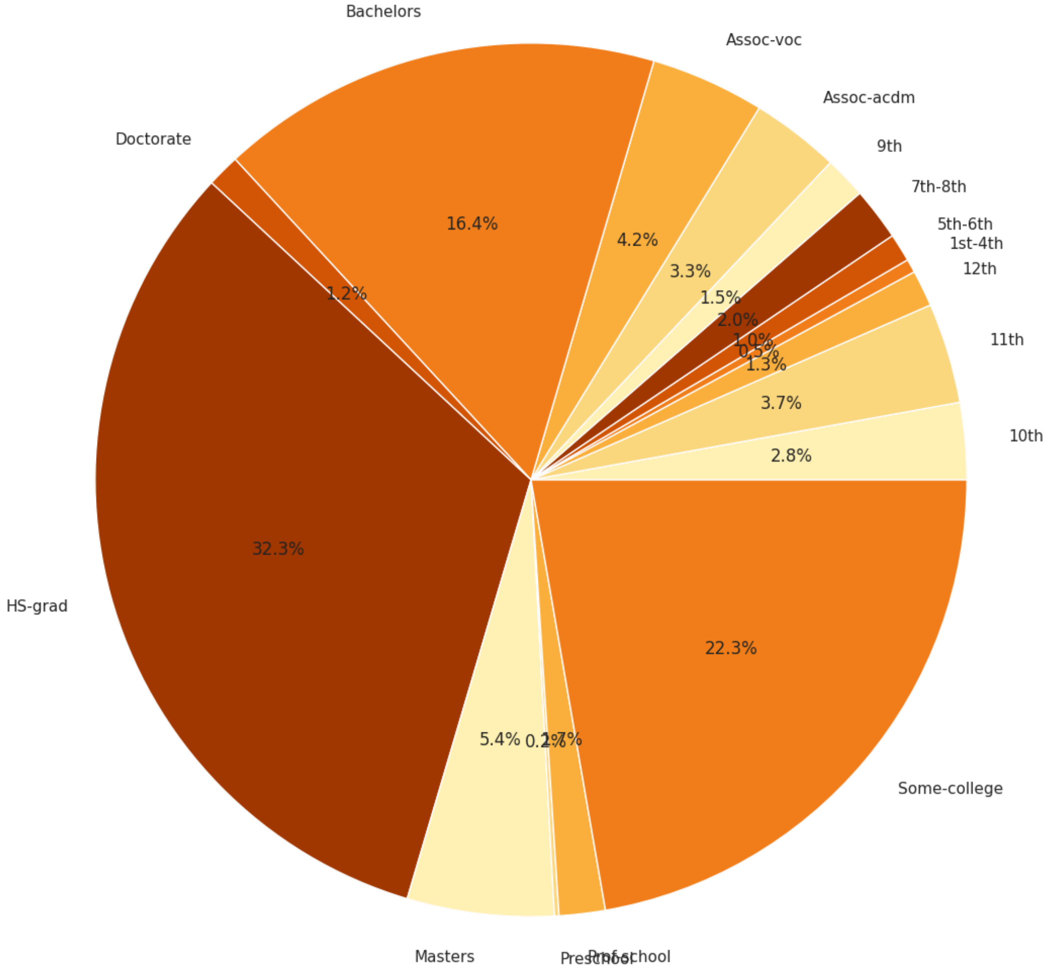
This too is slightly normally distributed around the number ten.

Workclass distribution



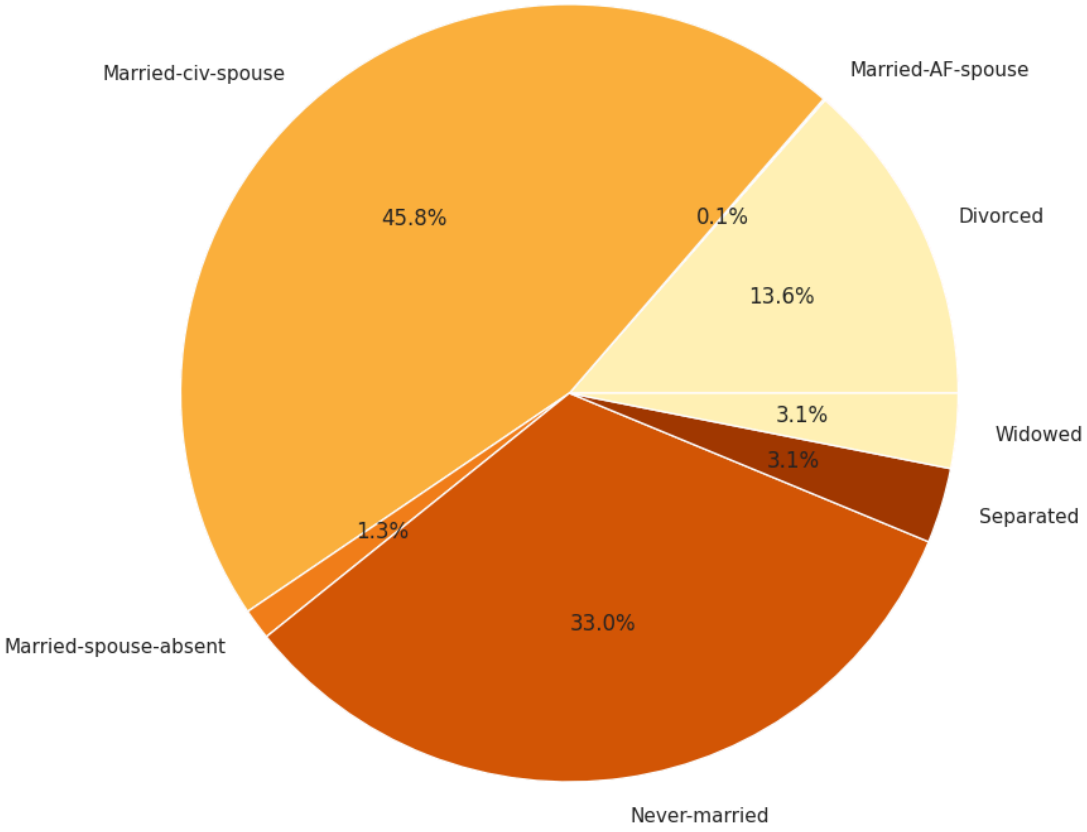
More than 75% of the people work in the private sector as shown above.

Education Distribution



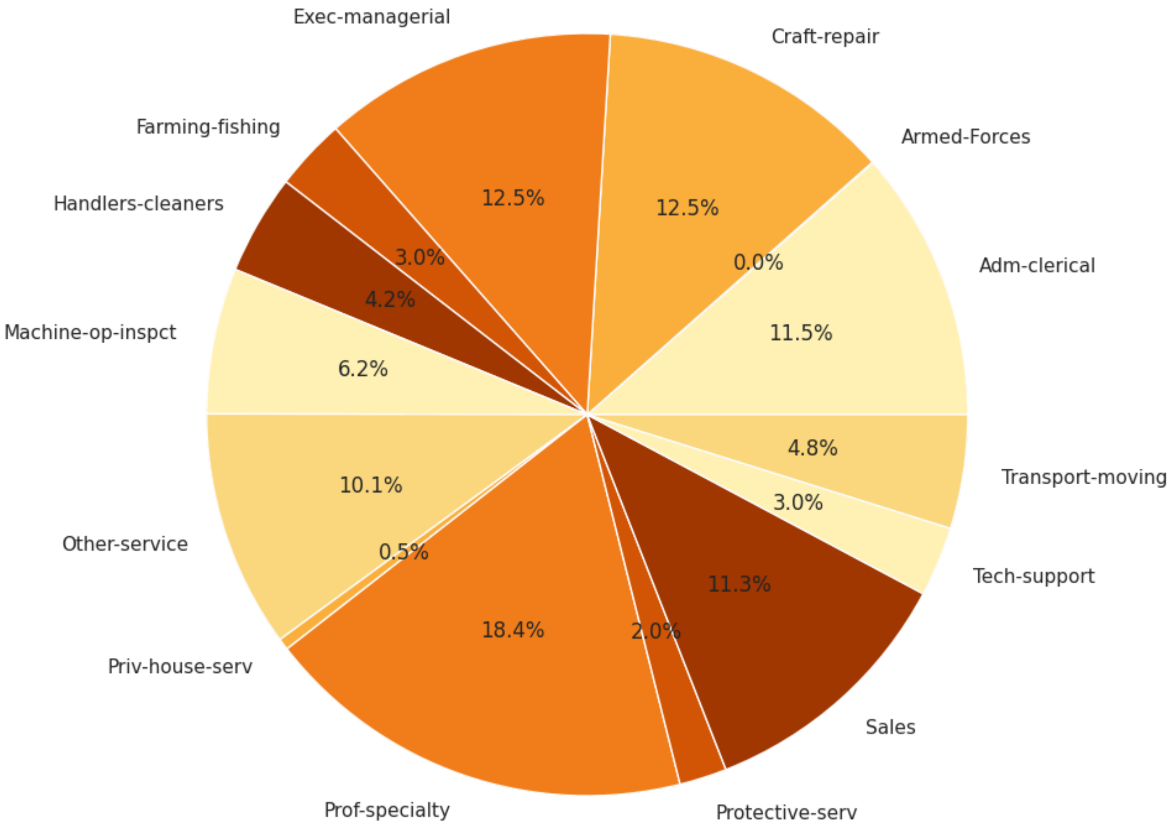
Most of the people have graduated from high school as the graph suggest followed by some college.

Marital-status distribution



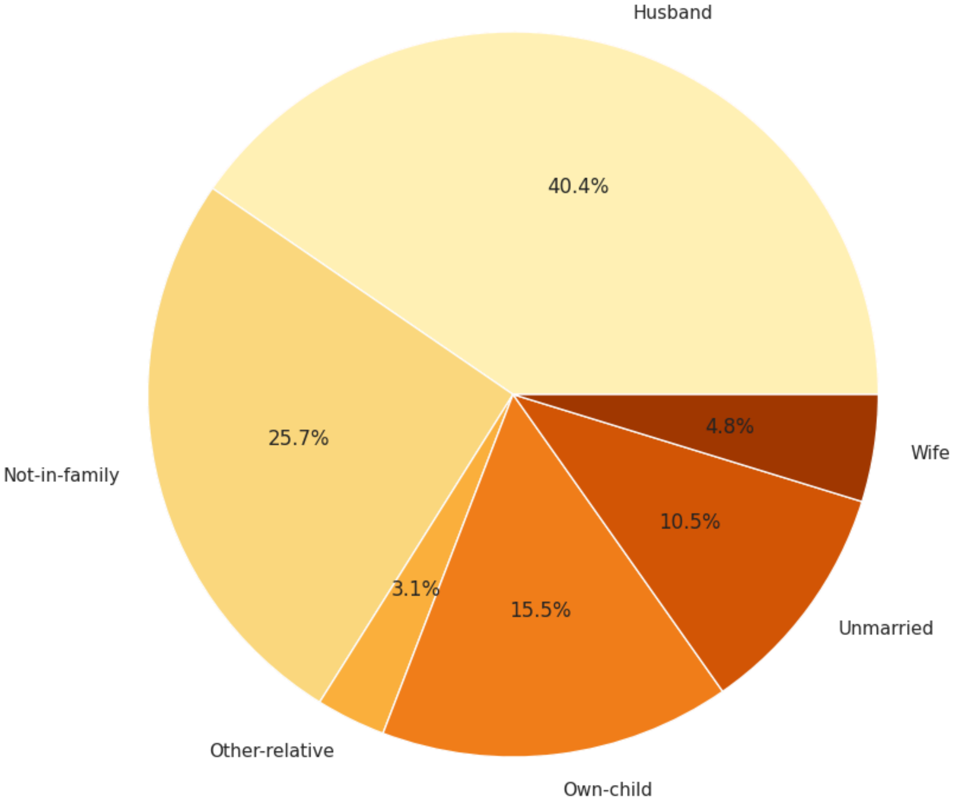
More than 45% of the data are Married civ spouse which is the highest and lowest being married af spouse at just 0.1%.

Occupation



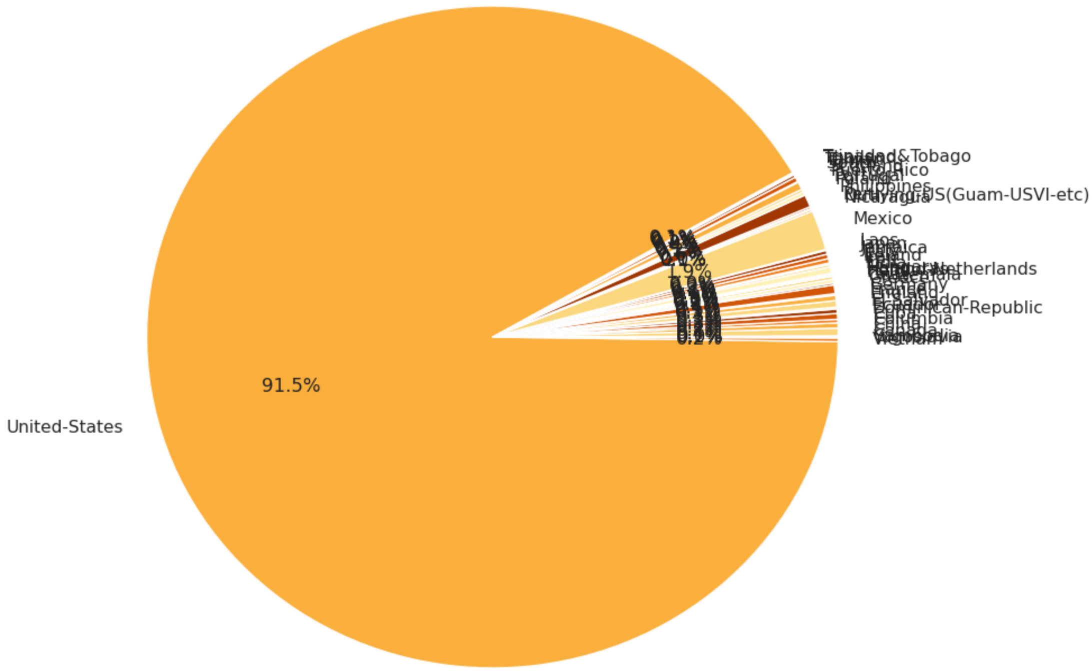
Here all the values are close to each other with the professional speciality occupation being the highest at 18.4%

Relationship



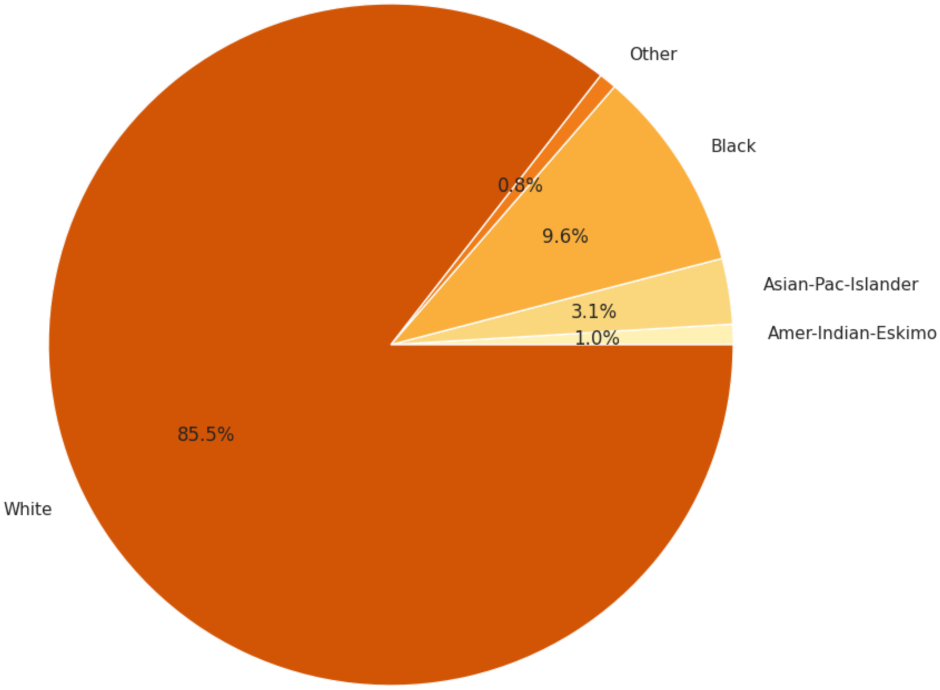
Most of the people here are husbands with a percentage of 40.4% while the percentage of wives in the dataset is 4.8%

Native country



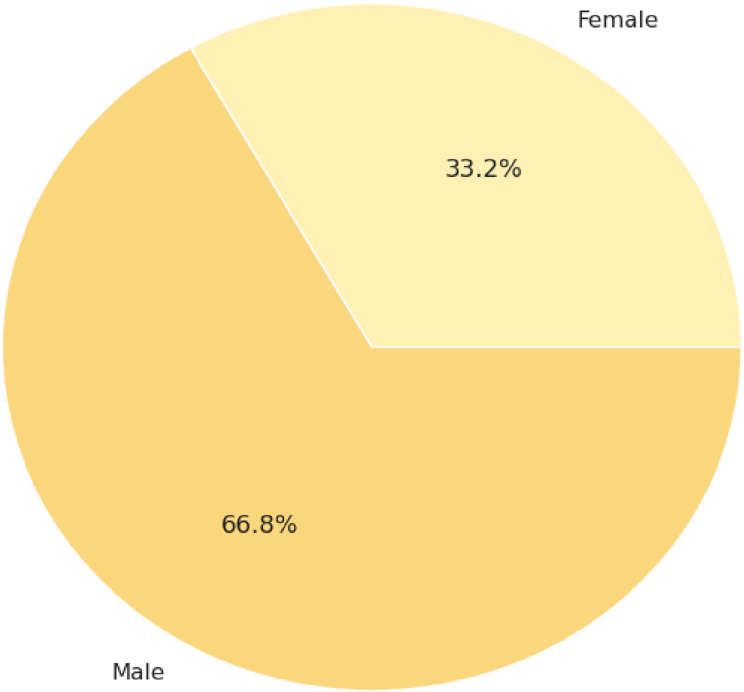
More than 90% of people in this dataset are from the United states.

Race distribution



The majority of data collected was from white people being more than 85% while 9.6% of the data was from black people

Sex distribution

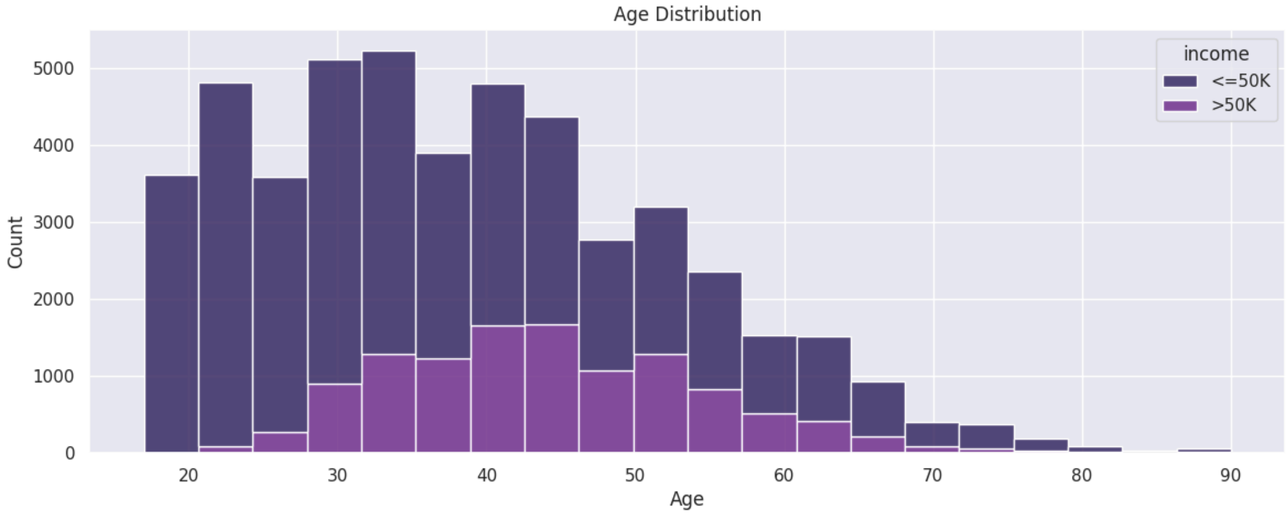


As shown above, 66.8% for males, and 33.2% were females

Now that we understand the columns individually, the next step is to see how each of them is related to the target variable

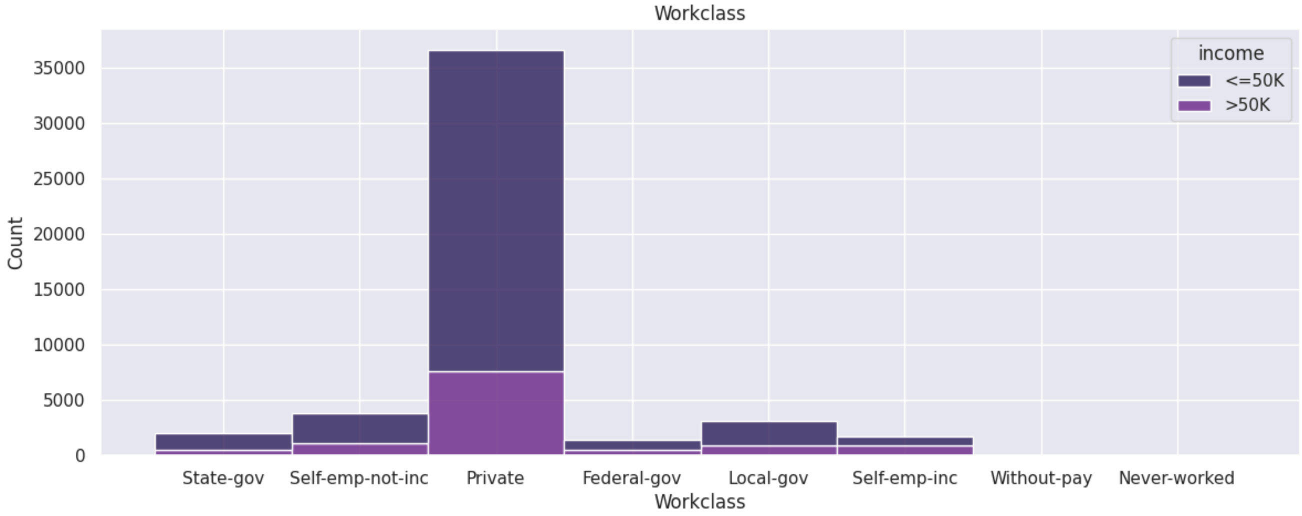
Bivariate analysis

Age and Income



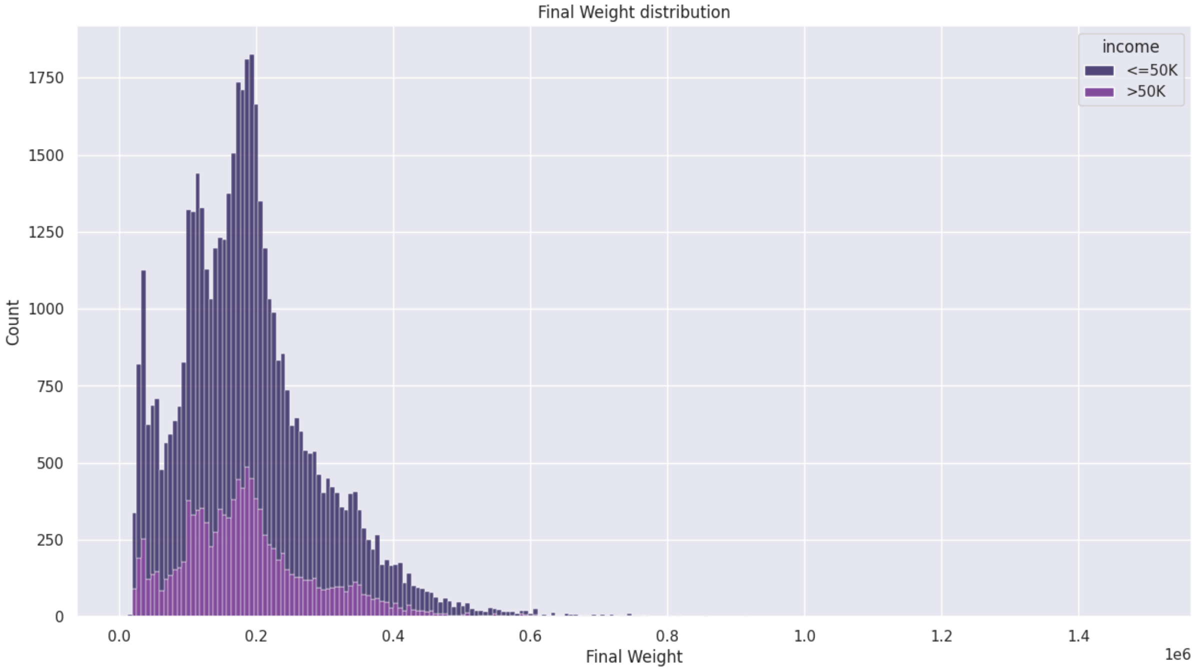
People start earning more than 50k once past 20, most of the people who earn more than 50k are in their forties.

Workclass and Income



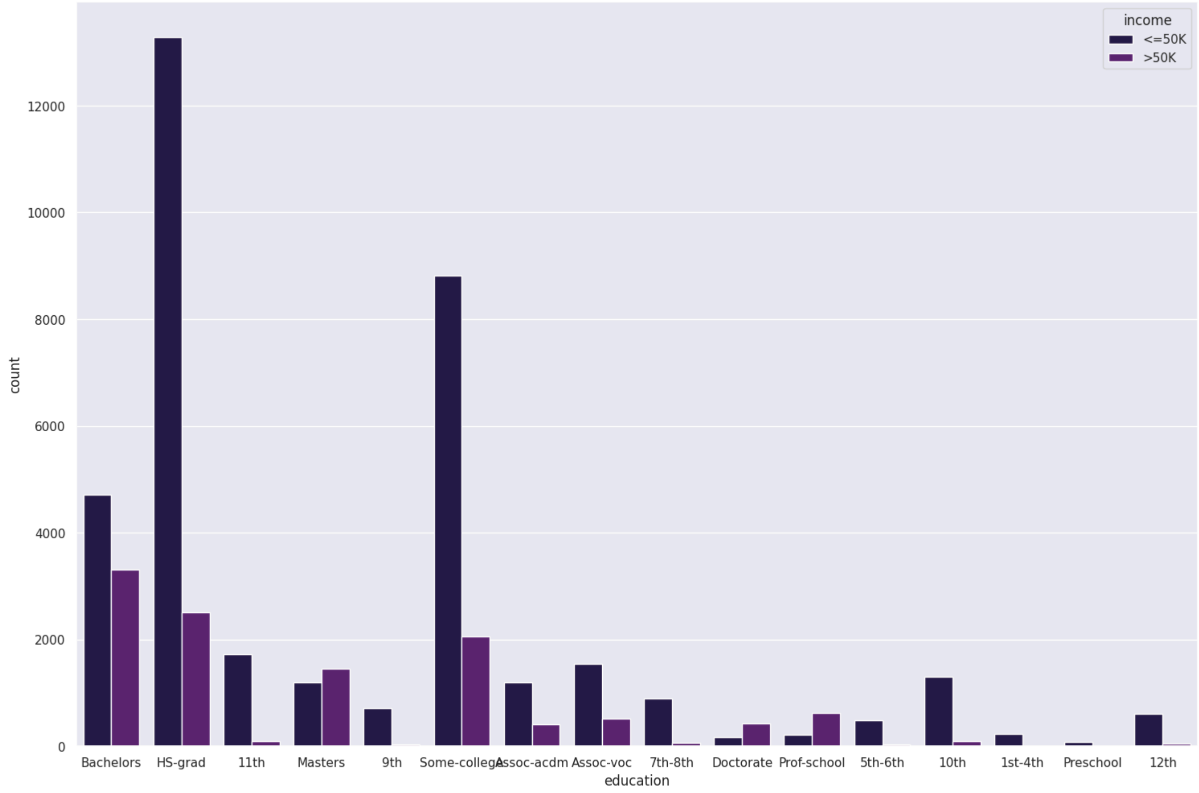
most of the people who earn more than 50k are in the private sector

Final Weight and Income



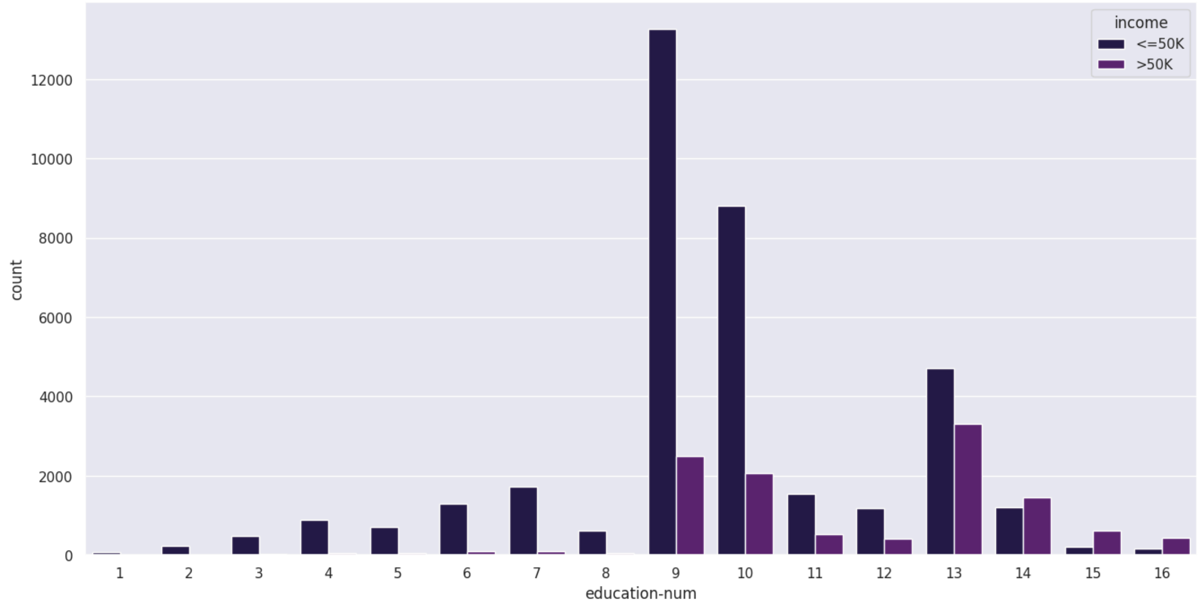
Shortly after the 0.2 mark the people who earn more than 50k die down slightly.

Education and Income



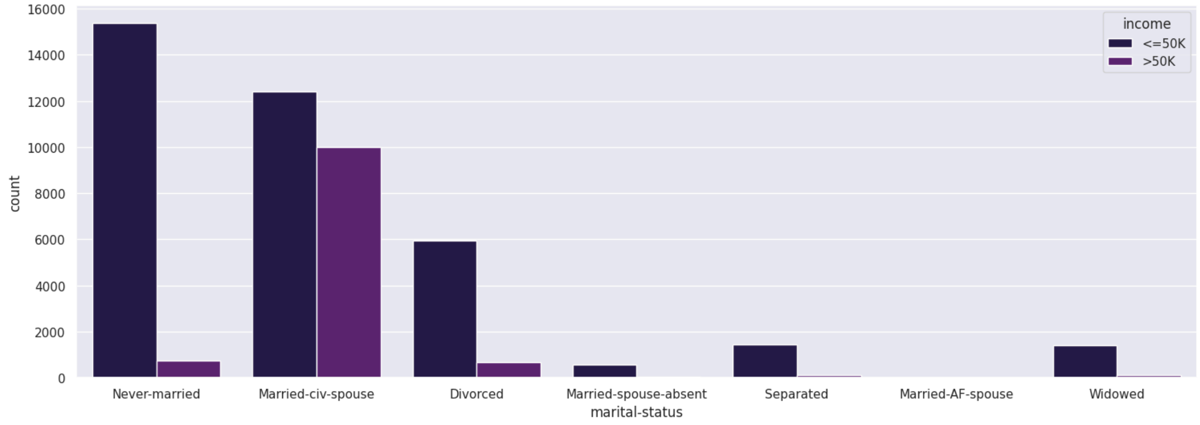
most of the people who earn more than 50k are HS grads, Bachelors or gone to some College

Education number and Income



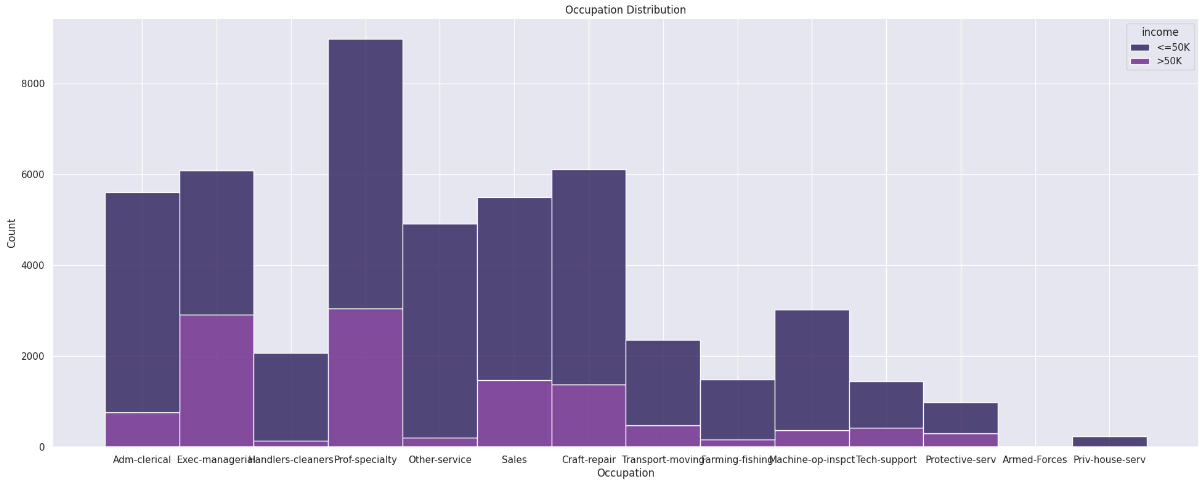
most of the people who earn more than 50k are from years 13,9 and 10.

Martial Status and Income



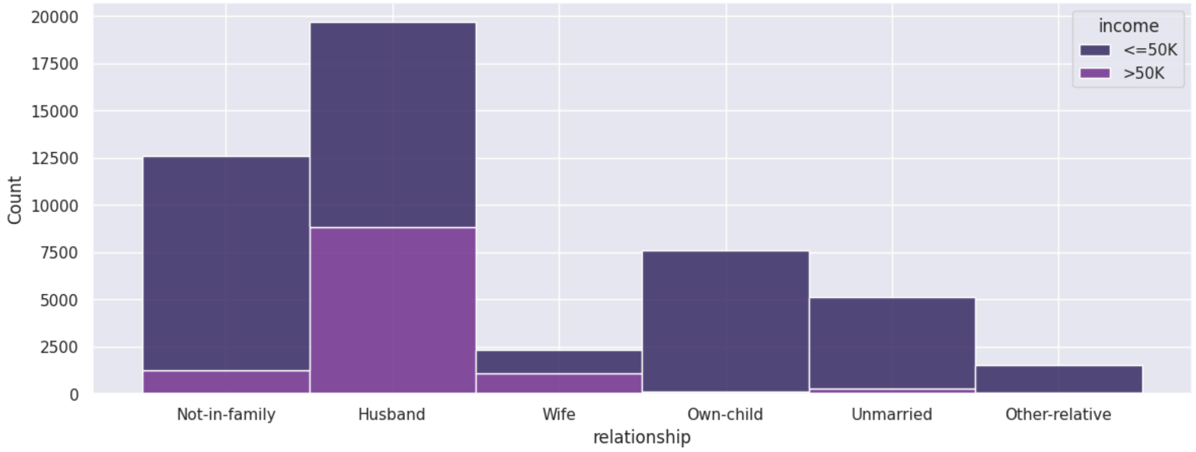
Marries civ spouse contains most of the people who earn more than 50k

Occupation vs Income



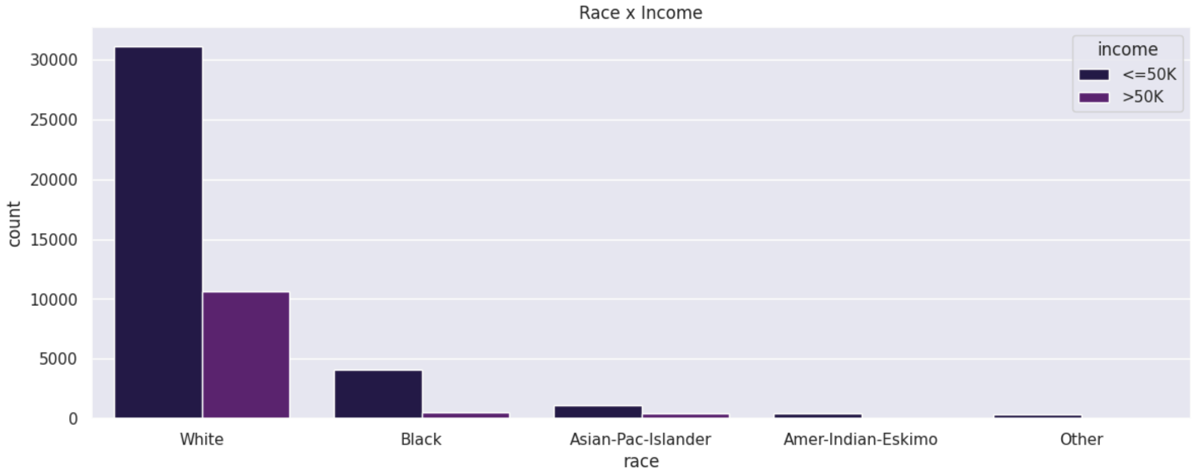
Exec management and prof specialty contain the bulk of the people who earn more than 50k

Relationship and Income



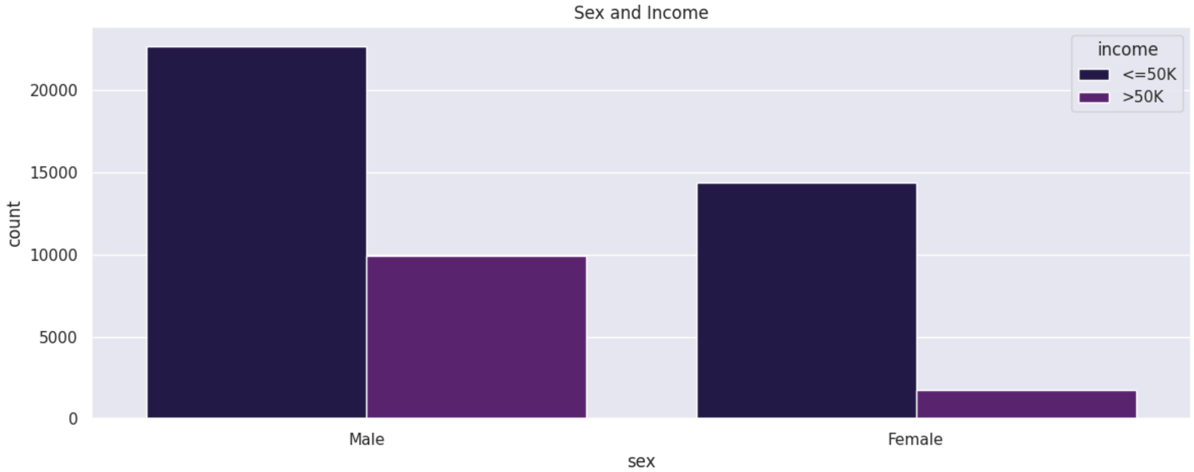
most of the people who earn more than 50k are either husbands or wives

Race and Income



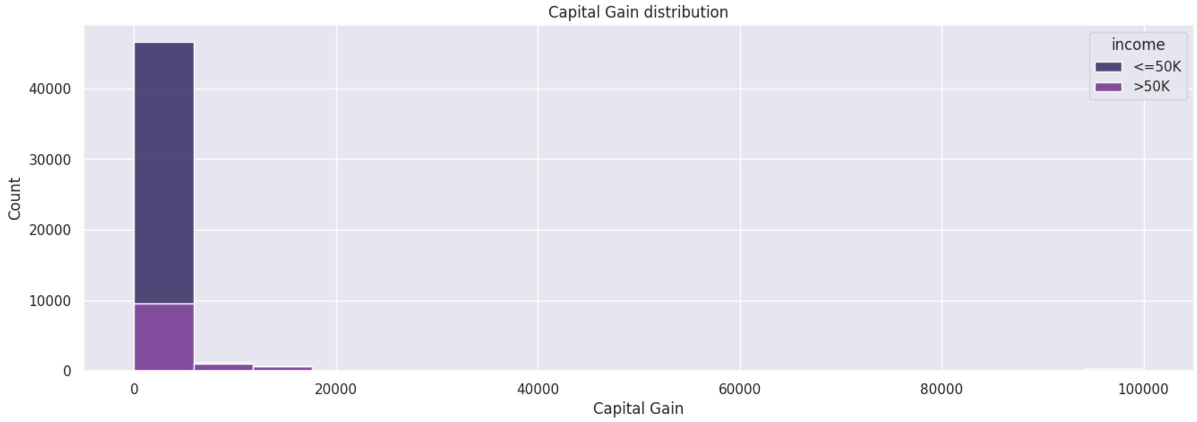
No surprise here since most of the data were collected from white folk

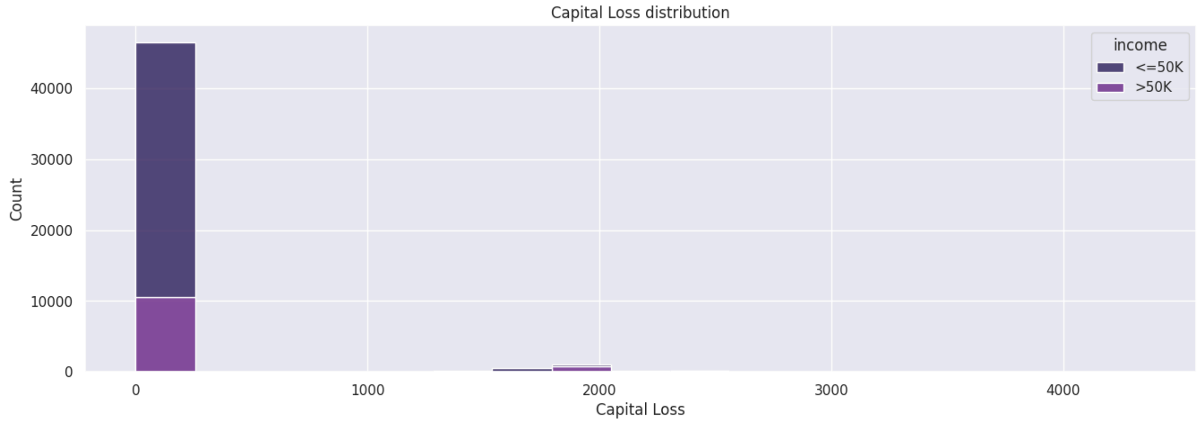
Sex and Income



Almost half of the males earn more than 50k

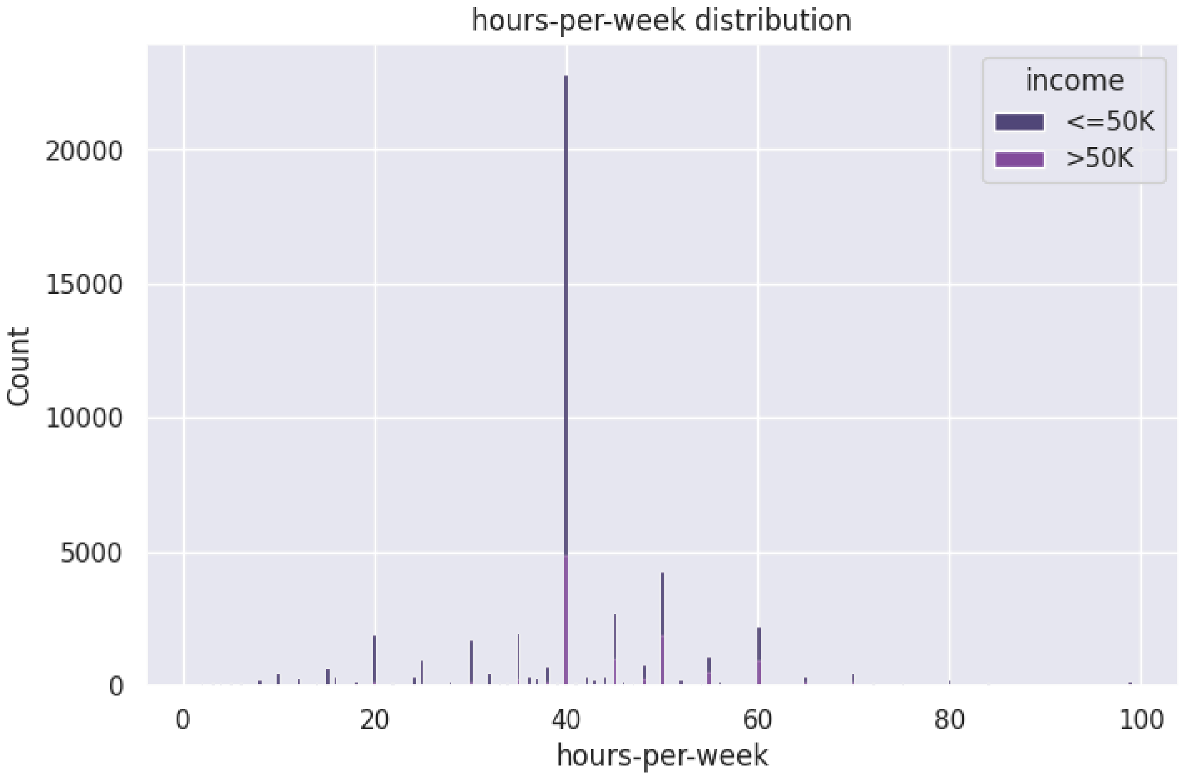
Capital gain and capital loss vs income





Capital gain and capital loss graphs do look like they contain the same results.

Hours per week and income



Here too since most of the people work 40 hours a week they hold the bulk of people who earn more than 50k too.

Model evaluation

The two models that were used to predict an individual's income were naive Bayes and random forest.

These models were evaluated according to the following criteria

- Accuracy score
- Mean squared error,
- F1 score

- Classification report
- Precision score
- Recall score
- Confusion matrix

Experimental results

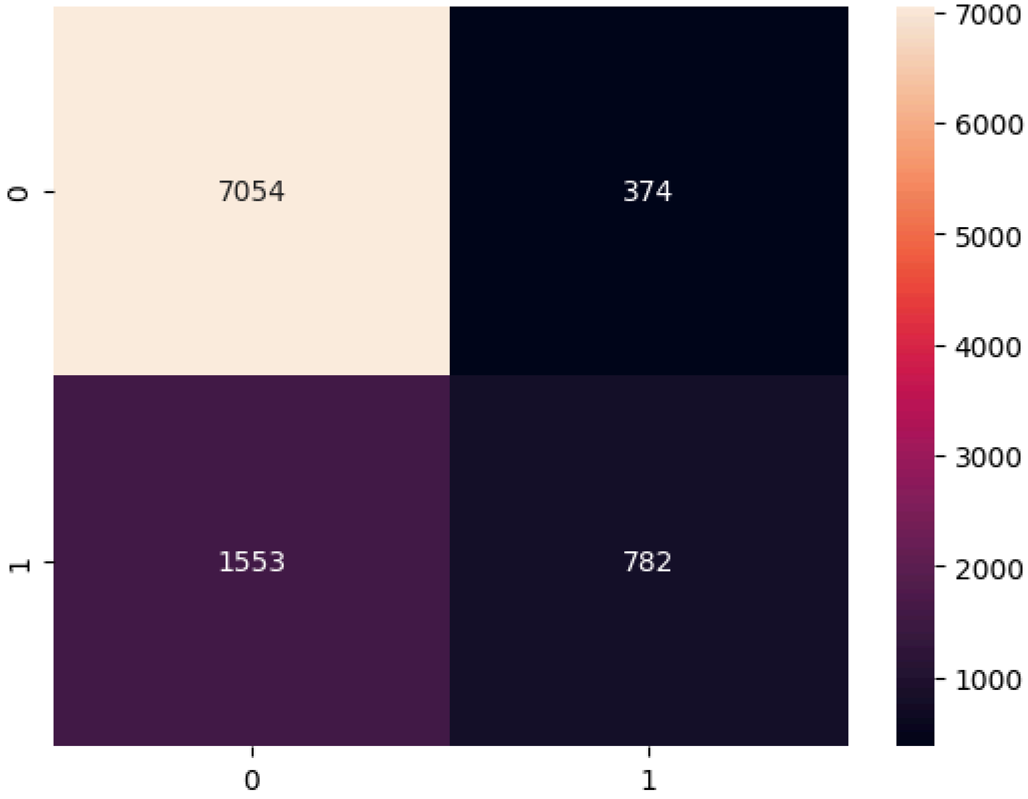
Naive Bayes

Accuracy score: 80.26%

Classification report:

	precision	recall	f1-score	support
0	0.82	0.95	0.88	7428
1	0.68	0.33	0.45	2335
Accuracy			0.80	9763
Macro avg	0.75	0.64	0.66	9763
Weighted avg	0.79	0.80	0.78	9763

Confusion matrix



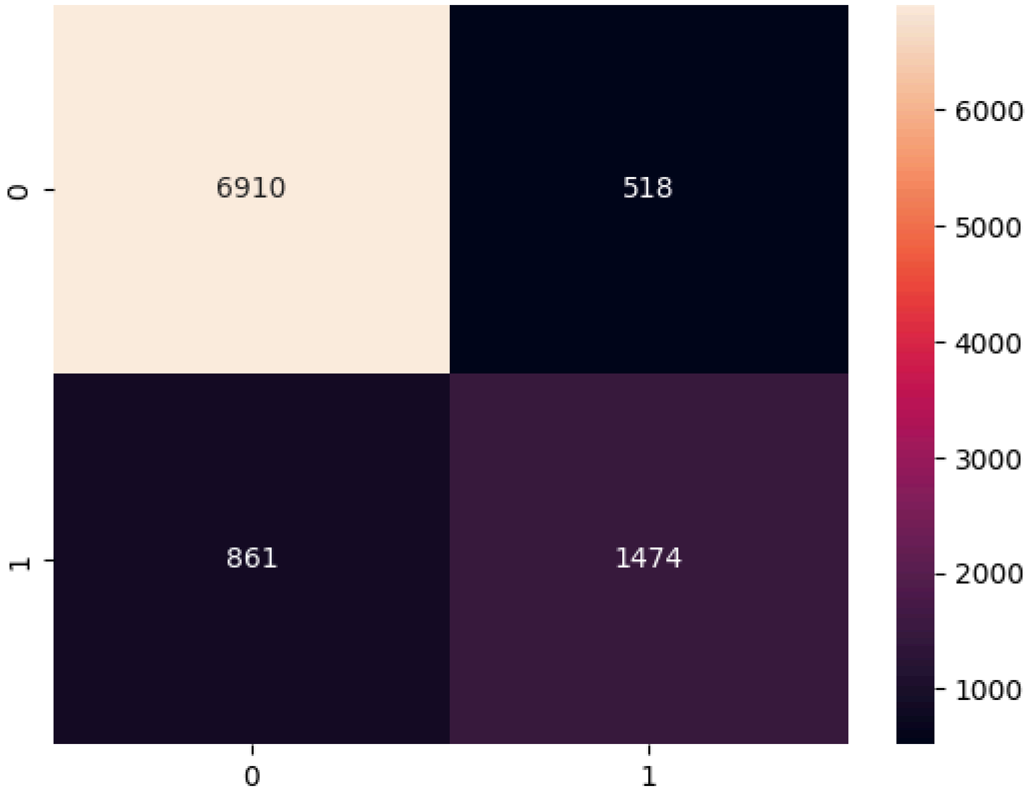
Random forest

Accuracy score: 85.87%

Classification report:

	precision	recall	f1-score	support
0	0.89	0.93	0.91	7428
1	0.74	0.63	0.68	2335
Accuracy			0.86	9763
Macro avg	0.81	0.78	0.80	9763
Weighted avg	0.85	0.86	0.85	9763

Confusion matrix



Limitations

- The target variables of the data didn't have even counts
- There were considerably more white people to blacks which may cause issues when predicting the income of someone black
- This model cant make predictions for people outside of the US since more than 90% of the data were of people from the US

Further enhancements

- Dimensionality reduction methods like PCA could have been used
- Other models like SVM or XGBoost must be performed to test if they outperform any of these two models
- Different encoders like One hot encoding could be tested on the categorical variables to check if the model accuracy increases

Appendix (Code)

```
!pip install ucimlrepo
```

```
import pandas as pd

url = "https://archive.ics.uci.edu/static/public/2/data.csv"

df = pd.read_csv(url)

# Handle duplicates
df.drop_duplicates(inplace=True)

import numpy as np
df.replace('?', np.nan, inplace = True)

df['workclass'] = df['workclass'].fillna(df['workclass'].mode()[0])
df['occupation'] = df['occupation'].fillna(df['occupation'].mode()[0])
df['native-country'] =
df['native-country'].fillna(df['native-country'].mode()[0])

df["income"].replace('<=50K.', "<=50K", inplace = True)
df["income"].replace('>50K.', ">50K", inplace = True)

print(df['income'].value_counts())
df['income'].value_counts().plot.pie(autopct= '%1.1f%%')

df['age'].hist()
sns.set_theme(style="darkgrid")
sns.set_palette("magma")
plt.xlabel('age')
plt.ylabel('count')

df['fnlwgt'].hist(figsize=(6,4))
plt.xlabel('fnlwgt')
plt.ylabel('count')

df['capital-gain'].hist(figsize=(6,4))
plt.xlabel('capital-gain')
plt.ylabel('count')

df['capital-loss'].hist(figsize=(6,4))
plt.xlabel('capital-loss')
plt.ylabel('count')

df['hours-per-week'].hist(figsize=(6,4))
plt.xlabel('hours-per-week')
plt.ylabel('count')
```

```
df['education-num'].hist(figsize=(6,4))
plt.xlabel('education-num')
plt.ylabel('count')

sns.set_palette("YlOrBr")
plt.figure(figsize=(16, 10))
plt.pie(df.groupby('workclass').size(),
labels=df.groupby('workclass').size().index, autopct='%1.1f%%')
plt.title('Workclass Distribution')

plt.figure(figsize=(19, 14))
plt.pie(df.groupby('education').size(),
labels=df.groupby('education').size().index, autopct='%1.1f%%')
plt.title('Education Distribution')

plt.figure(figsize=(16, 10))
plt.pie(df.groupby('marital-status').size(),
labels=df.groupby('marital-status').size().index, autopct='%1.1f%%')
plt.title('Marital-status Distribution')

plt.figure(figsize=(16, 10))
plt.pie(df.groupby('occupation').size(),
labels=df.groupby('occupation').size().index, autopct='%1.1f%%')
plt.title('Occupation Distribution')

plt.figure(figsize=(16, 10))
plt.pie(df.groupby('relationship').size(),
labels=df.groupby('relationship').size().index, autopct='%1.1f%%')
plt.title('Relationship Distribution')

plt.figure(figsize=(16, 10))
plt.pie(df.groupby('income').size(),
labels=df.groupby('income').size().index, autopct='%1.1f%%')
plt.title('Relationship Distribution')

plt.figure(figsize=(18, 10))
plt.pie(df.groupby('native-country').size(),
labels=df.groupby('native-country').size().index, autopct='%1.1f%%')

plt.figure(figsize=(12, 5))
plt.pie(df.groupby('race').size(), labels=df.groupby('race').size().index,
autopct='%1.1f%%')
plt.title('Race Distribution')
plt.figure(figsize=(8, 4))
plt.pie(df.groupby('sex').size(), labels=df.groupby('sex').size().index,
autopct='%1.1f%%')
plt.title('Sex Distribution')

##########################################
plt.figure(figsize=(14, 5))
sns.set_theme(style="darkgrid")
sns.set_palette("magma")
sns.histplot(data=df, x='age', hue='income', bins=20, multiple='stack')
plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Age Distribution')
plt.show()

plt.figure(figsize=(16, 8))
sns.histplot(data=df, x='workclass', hue='income', multiple='stack')
plt.xlabel('Workclass')
plt.title('Workclass')

plt.figure(figsize=(15, 8))
sns.histplot(x='fnlwgt', data=df, hue='income', multiple='stack')
plt.title('Final Weight distribution')
plt.xlabel('Final Weight')
```

```
df.groupby('education').size()
df.groupby('education-num').size()

plt.figure(figsize=(18, 12))
sns.countplot(x='education', data=df, hue='income')

plt.figure(figsize=(16, 8))
sns.countplot(x='education-num', data=df, hue='income')

print(df.groupby('marital-status').size())

plt.figure(figsize=(18, 6))
sns.countplot(data=df, x='marital-status', hue='income')

print(df.groupby('occupation').size())

plt.figure(figsize=(24, 9))
sns.histplot(data=df, x='occupation', hue='income', multiple='stack')
plt.xlabel('Occupation')
plt.title('Occupation Distribution')

print(df.groupby('relationship').size())

plt.figure(figsize=(14, 5))
sns.histplot(data=df, x='relationship', hue='income', multiple='stack')

print(df.groupby('race').size())

plt.figure(figsize=(14, 5))
sns.countplot(data=df, x='race', hue='income')
plt.title('Race x Income')

plt.figure(figsize=(14, 5))
sns.countplot(data=df, x='sex', hue='income')
plt.title('Sex and Income')

plt.figure(figsize=(16, 5))
sns.histplot(x='capital-gain', data=df, hue='income', multiple='stack')
plt.title('Capital Gain distribution')
plt.xlabel('Capital Gain')
plt.show()

plt.figure(figsize=(16, 5))
sns.histplot(x='capital-loss', data=df, hue='income', multiple='stack')
plt.title('Capital Loss distribution')
plt.xlabel('Capital Loss')

plt.figure(figsize=(8, 5))
sns.histplot(x='hours-per-week', data=df, hue='income', multiple='stack')
plt.title('hours-per-week distribution')
plt.xlabel('hours-per-week')

plt.figure(figsize=(8, 5))
sns.histplot(x='native-country', data=df, hue='income', multiple='stack')
plt.title('native-country distribution')
plt.xlabel('native-country')

###################################
from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()
for column in df.columns:
    if df[column].dtype == 'object':
        df[column] = le.fit_transform(df[column])

X = df.drop(['income'], axis=1)

y = df['income']

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, y, train_size=0.8,
random_state=43)

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

from sklearn.metrics import accuracy_score, mean_squared_error, f1_score,
classification_report, precision_score, recall_score, confusion_matrix
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier

# Naive Bayes Model
gnb = GaussianNB()
gnb.fit(x_train, y_train)
y_pred = gnb.predict(x_test)
```

```
print(‘* \n\n', accuracy_score(y_test, y_pred), '\n')
print(' \n\n', classification_report(y_test, y_pred), '\n')
print('')
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')

#Random Forest Classifier Model
rfc = RandomForestClassifier()
rfc.fit(x_train, y_train)
y_pred = rfc.predict(x_test)

print(' \n\n', accuracy_score(y_test, y_pred), '\n')
print(' \n\n', classification_report(y_test, y_pred), '\n')
print('’)
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt='d')
```