Importing Libraries

```
import numpy
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import BernoulliNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import classification_report
```

Reading Data Frame

```
df = pd.read_csv('payment_fraud.csv')
df.head()
```

```
   accountAgeDays  numItems  localTime paymentMethod
paymentMethodAgeDays  \
0              29         1   4.745402        paypal
28.204861
1             725         1   4.742303   storecredit
0.000000
2             845         1   4.921318    creditcard
0.000000
3             503         1   4.886641    creditcard
0.000000
4            2000         1   5.040929    creditcard
0.000000

   label
0      0
1      0
2      0
3      0
4      0
```

Encoding String Data Fields

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
df['paymentMethod'] = labelencoder.fit_transform(df['paymentMethod'])
df.head()
```

```
   accountAgeDays  numItems  localTime  paymentMethod
paymentMethodAgeDays  \
0              29         1   4.745402              1
28.204861
```

```
1              725          1   4.742303               2
0.000000
2              845          1   4.921318               0
0.000000
3              503          1   4.886641               0
0.000000
4             2000          1   5.040929               0
0.000000

   label
0      0
1      0
2      0
3      0
4      0
```

Classification Algorithms

```
decisiontree = DecisionTreeClassifier()
logisticregression = LogisticRegression()
knearestclassifier = KNeighborsClassifier()
bernoulli_naiveBayes = BernoulliNB()
passiveAggressive = PassiveAggressiveClassifier()

x = np.array(df[["accountAgeDays", "numItems", "localTime",
"paymentMethod", "paymentMethodAgeDays"]])
y = np.array(df[["label"]])
```

Dividing data into test and train sets

```
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.10,
random_state=42)

knearestclassifier.fit(X_train, y_train)

KNeighborsClassifier()

decisiontree.fit(X_train, y_train)

DecisionTreeClassifier()

logisticregression.fit(X_train, y_train)

LogisticRegression()

passiveAggressive.fit(X_train, y_train)

PassiveAggressiveClassifier()

data1 = {"Classification Algorithms": ["KNN Classifier", "Decision
Tree Classifier",
```

```
                                          "Logistic Regression", "Passive
Aggressive Classifier"],
      "Score": [knearestclassifier.score(x, y), decisiontree.score(x,
y),
                logisticregression.score(x, y),
passiveAggressive.score(x, y)]}
score = pd.DataFrame(data1)
score
```

```
/Users/sravya/anaconda3/lib/python3.10/site-packages/sklearn/
base.py:420: UserWarning: X does not have valid feature names, but
KNeighborsClassifier was fitted with feature names
  warnings.warn(
/Users/sravya/anaconda3/lib/python3.10/site-packages/sklearn/base.py:4
20: UserWarning: X does not have valid feature names, but
DecisionTreeClassifier was fitted with feature names
  warnings.warn(
/Users/sravya/anaconda3/lib/python3.10/site-packages/sklearn/base.py:4
20: UserWarning: X does not have valid feature names, but
LogisticRegression was fitted with feature names
  warnings.warn(
/Users/sravya/anaconda3/lib/python3.10/site-packages/sklearn/base.py:4
20: UserWarning: X does not have valid feature names, but
PassiveAggressiveClassifier was fitted with feature names
  warnings.warn(
```

```
       Classification Algorithms     Score
0                KNN Classifier  0.999924
1       Decision Tree Classifier  1.000000
2            Logistic Regression  1.000000
3  Passive Aggressive Classifier  1.000000
```

According to the above output, the Decision Tree classification, Logistic Regression, passive Aggressive Classifier algorithm perform with the perfect score on this dataset.