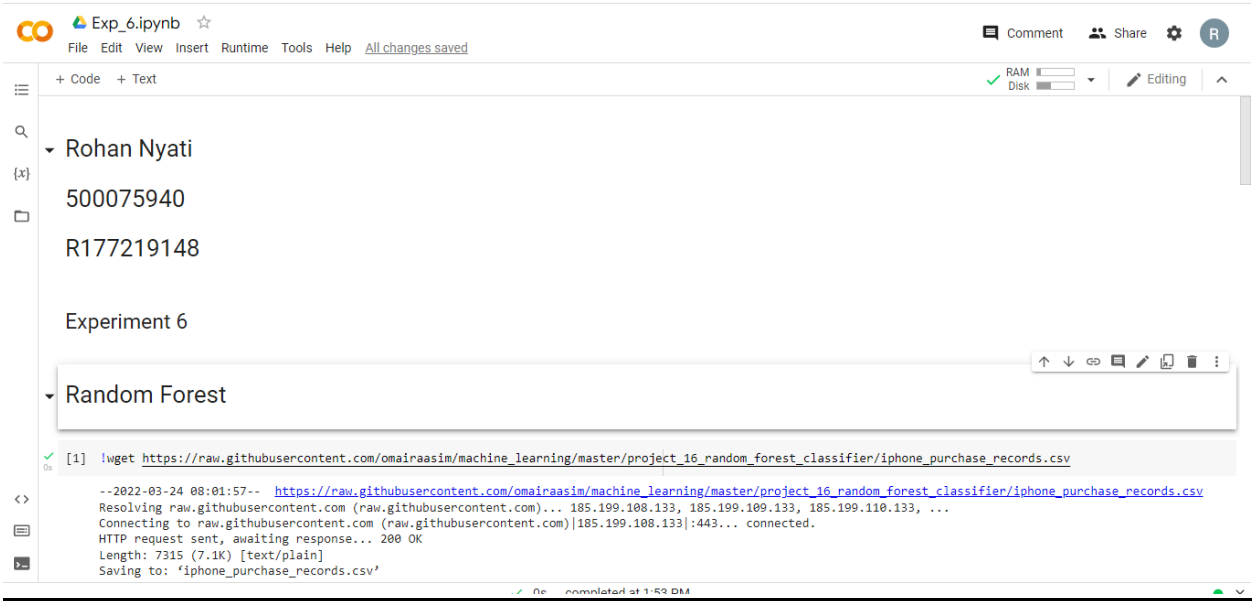
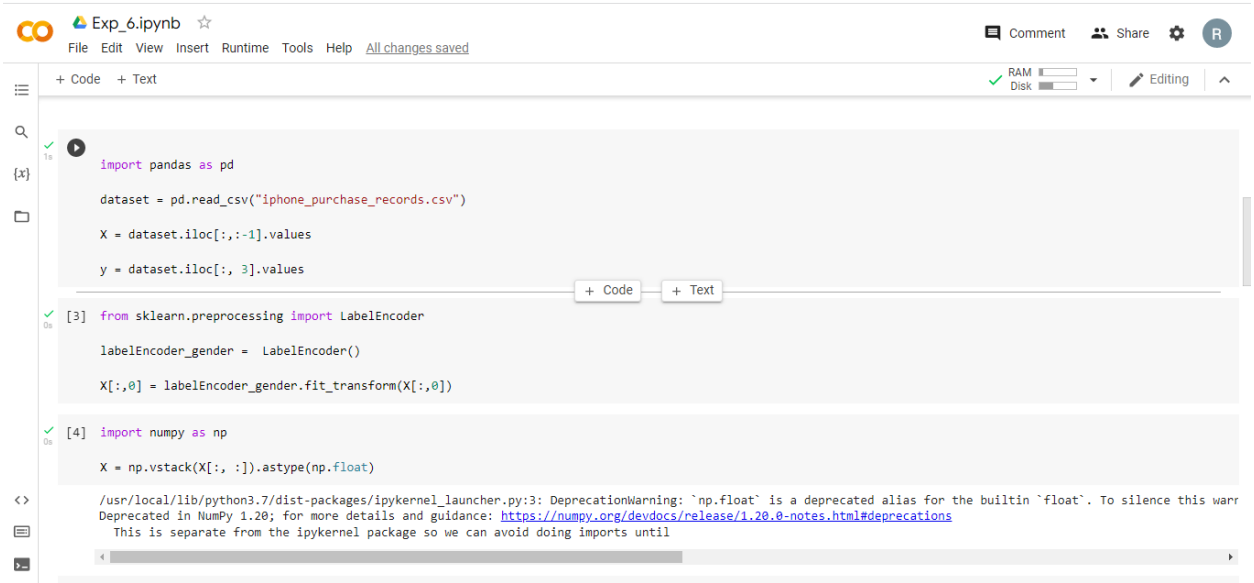
Rohan Nyati
500075940
R177219148
Batch 5

Experiment – 6





Exp_6.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share R

+ Code + Text

RAM Disk Editing

```
[5] 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)
```

```
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=100, criterion="entropy", random_state=0)

classifier.fit(X_train, y_train)
```

```
RandomForestClassifier(criterion='entropy', random_state=0)
```

```
[7] y_pred = classifier.predict(X_test)
```

```
[8] from sklearn import metrics

    cm = metrics.confusion_matrix(y_test, y_pred)

    print(cm)

    accuracy = metrics.accuracy_score(y_test, y_pred)

    print("Accuracy score:",accuracy)
```

Exp_6.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share R

+ Code + Text

RAM Disk Editing

```
print(cm)

accuracy = metrics.accuracy_score(y_test, y_pred)

print("Accuracy score:",accuracy)

precision = metrics.precision_score(y_test, y_pred)

print("Precision score:",precision)

recall = metrics.recall_score(y_test, y_pred)

print("Recall score:",recall)
```

```
[[64  4]
 [ 4 28]]
Accuracy score: 0.92
Precision score: 0.875
Recall score: 0.875
```

Decision Tree

```
[18] from sklearn.tree import DecisionTreeClassifier
     classifier = DecisionTreeClassifier(criterion = "entropy", random_state=0)
     classifier.fit(X_train, y_train)
```

Exp_6.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share R

+ Code + Text

RAM Disk Editing

```
[18] classifier = DecisionTreeClassifier(criterion = "entropy", random_state=0)
     classifier.fit(X_train, y_train)

     DecisionTreeClassifier(criterion='entropy', random_state=0)

[19] clf.fit(X_train,y_train)

     DecisionTreeClassifier(random_state=0)

     from sklearn import metrics
     cm = metrics.confusion_matrix(y_test, y_pred)
     print(cm)
     accuracy = metrics.accuracy_score(y_test, y_pred)
     print("Accuracy score:",accuracy)
     precision = metrics.precision_score(y_test, y_pred)
     print("Precision score:",precision)
     recall = metrics.recall_score(y_test, y_pred)
     print("Recall score:",recall)

     [[64  4]
      [ 4 28]]
     Accuracy score: 0.92
     Precision score: 0.875
     Recall score: 0.875
```

```
print(metrics.classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.94	0.94	0.94	68
1	0.88	0.88	0.88	32
accuracy			0.92	100
macro avg	0.91	0.91	0.91	100
weighted avg	0.92	0.92	0.92	100