CSE578 DATA VISUALIZATION
Course Project Final Report

REETHU CHOWDARY VATTIKUNTA
Arizona State University
rvattiku@asu.edu | 1222619619

Business Objective and Goals

To increase the count of enrollment, the UVW College collaborated with XYZ Corporation to utilize their data to develop marketing profiles based on income as a key demographic. Creating marketing profiles for people who earn less than 50K and for people who earn more than 50K by utilizing the data from US Census Bureau by mainly based on attributes like age, occupation, gender, education level and marital status. To better target their marketing efforts, the UVW College marketing team is working to develop an application that can aggregate the variables used to develop their suggested model and forecast an individual's income based on input parameters.

As data analyst at XYZ Corporation, the primary goal is to analyze the various factors that affect an individual's salary. The aim of the analysis is to assist UVW College in enhancing their admissions strategy, as they have recognized that an individual's compensation is a significant factor in promoting their degree programs.

Assumptions

- Accuracy

It is assumed that the information provided to us is reliable and accurate. We believe that the dataset is an accurate representation of the population and assume no responsibility for any errors within the dataset. Our analysis assumes that the sample population is an accurate representation of the overall population.

- Attribute

Another assumption is that salary is a significant demographic that plays a vital role in the selection of other criteria.

- Timely accuracy

We assume that the information presented to us is both current and relevant. It is that the sample data provided accurately reflects the real population, and that the information presented is consistent throughout.

User Stories

1. **SEX**: As a member of marketing team, I require the analysis of whether an individual's sex is a significant factor in determining their income. This analysis will enable us to include this factor in income prediction models, improving the accuracy of our predictions.
2. **NUMBER of HOURS:** To determine if the number of hours worked per week is a significant factor in predicting income. Therefore, I require an analysis to establish if the correlation between hours per week and income is strong enough to justify including it in my prediction model.
3. **EDUCATION & AGE:** To determine if a person's education level and age, when considered together, have a significant impact on their income. I need an analysis to establish if the correlation between education level, age, and income is strong enough to justify including them as factors in my prediction model.
4. **RACE**: To determine if an individual's race is a significant factor in predicting their income. Therefore, I require an analysis to establish if there is a correlation between race and income that is strong enough to justify including it as a factor in my prediction model.
5. **RELATIONSHIP STATUS & AGE:** To determine if an individual's relationship status

and age, when considered together, have a significant impact on their income. I require an analysis to establish if the correlation between relationship status, age, and income is strong enough to justify including them as factors in my prediction model.

Visualizations

User story 1: Sex w.r.t Income

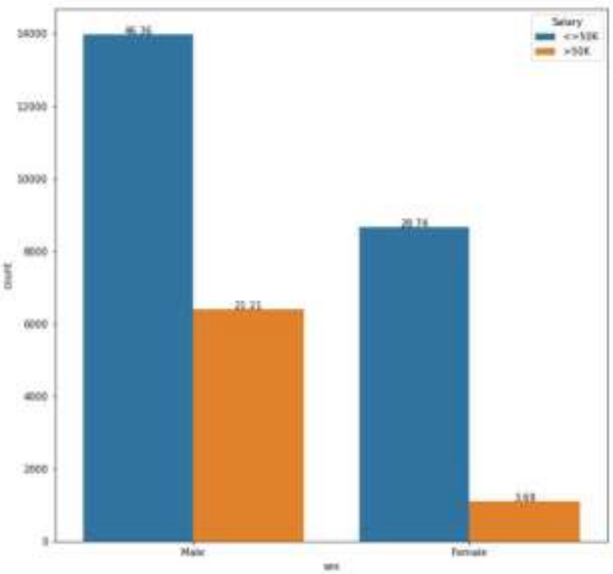

Fig 1: Count plot of sex corresponding to salary

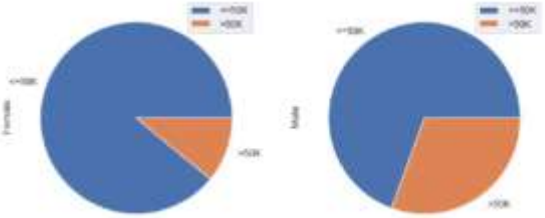

Fig 2: Pie chart of gender w.r.t salary

The charts represented by Figures 1 and 2 showcase the relationship between gender (Male, Female) and income levels. It is observed that a significant proportion (47%) of the total number of individuals who earn less than 50k per year belong to the male category. In contrast, only a small fraction (3%) of the total number of individuals who earn more than 50k per year belong to the female category.

Furthermore, the charts indicate that there are more individuals earning less than 50k per year than those earning more than 50k per year. The data also reveals that males generally receive higher pay than females. Overall, these findings highlight the gender-based disparities in income levels, which is an issue that needs to be addressed to achieve equal opportunities and eliminate discrimination in the workforce.

User Story 2: Hours w.r.t Income

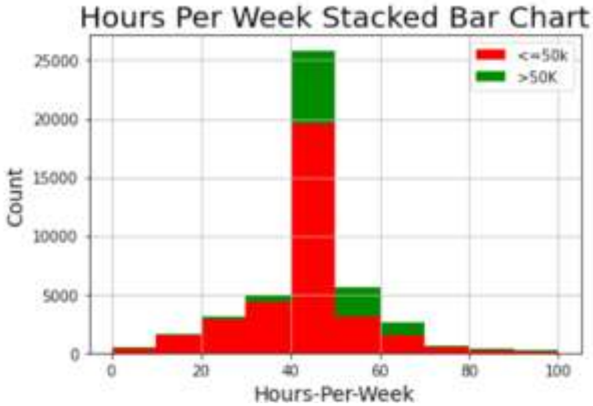

Fig 3: Stacked bar chart of hours-per-week vs Income

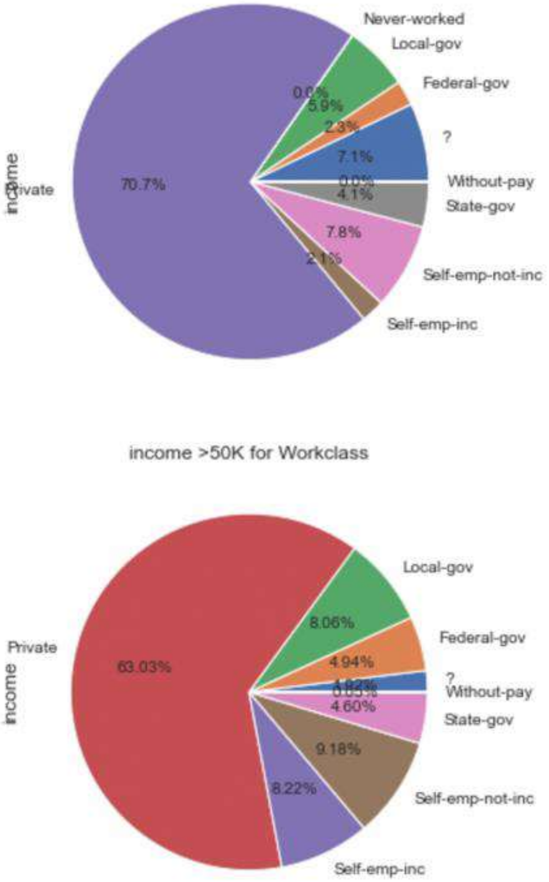

Fig 4: Pie chart of Hours per week and Income

Figures 3 and 4 demonstrate a stacked bar chart that illustrates the distribution of incomes based on the frequency count of individuals. The data indicates that most workers typically work between 40 to 50 hours per week, regardless of their income levels. However, the chart also reveals that individuals with higher incomes tend to work longer hours, with those earning more than $50,000 per year putting in more hours than those earning less than 50k per year.

It is evident that individuals earning less than 50k per year tend to work fewer hours, typically ranging between 30 to 50 hours per week, as indicated by the right-skewed distribution. On the other hand, individuals earning more than 50k per year tend to work significantly more than 40 hours per week, as shown by the left-skewed distribution. The data further indicates that the amount of income earned is often correlated with the number of hours worked. Those who work longer hours tend to earn more income, which is reflected in the left-skewed distribution of individuals with higher incomes.

In summary, these findings highlight the relationship between income and weekly working hours, indicating that those who work longer hours tend to earn higher incomes.

Usage Story 3: Education and age with respect to income

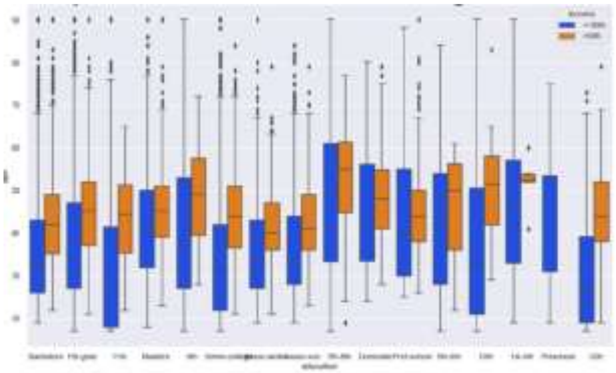

Fig 5: Box plot for attributes Education age vs Income

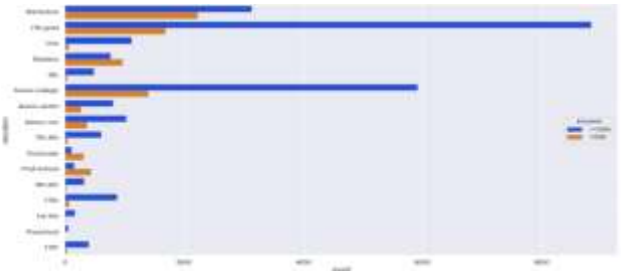

Fig 6: Count plot of education w.r.t age

The boxplot graph shown in Figure 7 illustrates the relationship between education and age with respect to income levels. One of the significant observations is that the median age of individuals earning an income greater than 50k per year is higher than those earning less than 50k per year. This suggests that as individuals grow older, they tend to earn a higher income. The graph also highlights that the majority of the population belongs to the income group earning less than 50k per year. Additionally, the data shows that the highest median age for individuals earning more than 50k per year is in the 7-8th education category. On the other hand, the highest median age range for individuals earning less than 50k per year is also in the 7-8th education category, while the lowest median age range for this income group is in the 11th class.

Overall, the boxplot graph showcases the relationship between education, age, and income levels. It provides valuable insights into the trends and patterns associated with individuals' income based on their education and age, highlighting the importance of education and experience in determining earning potential.

Usage Story 4: Race w.r.t Income

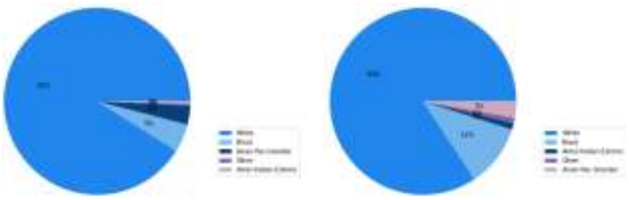

Figure 7: Pie chart of race w.r.t income >50k and <50k

The data presented in Figure 9 indicates that many individuals earning an income above 50k per year are white, followed by black and Asian individuals. Specifically, 91% of individuals in the higher income bracket are white. On the other hand, most individuals earning an income below 50k per year are also white, followed by black and Asian individuals. In this group, 84% of individuals are white.

This information provides insights into the racial disparities that exist in the distribution of income levels. The data indicates that there is a significant difference in the proportion of white individuals compared to black and Asian individuals earning above and below 50k per year. This suggests that there may be systemic and institutional factors that contribute to these disparities. Overall, this data highlights the need for further investigation and efforts to address these disparities and promote more equitable distribution of income across different racial groups.

User Story 5: Relationship and age w.r.t Income

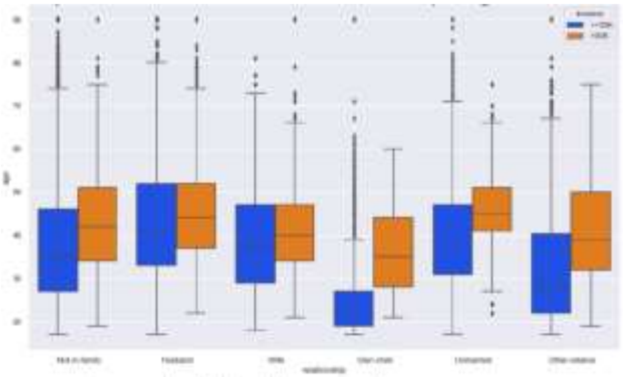

Fig 8: Box plot of relationship w.r.t age

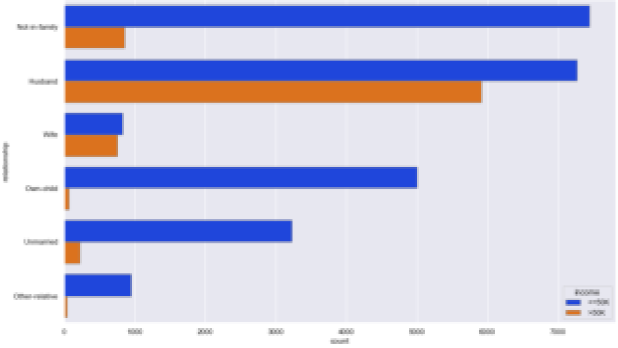

Fig 9: Count plot of age corresponding to relationship

Figure 8, presents a box plot graph that displays the relationship status of individuals with respect to their age. The plot also shows the outliers present in each category. The graph indicates that the mean age is higher for individuals earning an income above 50k per year. Additionally, the husband and not-in-family categories have the highest median age for those earning above 50k per year, whereas the own-child category has the lowest median age. Similarly, for individuals earning less than 50k per year, the husband category has the highest median age, and the own-child category has the lowest median age.

In contrast, Figure 9 displays a count plot of the age distribution for different relationship categories based on their income levels. The plot reveals that the not-in-family relationship category earns less than 50k per year and comprises most of the total individuals, whereas the own-child and other relative categories earning above 50k per year are the minority. Furthermore, the plot indicates that the total count of individuals earning less than 50k per year is higher than those earning more than 50k per year.

These findings provide valuable insights into the relationship between age, relationship status, and income levels, highlighting the need for further investigation and efforts to address income disparities across different relationship categories.

Data Preparation and Data Cleaning Code

```
# read adult.data and adult.names using
pandas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

import os
import warnings
warnings.filterwarnings('ignore')

df = pd.read_csv('adult.data',
header=None)
# print first lines of data
```

```
print(df.head())

with open('adult.names', 'r') as f:
    lines = f.readlines()
# get the headers from the file
df_headers = pd.DataFrame(lines)

df_headers = df_headers[0].str.split(':',
expand=True)
df_headers = df_headers[0].str.replace('-
', '_')
df_headers = df_headers.str.strip()
df_headers = df_headers.str.lower()
df_headers = df_headers.str.replace(' ',
'_')
headers = df_headers.values.tolist()
headers.append('income')
df.columns = headers
# print first lines of data
print(df.head())

df.to_csv('adult.csv', index=False)
print(df.head())
```

Data Cleaning

```
# read adult.csv using pandas and clean
the data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import warnings
warnings.filterwarnings('ignore')
# read the data
df = pd.read_csv('adult.csv')
# print first lines of data
print(df.head())
# print the shape of the data
print(df.shape)
# print the data types of the data
print(df.dtypes)
```

```
# print the number of missing values in
each column
print(df.isnull().sum())
# count number of "?" in each column
print("count of ?", df.eq('?').sum())
# cleaning the data by replacing "?" with
with either average or mode or most
frequent value
#for column  age replace ? with mean
df['age'] = df['age'].replace('?',
df['age'].mean())
#for column  workclass replace ? with
mode
df['workclass'] =
df['workclass'].replace('?',
df['workclass'].mode()[0])
#for column  fnlwgt replace ? with mean
df['fnlwgt'] = df['fnlwgt'].replace('?',
df['fnlwgt'].mean())
df['occupation'] =
df['occupation'].replace('?', 'Prof-
specialty')
df['native_country'] =
df['native_country'].replace('?',
'United-States')
# remove rows with income = '?'
df = df[df['income'] != '?']
# fill rest coulmns with mode
df['education'] =
df['education'].replace('?',
df['education'].mode()[0])
df['marital_status'] =
df['marital_status'].replace('?',
df['marital_status'].mode()[0])
df['occupation'] =
df['occupation'].replace('?',
df['occupation'].mode()[0])
df['relationship'] =
df['relationship'].replace('?',
df['relationship'].mode()[0])
# print(df.eq('?').sum())  checked
# data cleaning done
print(df.head())

# grouping
# group by income csv files
# make a csv files for each income group
# group by income
grouped = df.groupby('income')
# print the number of rows in each group
print(grouped.size())
# make a csv files for each income group
print(df['income'].unique())

grouped.get_group('
<=50K').to_csv('less_income.csv',
index=False)
grouped.get_group('
>50K').to_csv('more_income.csv',
index=False)
# print the number of rows in each group
print(grouped.size())

# data visualization
# plot the histogram of age
df['age'].hist()
plt.show()
# plot the histogram of education
df['education'].hist()
plt.show()
# plot the histogram of marital_status
df['marital_status'].hist()
plt.show()
# plot the histogram of occupation
df['occupation'].hist()
plt.show()
# plot the histogram of relationship
df['relationship'].hist()
plt.show()
```

AGE & RELATIONSHIP

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
# Read the CSV file and assign column
names to the dataframe
df = pd.read_csv("adult.data",
header=None)
df.columns = ["age", "workclass",
"fnlwgt", "education", "education-num",
"maritalstatus",
              "occupation",
"relationship", "race", "sex", "capital-
gain", "capital-loss",
              "hours-per-week", "native-
country", "income"]

# Create a box plot comparing the age
distribution for different relationship
statuses and income groups
sns.set(rc={'figure.figsize':(15,9)})
sns.boxplot(x='relationship', y='age',
data=df, palette='bright', hue='income')
plt.title("Box plot for attributes -
education age Vs Income", fontsize=40,
color="Green")

# Create a count plot showing the number
of individuals with different ages for
different income groups
sns.set(rc={'figure.figsize':(15,9)})
sns.countplot(y='age', hue='income',
data=df, palette='bright',
edgecolor='.6')
```

AGE AND EDUCATION

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from collections import Counter

# Load dataset
```

```
df = pd.read_csv("adult.data",
header=None, names=["age", "workclass",
"fnlwgt", "education", "education-num",

          "maritalstatus", "occupation",
"relationship", "race", "sex",

          "capital-gain", "capital-loss",
"hours-per-week", "native-country",

          "income"])

# Get counts for income groups
above_count =
df["income"].value_counts()[">50K"]
below_count =
df["income"].value_counts()["<=50K"]

# Analyze numerical data
def analyze_numerical_data(column):
    above_50k_data = df[df["income"] ==
">50K"][column]
    below_50k_data = df[df["income"] ==
"<=50K"][column]
    print(column)
    print("Mean")
    print("Above 50K = " +
str(above_50k_data.mean()))
    print("Below 50K = " +
str(below_50k_data.mean()))
    print("Median")
    print("Above 50K = " +
str(above_50k_data.median()))
    print("Below 50K = " +
str(below_50k_data.median()))
    print("Standard Deviation")
    print("Above 50K = " +
str(above_50k_data.std()))
    print("Below 50K = " +
str(below_50k_data.std()))
    sns.boxplot(x="income", y=column,
data=df)
    plt.title("Box plot for attributes -
"+column+" Vs Income", fontsize=14,
color="Green")
    plt.show()
    sns.histplot(data=df, x=column,
hue="income", kde=True, multiple="stack",
edgecolor=".6", linewidth=.6,
                 palette="bright")
    plt.title("Histogram for attributes -
"+column+" Vs Income", fontsize=14,
color="Green")
    plt.show()

# Analyze categorical data
def analyze_categorical_data(column):
    above_50k = df[df["income"] ==
">50K"][column].value_counts()
    below_50k = df[df["income"] ==
"<=50K"][column].value_counts()
    plt.pie(above_50k,
labels=above_50k.index,
autopct='%1.0f%%')
    plt.title("Distribution of "+column+"
for >50K", fontsize=14, color="Green")
    plt.show()
    plt.pie(below_50k,
labels=below_50k.index,
autopct='%1.0f%%')
    plt.title("Distribution of "+column+"
for <=50K", fontsize=14, color="Green")
    plt.show()

# Analyze data per unique value
def analyze_per_unique_value(column):
    unique_values = df[column].unique()
    for val in unique_values:
        val_df = df[df[column] == val]
        above_50k_count =
len(val_df[val_df["income"] ==
">50K"].index)
        below_50k_count =
len(val_df[val_df["income"] ==
"<=50K"].index)
        plt.pie([below_50k_count,
above_50k_count],
                labels=["<=50K (Count-
"+str(below_50k_count)+")", ">50K (Count-
"+str(above_50k_count)+")"],
                autopct='%1.0f%%')
        plt.title(column+": "+val,
fontsize=14, color="Green
```

HOURS & WORK

```
# Import necessary libraries and load
data
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.mosaicplot
import mosaic

colnames = ['age', 'workclass', 'fnlwgt',
'education', 'education_num',
'marital_status', 'occupation',
            'relationship', 'race',
'sex', 'capital_gain', 'capital_loss',
'hours_per_week', 'native_country',
            'income']

train = pd.read_csv("adult.data",
names=colnames, header=None)
train.drop(columns=train.columns[0],
axis=1, inplace=True)
train = train[['hours_per_week',
'workclass', 'income']]

test = pd.read_csv("adult.test",
names=colnames, header=None,
index_col=False, skipinitialspace=True)
test['income'] =
test['income'].str.rstrip('.')

df = pd.concat([train, test],
ignore_index=True)

# Create histograms of hours-per-week for
different income levels
df_less_than_fifty = df[df['income'] ==
'<=50K'].copy()
df_more_than_fifty = df[df['income'] ==
'>50K'].copy()

plt.hist(df_less_than_fifty['hours_per_we
ek'], bins=[0, 10, 20, 30, 40, 50, 60,
70, 80, 90, 100], color=['red'])
plt.title("Hours-Per-Week Graph of
Salary<50K", fontsize=20)
plt.xlabel("Hours-Per-Week", fontsize=14)
plt.ylabel("Frequency Count",
fontsize=14)

plt.hist(df_more_than_fifty['hours_per_we
ek'], bins=[0, 10, 20, 30, 40, 50, 60,
70, 80, 90, 100], color=['green'])
plt.title("Hours-Per-Week Graph of
Salary>=50K", fontsize=20)
plt.xlabel("Hours-Per-Week", fontsize=14)
plt.ylabel("Frequency Count",
fontsize=14)

# Create stacked bar chart of hours-per-
week for different income levels
less_hpw =
list(df_less_than_fifty['hours_per_week']
).copy()
more_hpw =
list(df_more_than_fifty['hours_per_week']
).copy()
dictionary = {'<=50K': less_hpw, '>50K':
more_hpw}
df_hpw = pd.DataFrame(dict([(k,
pd.Series(v)) for k, v in
dictionary.items()]))
plt.hist(df_hpw, bins=[0, 10, 20, 30, 40,
50, 60, 70, 80, 90, 100], histtype='bar',
stacked=True, color=['red', 'green'])
plt.title("Hours Per Week Stacked Bar
Chart", fontsize=20)
plt.xlabel("Hours-Per-Week", fontsize=14)
plt.ylabel("Count", fontsize=14)
plt.legend(['<=50k', '>50K'])
plt.grid()
```

```
# Create pie charts of workclass distribution for different income levels
df2 = df[['workclass', 'income', 'hours_per_week']]
df = df[['workclass', 'income']]
count_greater_fifty = df.loc[df['income'] == '>50K'].groupby('workclass').count()
count_lesser_fifty = df.loc[df['income'] == '<=50K'].groupby('workclass').count()

plt.pie(count_greater_fifty['income'], labels=count_greater_fifty.index, autopct='%1.1f%%', shadow=True, startangle=90)
```

RACE & INCOME

```
import numpy as np
import pandas as pd
from collections import Counter
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.graphics.mosaicplot import mosaic

%matplotlib inline

# Load the data
df = pd.read_csv("adult.data.filtered", index_col=False)
df.drop(columns=df.columns[0], axis=1, inplace=True)

# Split the data into income groups
less_than_fifty = df[df["income"] == "<=50K"]
more_than_fifty = df[df["income"] == ">50K"]

# Print the counts of each income group
print(f"Count(Above 50K) = {len(more_than_fifty)}")
print(f"Count(Below 50K) = {len(less_than_fifty)}")

# Print the first few rows of the data
df.head()

# Plot the race distribution for each income group
attribute = "race"
above_fifty_counts = Counter(more_than_fifty[attribute])
below_fifty_counts = Counter(less_than_fifty[attribute])

# Plot the pie charts
fig, axes = plt.subplots(ncols=2, nrows=1, figsize=(15,15))
colors = ['#2085ec','#72b4eb','#0a417a','#8464a0','#cea9bc','#323232']
axes[0].set_title("PIE CHART OF RACE DISTRIBUTION INCOME ABOVE 50K")
axes[0].pie(above_fifty_counts.values(), autopct='%0.0f%%', colors=colors)
axes[0].legend(above_fifty_counts.keys(), bbox_to_anchor=(1.0,0.4))
axes[1].set_title("PIE CHART OF RACE DISTRIBUTION INCOME BELOW 50K")
axes[1].pie(below_fifty_counts.values(), autopct='%0.0f%%', colors=colors)
axes[1].legend(below_fifty_counts.keys(), bbox_to_anchor=(1.0,0.4))
plt.tight_layout()
plt.show()
plt.close()
```

GENDER vs INCOME

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
from sklearn.linear_model import
LogisticRegression
from sklearn.metrics import
classification_report, confusion_matrix
from sklearn.model_selection import
train_test_split
from sklearn.preprocessing import
LabelEncoder, OrdinalEncoder,
MinMaxScaler

# Load data from CSV file
colnames = ['age', 'workclass', 'fnlwgt',
'education', 'education_num',
'marital_status', 'occupation',
            'relationship', 'race',
'sex', 'capital_gain', 'capital_loss',
'hours_per_week', 'native_country',
'income']
df = pd.read_csv("adult.csv",
names=colnames, header=None)
print(df.head())

# Select only female data and plot a
countplot
df_female = df[df.sex == "Female"]
sns.countplot(data=df, x='sex',
hue='income')

# Create a sex and income cross table and
plot it as a pie chart
sexcross = pd.crosstab(df['sex'],
df['income'])
sexcross.T.plot.pie(subplots=True,
figsize=(11, 12))

# Split the data into train and test sets
X = df.drop('income', axis=1)
y = df['income']
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=0)

# Preprocess categorical variables using
LabelEncoder and OrdinalEncoder
categorical_cols = ['workclass',
'education', 'marital_status',
'occupation', 'relationship', 'race',
'sex', 'native_country']
for col in categorical_cols:
    le = LabelEncoder()
    X_train[col] =
le.fit_transform(X_train[col].astype(str)
)
    X_test[col] =
le.transform(X_test[col].astype(str))

oe = OrdinalEncoder()
X_train[['income']] =
oe.fit_transform(X_train[['income']])
X_test[['income']] =
oe.transform(X_test[['income']])

# Scale numerical variables using
MinMaxScaler
scaler = MinMaxScaler()
numerical_cols = ['age', 'fnlwgt',
'education_num', 'capital_gain',
'capital_loss', 'hours_per_week']
X_train[numerical_cols] =
scaler.fit_transform(X_train[numerical_co
ls])
X_test[numerical_cols] =
scaler.transform(X_test[numerical_cols])

# Fit and train the logistic regression
model
model =
LogisticRegression(random_state=0)
model.fit(X_train, y_train)

# Predict using the test set
y_pred = model.predict(X_test)

# Evaluate the model's performance
print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test,
y_pred))
```