Predictive Analysis

```
import pandas as pd
from sklearn.naive_bayes import GaussianNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import numpy as np
import itertools
```

A function that reads the dataset and returns the data as a dataframe

```
def read_data():
    data = pd.read_csv(r'adult.csv')
    print(data.head())
    return(data)
```

This function handles the pre-processing of the data, The main procedure followed where,

1)Replacing the missing value of categorical attribute with their respective modes

2)One-hot encoding the data

Note that all of the values of numerical attributes where present

```
def pre_processing(data):
    categorical_att = ['workclass', 'education', 'marital-status', 'occupation', 'relatior
    for att in categorical_att:
        mode = data[att].mode()[0]
        data[att].replace(['?'], mode, inplace = True)

    for att in categorical_att:
        if(att == 'income'):
            pass
        else:
            one_hot = pd.get_dummies(data[att])
            data = data.drop(att, axis = 1)
            data = data.join(one_hot)

    print(data.head())
    return(data)
```

This is the function that fits a gaussian model to the data and predict.

Confusion matrix of the same has been constructed.

```
def gaussian_nb(data_train, data_test, target_train, target_test):
    print("fitting with gaussian naive bayes ... ")
    gnb = GaussianNB().fit(data_train, target_train)
```

```
    gnb = GaussianNB().fit(data_train, target_train)
    gnb_predictions = gnb.predict(data_test)

    accuracy = gnb.score(data_test, target_test)
    print("The accuracy of the model is : ", accuracy)
    cm = confusion_matrix(target_test, gnb_predictions)
    print("The confusion matrix is : ")
    print(cm)
```

This is the function that fits a decision tree model to the data and predict. The depth of the decior validation.

Confusion matrix of the same has been constructed.

```
def decision_tree(data_train, data_test, target_train, target_test):
    print("fitting with decision tree of depth 5 ... ")
    dtree_model = DecisionTreeClassifier(max_depth = 5).fit(data_train, target_train)
    dtree_predictions = dtree_model.predict(data_test)

    accuracy = dtree_model.score(data_test, target_test)
    print("The accuracy of the model is : ", accuracy)
    cm = confusion_matrix(target_test, dtree_predictions)
    print("The confusion matrix is : ")
    print(cm)
```

This is the main runner code, here train_test_split() function is used to split the training data into t the data randomly given to training the model.

The accuracy and confusion matrix are constructed using the test dataset.

```
ata()
a = pre_processing(data)
st(set(processed_data.columns) - {'income'})
ome'

ata_test, target_train, target_test = train_test_split(processed_data[features], processed
ata_train, data_test, target_train, target_test)
(data_train, data_test, target_train, target_test)
```

Calling main ....

```
main()
```

```
   age  workclass  fnlwgt  ... hours-per-week  native-country income
0   25    Private  226802  ...             40   United-States  <=50K
1   38    Private   89814  ...             50   United-States  <=50K
2   28  Local-gov  336951  ...             40   United-States   >50K
3   44    Private  160323  ...             40   United-States   >50K
4   18          ?  103497  ...             30   United-States  <=50K

[5 rows x 15 columns]
   age  fnlwgt  educational-num  ...  United-States  Vietnam  Yugoslavia
0   25  226802                7  ...              1        0           0
1   38   89814                9  ...              1        0           0
2   28  336951               12  ...              1        0           0
3   44  160323               10  ...              1        0           0
4   18  103497               10  ...              1        0           0

[5 rows x 106 columns]
fitting with gaussian naive bayes ...
The accuracy of the model is :  0.7941597993550699
The confusion matrix is :
[[28134  1588]
 [ 6455  2897]]
fitting with decision tree of depth 5 ...
The accuracy of the model is :  0.8487997133643855
The confusion matrix is :
[[28447  1275]
 [ 4633  4719]]
```