Problem Statements

1.Design a machine learning algorithm (MLA) on the dataset of your interest.

Designing the Machine Learning Algorithm on the **Income Data csv** of the certain people. With the attributes of:

1. **Age**
2. **Workclass**
3. **Fnlwgt (Final weight)**
4. **Education**
5. **Education-num**
6. **Occupation**
7. **Race**
8. **Gender**
9. **Capital-gain**
10. **Capital-Loss**
11. **Hours-per-week**
12. **Native-Country**
13. **Income > 50K (50,000)**

age	workclass	fnlwgt	education	educational-num	occupation

race	gender	capital-gain	capital-loss	hours-per-week	native-country	income_>50K

And here by performing the Machine Learning Algorithm like k-NN classification, Decision Tree, Confusion Matrix and Various plot using seaborn and Matplotlib Library.

Using the Spyder platform to execute all the programs or Machine Learning Algorithm.

Algorithm:

1. Import the required libraries: pandas, warnings, seaborn, matplotlib.pyplot, numpy, ConfusionMatrixDisplay, and classification_report.

2. Suppress warnings using warnings.filterwarnings('ignore').

3. Read the dataset using pd.read_csv("income_data.csv") and store it in a variable called data_set.

4. Print the dataset using print(data_set).

5. Plot the countplot of workclass using sns.countplot(x='workclass', data=data_set,) and plt.show().

6. Define a function called graph that takes a parameter y.

7. Inside the graph function, plot the boxplot using sns.boxplot(x='workclass', y=y, data=data_set) and plt.figure(figsize=(13, 15)).

8. Call the graph function with 'age', 'hours-per-week', and 'educational-num' as parameters.

9. Plot the heatmap using sns.heatmap(data_set.corr(method='pearson').drop(['age'],axis=1).drop(['age'],axis=0),annot=True).

10. Plot the violin plot using fig, ax= plt.subplots(figsize = (9, 7)) and sns.violinplot(ax = ax , x = data_set["age"],y=data_set["workclass"]).

11. Plot the pairplot using sns.pairplot(data_set.drop(['workclass'], axis = 1), hue = 'education', height=2) and plt.show().

12. Split the dataset into training and testing sets using train_test_split(x, y, test_size=0.25, random_state=0).

13. Fit the DecisionTreeClassifier to the training set using DecisionTreeClassifier(max_depth=5, random_state=1) and classifier.fit(x_train, y_train).

14. Predict the test set result using y_pred= classifier.predict(x_test).

15. Calculate the accuracy score using acs=accuracy_score(y_test, y_pred).

16. Create the confusion matrix using cm = confusion_matrix(y_test, y_pred) and ConfusionMatrixDisplay(confusion_matrix=cm).plot().

17. Print the classification report using print("Classification Report:\n" ,classification_report(y_test, y_pred)).

18. Print the accuracy score using print("Accuracy =" ,acs).

Code:

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
from sklearn.metrics import ConfusionMatrixDisplay
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report

#Reading dataset
data_set=pd.read_csv("income_data.csv")
print(data_set)

#Ploting the countplot of workclass

sns.countplot(x='workclass', data=data_set,)
plt.show()

#Plotting the boxplot
def graph(y):

    sns.boxplot(x='workclass', y=y, data=data_set)
    plt.figure(figsize=(13, 15))

graph('age')
graph('hours-per-week')
graph('educational-num')

plt.show()

#plotting heatmap
sns.heatmap(data_set.corr(method='pearson').drop(['age'],axis=1).drop(['age'],axis=0),annot=True);


#violin plot
fig, ax= plt.subplots(figsize = (9, 7))
sns.violinplot(ax = ax , x = data_set["age"],y=data_set["workclass"])

#plotting pairplot
sns.pairplot(data_set.drop(['workclass'], axis = 1),
             hue = 'education', height=2)
plt.show()

#DecisionTree

x= data_set.iloc[:, 0:1].values
y= data_set.iloc[:, -1].values

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)
```

```
#fitting K-NN classifier to the training set
from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(max_depth=5, random_state=1)

classifier.fit(x_train, y_train)
#predicting the test set result
y_pred= classifier.predict(x_test)
result=classifier.score(x_test, y_test)

#creating the confusion matrix
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
disp= ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.show()

print("Classification Report:\n" ,classification_report(y_test, y_pred))

#print(cm)

acs=accuracy_score(y_test, y_pred)
print("Accuracy =" ,acs)
```

Console:

```
      age     workclass  fnlwgt  ... hours-per-week  native-country income_>50K
0      67       Private  366425  ...             60   United-States           1
1      17       Private  244602  ...             15   United-States           0
2      31       Private  174201  ...             40   United-States           1
3      58     State-gov  110199  ...             40   United-States           0
4      25     State-gov  149248  ...             40   United-States           0
...   ...           ...     ...  ...            ...             ...         ...
1169   38     State-gov   34364  ...             40   United-States           0
1170   28     State-gov  175325  ...             40   United-States           0
1171   44  Self-emp-inc  121352  ...             50   United-States           0
1172   60  Self-emp-inc   93272  ...             60   United-States           0
1173   34     State-gov  118551  ...             25             NaN           1

[1174 rows x 13 columns]
<Figure size 936x1080 with 0 Axes>
```

Outputs and answering the given questions.

2.Visualise the dataset using Matplotlib and Seaborn

Importing the libraries

```
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as nm
import matplotlib.pyplot as mtp
import pandas as pd
from sklearn.metrics import ConfusionMatrixDisplay
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
```

Importing various libraries required for performing machine learning algorithm which includes **Matplotlib** and **Seaborn**

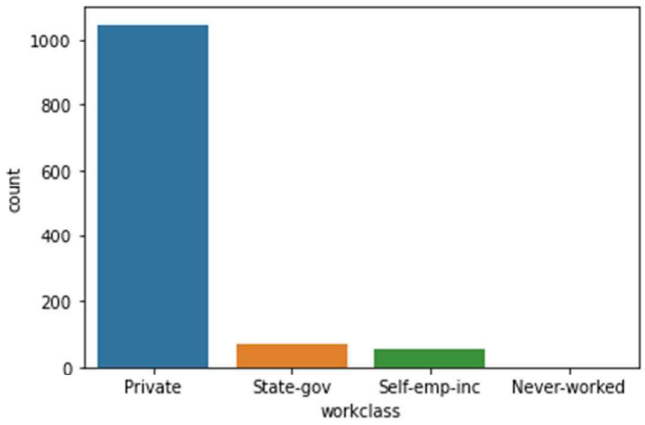
```
import seaborn as sns
import matplotlib.pyplot as plt
```

For reading the csv file and plotting the count-plot for **workclass.**

```
#Reading dataset
data_set=pd.read_csv("income_data.csv")
print(data_set)

#Ploting the countplot of workclass

sns.countplot(x='workclass', data=data_set,)
plt.show()
```

Here, **matplotlib** as defined as **plt.show().**

Count plot

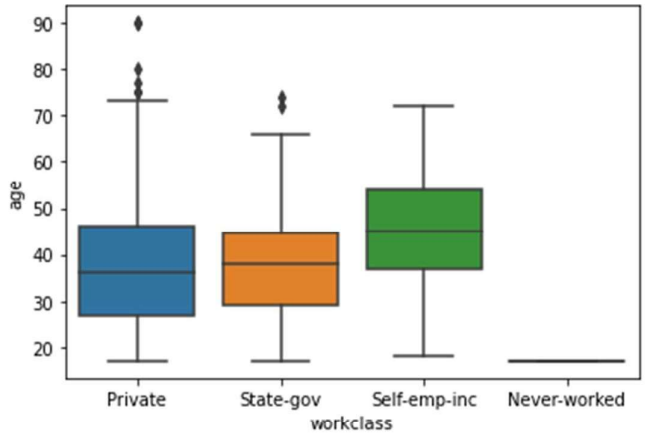


According to the given data of **Income data csv** the count plot for workclass which has 4 different workclasses- **Private, State-Gov, Self-Emp-inc & Never-worked.** And here the count plot shows that count of people who works in private companies are more, next comes the state-Government employees.

Box Plot

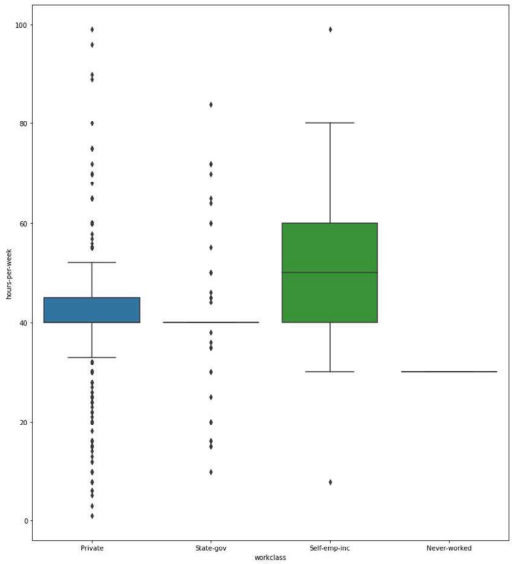
To plot the box plot here by we used **seaborn** library and defined **sns.**
So, here we are plotting the boxplot for **age , hours-per-week & educational-num.**

Graphs:
Age Boxplot



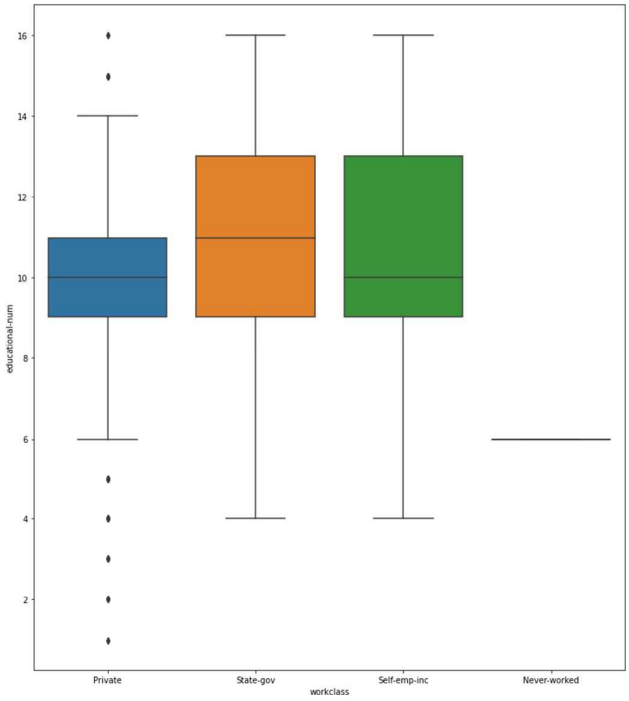
Age boxplot here we can know the graphical representation of how many different age group people works in different work classes.

Hours-per-week Boxplot



This graph represents, in different work class how many hours of work a person has been done.

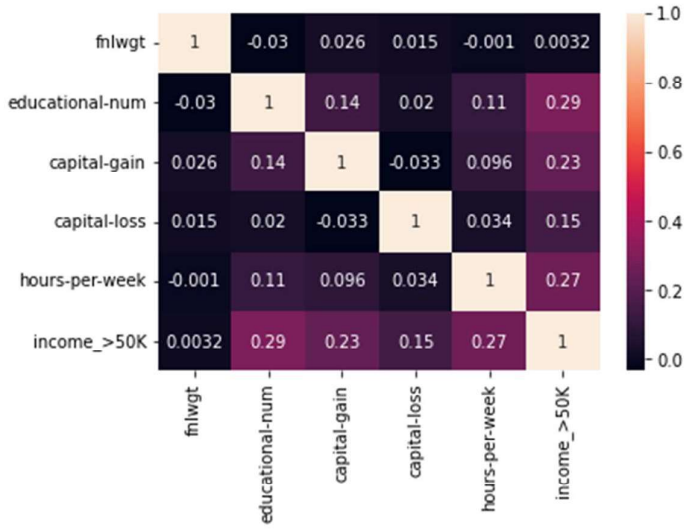
Educational-number



This represents the number of Educational level of people and their work-classes.

Heat Map:

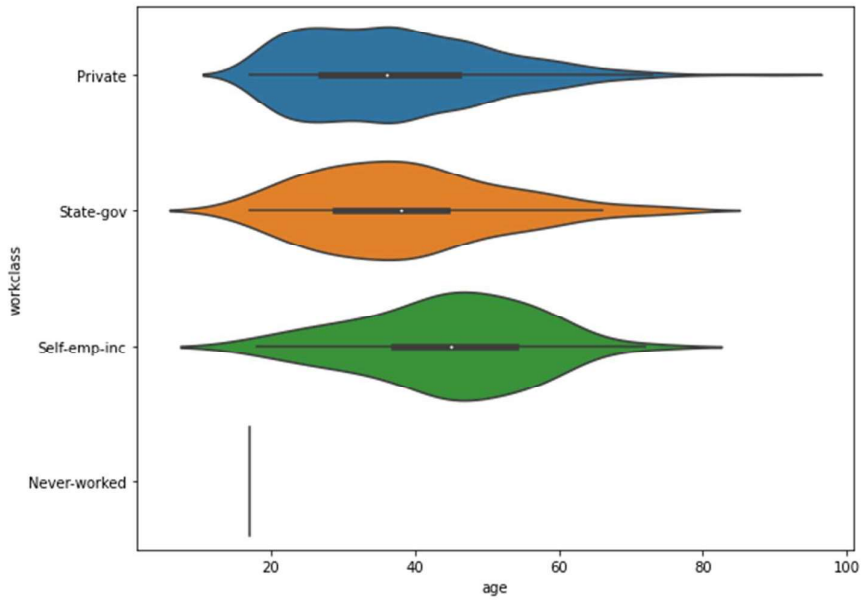
```
#plotting heatmap
sns.heatmap(data_set.corr(method='pearson').drop(['age'],axis=1).drop(['age'],axis=0),annot=True);
```



Violin Plot:

```
#violin plot
fig, ax= plt.subplots(figsize = (9, 7))
sns.violinplot(ax = ax , x = data_set["age"],y=data_set["workclass"])
```

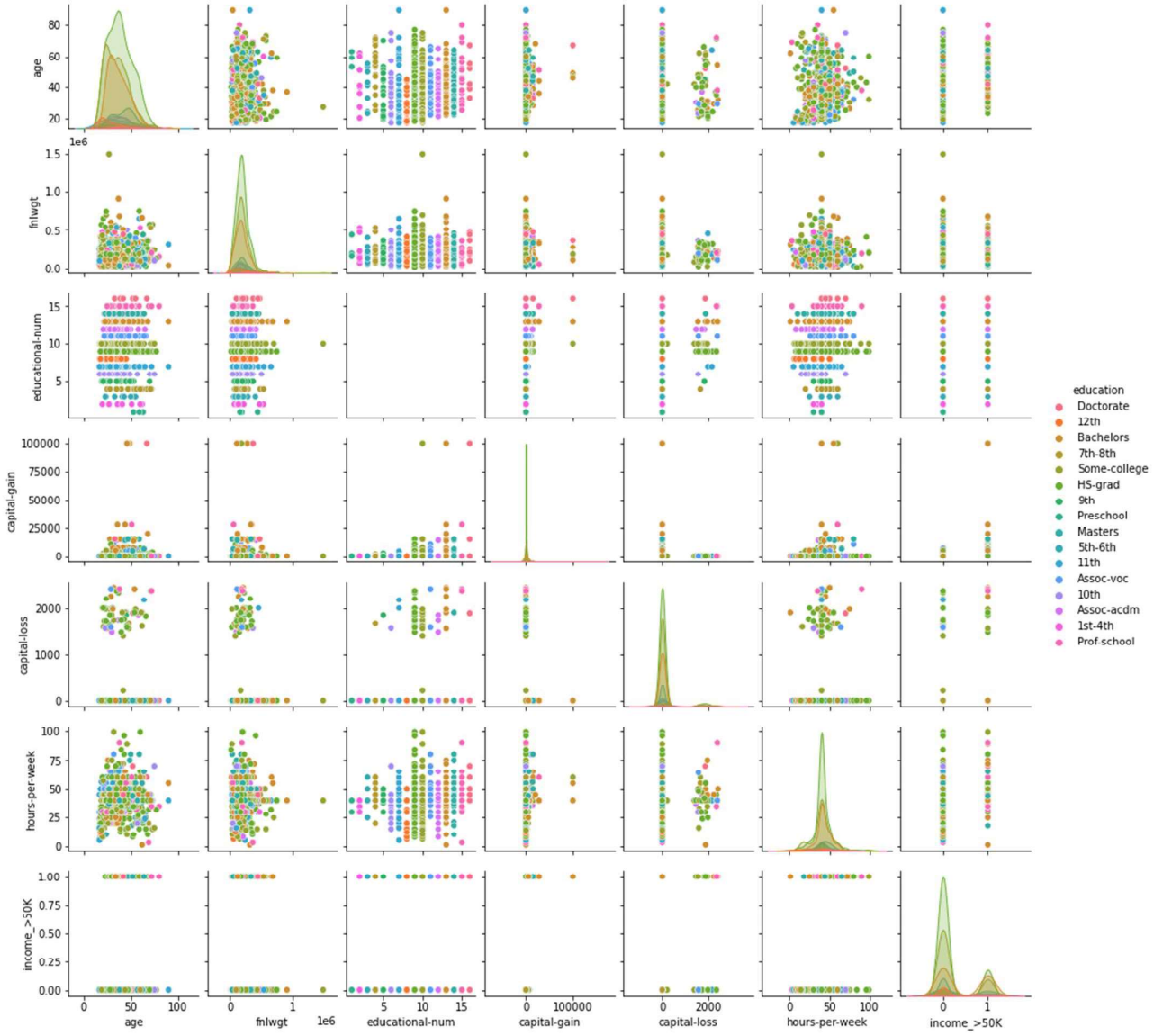
Here by plotting the violin plot for different age group and their work-class.



Pair Plotting:

```
#plotting pairplot
sns.pairplot(data_set.drop(['workclass'], axis = 1),
             hue = 'education', height=2)
plt.show()
```

Pair plot of different work-class with their different attributes.

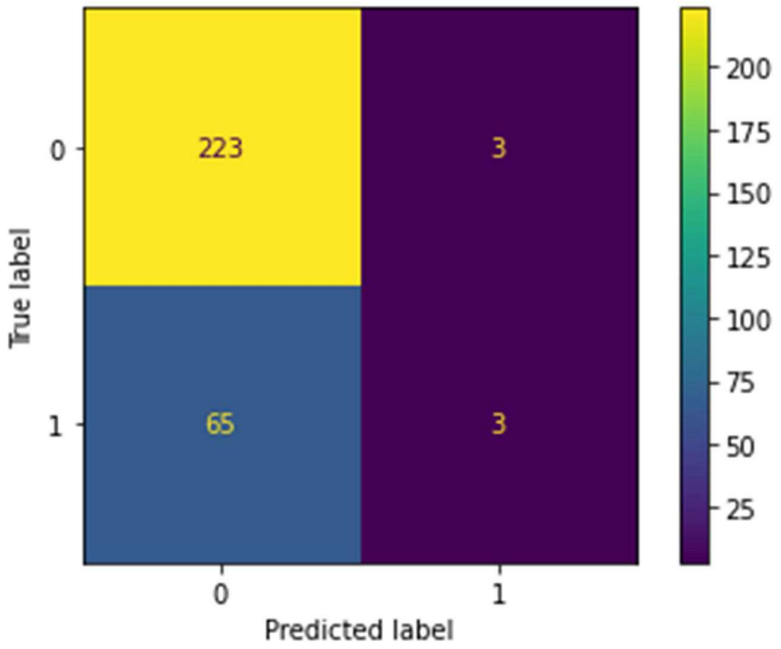


3.Discuss the performance of the MLA on the dataset

Performing the Decision Tree Machine Learning Algorithm with confusion matrix for the different age groups.

```
#DecisionTree

x= data_set.iloc[:, 0:1].values
y= data_set.iloc[:, -1].values

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)

#fitting K-NN classifier to the training set
from sklearn.tree import DecisionTreeClassifier

classifier = DecisionTreeClassifier(max_depth=5, random_state=1)

classifier.fit(x_train, y_train)
#predicting the test set result
y_pred= classifier.predict(x_test)
result=classifier.score(x_test, y_test)

#creating the confusion matrix
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
disp= ConfusionMatrixDisplay(confusion_matrix=cm)
disp.plot()
plt.show()

print("Classification Report:\n" ,classification_report(y_test, y_pred))

#print(cm)

acs=accuracy_score(y_test, y_pred)
print("Accuracy =" ,acs)
```

Confusion Matrix:



Classification Report with Accuracy:

```
Classification Report:
              precision    recall  f1-score   support

           0       0.77      0.99      0.87       226
           1       0.50      0.04      0.08        68

    accuracy                           0.77       294
   macro avg       0.64      0.52      0.47       294
weighted avg       0.71      0.77      0.69       294

Accuracy = 0.7687074829931972
```

Conclusion:

Concluding this Assignment, here by I have performed the various plotting using seaborn and matplot libraries and DecisionTreeClassifier Machine Learning Algorithm on income data set.
This data set shows that income of people and their attribute like age, education and work class etc.
I have got Accuracy of 0.7687 = 76.87% of accuracy.