ASSIGNMENT 4

NAME : FALAQ KHAN

BATCH:Machine Learning and AI Batch A3

1. Explain the significance of F1-Score.

>>> F1-Score provides a way to combine both precision and recall into a single measure that captures both properties.
Alone, neither precision or recall tells the whole story. We can have excellent precision with terrible recall, or alternately, terrible precision with excellent recall. F1-Score provides a way to express both concerns with a single score.
Once precision and recall have been calculated for a binary or multiclass classification problem, the two scores can be combined into the calculation of the F-Measure.
F1-Score = (2 * Precision * Recall) / (Precision + Recall)
This is the harmonic mean of the two fractions. This is sometimes called the F-Score or the F-Measure and might be the most common metric used on imbalanced classification problems.

2. **Use RMSE in Python on any arbitrary model.**

>>>

```
from sklearn.metrics import mean_squared_e
import math y_actual = [1,2,3,4,5] y_predicted = [1.6,2.5,2.9,3,4.1]
```

```
MSE = mean_squared_error(y_actual, y_predicted)
RMSE = math.sqrt(MSE)
print("Root Mean Square Error:\n")
 print(RMSE)
```

Output: Root Mean Square Error: 0.6971370023173351

3. **Give appropriate examples where precision, recall, accuracy would be ideal metrics to do model selection**.

>>> **Precision**: from sklearn.metrics import precision_score

```
precision = precision_score(Y_test, predicted, average='binary')
 print('Precision: %.3f' % precision)
```

Output: 0.708

Recall: from sklearn.metrics import recall_score

```
 recall = recall_score(Y_test, predicted, average='binary')
```

print('Recall: %.3f' % recall)

Output: 0.554

Accuracy:

import pandas

from sklearn import model_selection

from sklearn.linear_model import LogisticRegression
url="https://raw.githubusercontent.com/jbrownlee/Datasets/master/pima-indians-

diabetes.data.csv"

names = ['preg', 'plas', 'pres', 'skin', 'test', 'mass', 'pedi', 'age', 'class'] dataframe =
pandas.read_csv(url, names=names)

array = dataframe.values X = array[:,0:8]

Y = array[:,8]

```
kfold = model_selection.KFold(n_splits=10, random_state=7, shuffle=True) model =
LogisticRegression(solver='liblinear')

scoring = 'accuracy'

results = model_selection.cross_val_score(model, X, Y, cv=kfold, scoring=scoring)
print("Accuracy: %.3f (%.3f)" % (results.mean(), results.std()))
```

Output: 0.771 (0.051)

(4&5 I COULDNT FIND APPROPRIATE ANSWERS. I'LL TRY SEARCHING THEM AND ADDING IT TO MY NOTES.)