Trial 2 with normalised attributes

Loading Modules and Functions

In [ ]:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import classification_report,accuracy_score,confusion_matrix
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd
```

Loading Data

In [ ]:

```
df = pd.read_csv("../processed.csv")
df
```

Out[ ]:

	Initial_Price	Final_Price	Win_Flag	Mac_Flag	Linux_Flag	Positive_Reviews	Negative_Reviews	Memory_MB	Storage_MB	target
0	52.0	52.0	True	True	False	57.0	7.0	1024	50	1
1	0.0	0.0	True	True	False	53.0	6.0	2048	3072	1
2	0.0	0.0	True	False	False	133.0	69.0	2048	100	0
3	530.0	530.0	True	False	False	22.0	9.0	2048	500	0
4	229.0	229.0	True	True	True	226.0	44.0	2048	1500	1
...	...	...	...	...	...	...	...	...	...	...
57467	85.0	85.0	True	False	False	0.0	4.0	4096	200	-1
57468	349.0	349.0	True	True	False	2.0	1.0	1024	1024	1
57469	164.0	164.0	True	False	False	8.0	1.0	4096	20480	1
57470	610.0	610.0	True	False	False	1.0	0.0	4096	3072	1
57471	570.0	285.0	True	False	False	0.0	1.0	1024	2048	-1

57472 rows × 10 columns

Normalising Continous Features

In [ ]:

```
def normalise(feature, df):
    mean = df[feature].mean()
    sd = df[feature].std()
    df[feature] = (df[feature] - mean) / sd

normalise("Initial_Price", df)
normalise("Final_Price", df)
normalise("Positive_Reviews", df)
normalise("Negative_Reviews", df)
normalise("Memory_MB", df)
normalise("Storage_MB", df)
df
```

Out[ ]:

	Initial_Price	Final_Price	Win_Flag	Mac_Flag	Linux_Flag	Positive_Reviews	Negative_Reviews	Memory_MB	Storage_MB	target
0	-0.271301	-0.258274	True	True	False	-0.034488	-0.033031	-0.004171	-0.004171	1
1	-0.322070	-0.309572	True	True	False	-0.034609	-0.033196	-0.004171	-0.004171	1
2	-0.322070	-0.309572	True	False	False	-0.032193	-0.022842	-0.004171	-0.004171	0
3	0.195385	0.213271	True	False	False	-0.035545	-0.032703	-0.004171	-0.004171	0
4	-0.098490	-0.083665	True	True	True	-0.029385	-0.026950	-0.004171	-0.004171	1
...	...	...	...	...	...	...	...	...	...	...
57467	-0.239082	-0.225720	True	False	False	-0.036210	-0.033524	-0.004171	-0.004171	-1
57468	0.018669	0.034715	True	True	False	-0.036149	-0.034017	-0.004171	-0.004171	1
57469	-0.161952	-0.147787	True	False	False	-0.035968	-0.034017	-0.004171	-0.004171	1
57470	0.273492	0.292190	True	False	False	-0.036179	-0.034182	-0.004171	-0.004171	1
57471	0.234439	-0.028421	True	False	False	-0.036210	-0.034017	-0.004171	-0.004171	-1

57472 rows × 10 columns

Splitting data to 33% Test , 66% Train

In [ ]:
```
y = df["target"]
X = df.drop(labels=["target","Win_Flag","Mac_Flag","Linux_Flag"],axis=1)


X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.33, random_state=100,shuffle=True)
```

Fitting the model

In [ ]:
```
bayes = GaussianNB()
bayes.fit(X_train, y_train)
```

Out[ ]:
```
▾ GaussianNB
GaussianNB()
```

Likelihood of predictions

In [ ]:
```
bayes.predict_proba(X_test)
```

Out[ ]:
```
array([[9.90667437e-01, 9.33256327e-03, 9.81563140e-12],
       [9.91290910e-01, 8.70909043e-03, 9.19860142e-12],
       [9.91237383e-01, 8.76261690e-03, 9.22623962e-12],
       ...,
       [9.91328087e-01, 8.67191274e-03, 9.17865198e-12],
       [9.87943714e-01, 1.20562857e-02, 1.28388006e-11],
       [9.91392419e-01, 8.60758149e-03, 9.16860493e-12]])
```

Evaluating Model with test data

In [ ]:
```
preds = bayes.predict(X_test)
print(f"Accuracy : {accuracy_score(y_test, preds)}")
```

```
Accuracy : 0.11879152167035748
```

Important Metrics

In [ ]:
```
print(classification_report(y_test, preds))
```

```
              precision    recall  f1-score   support

          -1       0.12      1.00      0.21      2123
           0       0.10      0.03      0.05      3563
           1       1.00      0.00      0.00     13280

    accuracy                           0.12     18966
   macro avg       0.41      0.34      0.09     18966
weighted avg       0.73      0.12      0.04     18966
```

Display Confusion Matrix

In [ ]:
```
confusion_matrix(y_test,preds)
```

Out[ ]:
```
array([[ 2116,     7,     0],
       [ 3451,   112,     0],
       [12224,  1031,    25]], dtype=int64)
```

2nd trial Conclusion

1. Bayes Classifier's performance fall to 11% accuracy , after normalising attributes and discarding flags to keep only continous attributes
2. This follows that the continous features of dataset were not *guassian* , that is they are not normally distributed.