1.) Dataset exploration

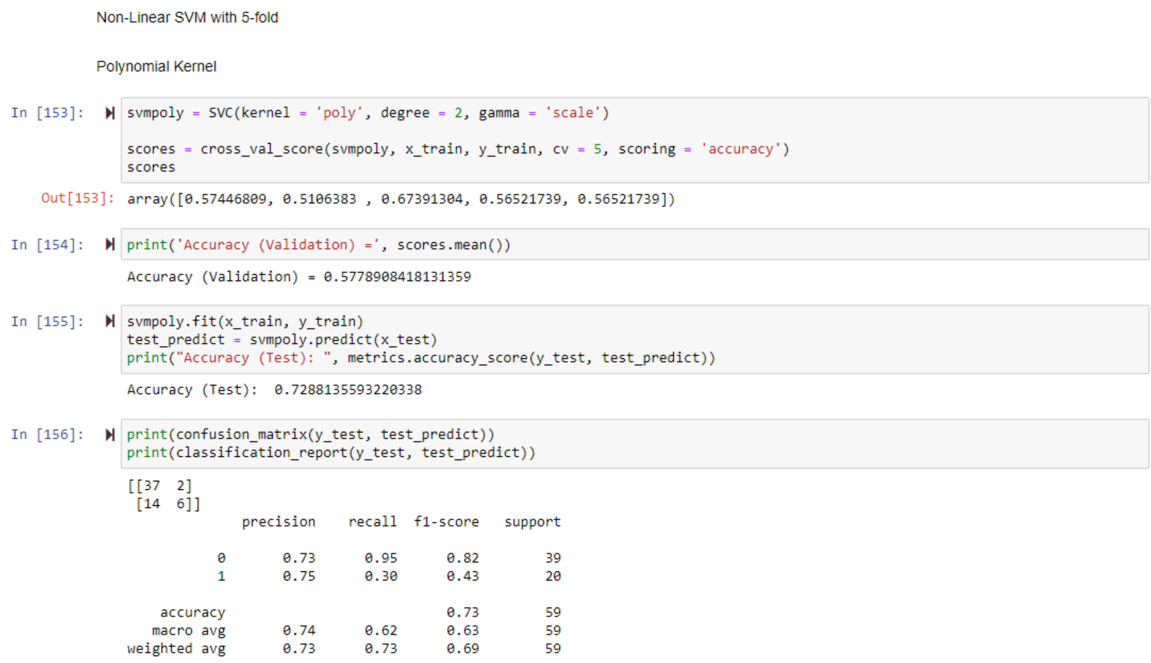
(a) The total count of depression is consisting of 155 (No) and 136 (Yes). The difference for depressed and non-depressed is small, hence the output variable is considered balance.

(b) Use F1 score to evaluate the performance of classification model because the number of depressed and non-depressed is quite balance.

(c) First use of KNN classifier to check on the accuracy with classifier (K=1/3/5), all 3 models are only having less than 50% accuracy. Then, try on with Naïve Bayes Classifier, the Test accuracy show better results.

	Accuracy	
Naive Bayes Classifier	**Validation**	**Test**
Gaussian Naive Bayes	53%	71%
Bernoulli Naive Bayes	59%	64%
Multinomial Naive Bayes	61%	66%

Eventually, decide to use non-linear kernel with k-fold cross validation. By increasing the k from 5-fold cross validation to 10-fold will not improve on the validation and test accuracy. By checking on the F1 score, the best fit model is Polynomial kernel with 5-fold cross validation, where the degree is 2 and gamma is set to scale.



Non-Linear SVM with 5-fold

Polynomial Kernel

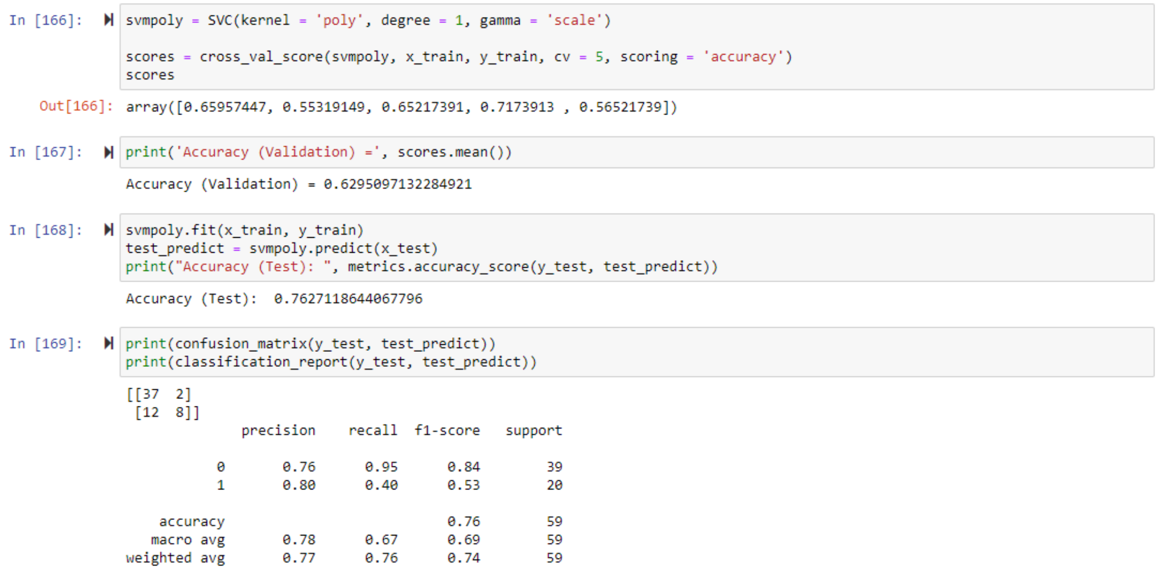
```
In [153]:  svmpoly = SVC(kernel = 'poly', degree = 2, gamma = 'scale')

           scores = cross_val_score(svmpoly, x_train, y_train, cv = 5, scoring = 'accuracy')
           scores

Out[153]:  array([0.57446809, 0.5106383 , 0.67391304, 0.56521739, 0.56521739])

In [154]:  print('Accuracy (Validation) =', scores.mean())

           Accuracy (Validation) = 0.5778908418131359

In [155]:  svmpoly.fit(x_train, y_train)
           test_predict = svmpoly.predict(x_test)
           print("Accuracy (Test): ", metrics.accuracy_score(y_test, test_predict))

           Accuracy (Test):  0.7288135593220338

In [156]:  print(confusion_matrix(y_test, test_predict))
           print(classification_report(y_test, test_predict))

           [[37  2]
            [14  6]]
                         precision    recall  f1-score   support

                      0       0.73      0.95      0.82        39
                      1       0.75      0.30      0.43        20

               accuracy                           0.73        59
              macro avg       0.74      0.62      0.63        59
           weighted avg       0.73      0.73      0.69        59
```

The comparison for SVM non-linear kernel with 10-fold and 5-fold cross validation as below.

10-fold	**Accuracy**		
SVM	**Validation**	**Test**	F1 score
Polynomial Kernel	57%	71%	71%
RBF Kernel	57%	59%	59%
Sigmoid Kernel	50%	32%	32%

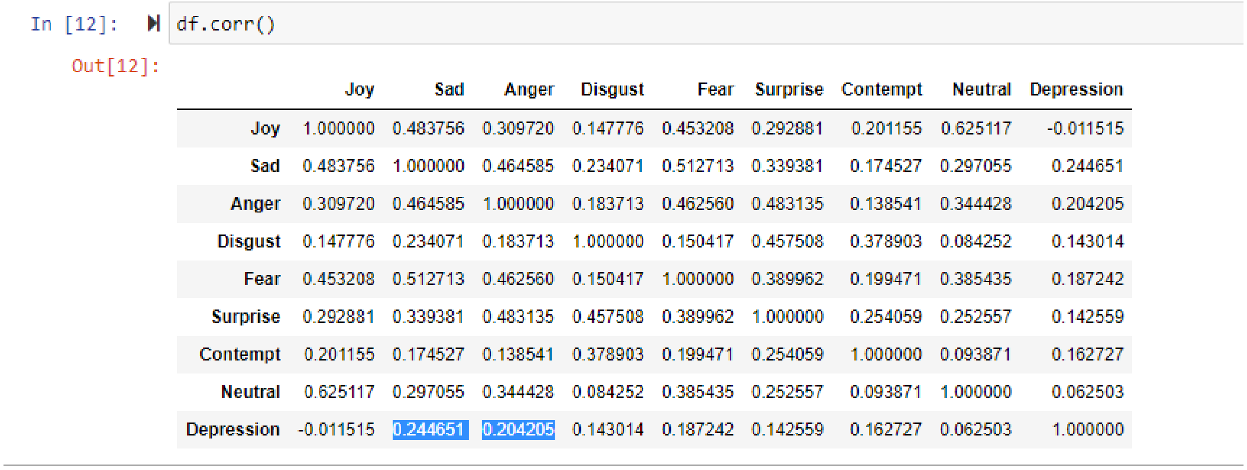
5-fold	**Accuracy**		
SVM	**Validation**	**Test**	**F1 score**
Polynomial Kernel	58%	73%	73%
RBF Kernel	57%	73%	73%
Sigmoid Kernel	57%	68%	68%

After select the machine algorithm, tune the gamma to improve the model performance. By changing the degree of gamma from 2 to 1, manage to improve the F1 score from 73% to 76% (result as below).



```
In [166]:  svmpoly = SVC(kernel = 'poly', degree = 1, gamma = 'scale')

           scores = cross_val_score(svmpoly, x_train, y_train, cv = 5, scoring = 'accuracy')
           scores

Out[166]:  array([0.65957447, 0.55319149, 0.65217391, 0.7173913 , 0.56521739])

In [167]:  print('Accuracy (Validation) =', scores.mean())

           Accuracy (Validation) = 0.6295097132284921

In [168]:  svmpoly.fit(x_train, y_train)
           test_predict = svmpoly.predict(x_test)
           print("Accuracy (Test): ", metrics.accuracy_score(y_test, test_predict))

           Accuracy (Test):  0.7627118644067796

In [169]:  print(confusion_matrix(y_test, test_predict))
           print(classification_report(y_test, test_predict))

           [[37  2]
            [12  8]]
                         precision    recall  f1-score   support

                      0       0.76      0.95      0.84        39
                      1       0.80      0.40      0.53        20

               accuracy                           0.76        59
              macro avg       0.78      0.67      0.69        59
           weighted avg       0.77      0.76      0.74        59
```

2.) Before start to run the dataset, drop the 'Gender' because this attribute did not contribute to model training. For Freq-PHO-Binary, KNN (Euclidean distance) is tested to have the highest accuracy if compare to other machine algorithms. For Norm-PHO-Binary, Decision Trees turn out to be the best model because it's having the highest validation accuracy.



```
In [12]:  df.corr()

Out[12]:
```

	Joy	Sad	Anger	Disgust	Fear	Surprise	Contempt	Neutral	Depression
Joy	1.000000	0.483756	0.309720	0.147776	0.453208	0.292881	0.201155	0.625117	-0.011515
Sad	0.483756	1.000000	0.464585	0.234071	0.512713	0.339381	0.174527	0.297055	0.244651
Anger	0.309720	0.464585	1.000000	0.183713	0.462560	0.483135	0.138541	0.344428	0.204205
Disgust	0.147776	0.234071	0.183713	1.000000	0.150417	0.457508	0.378903	0.084252	0.143014
Fear	0.453208	0.512713	0.462560	0.150417	1.000000	0.389962	0.199471	0.385435	0.187242
Surprise	0.292881	0.339381	0.483135	0.457508	0.389962	1.000000	0.254059	0.252557	0.142559
Contempt	0.201155	0.174527	0.138541	0.378903	0.199471	0.254059	1.000000	0.093871	0.162727
Neutral	0.625117	0.297055	0.344428	0.084252	0.385435	0.252557	0.093871	1.000000	0.062503
Depression	-0.011515	0.244651	0.204205	0.143014	0.187242	0.142559	0.162727	0.062503	1.000000

According to the correlation matrix, it showing that features like sad, anger and fear is having higher correlation with depression if compare to other variables. Among all of the features, sadness has the highest correlation with depression. Hence, use only feature of sadness will able to produce a better model in machine algorithm.

4.) Overall best performing model, by referring to F1 score, only achieve 70% for Norm-PHO-Binary dataset. It is even lower on F1 score for Freq-PHO-Binary dataset. To improve the performance of machine learning model, we can get more data samples. The more training data input to the algorithm, the more data it learns from, the algorithm would be workout more accurately. Secondly, understand which feature are the most important variable to the algorithm. Technique like Permutation Feature Importance (PFI) can be a method to identify the important feature and then help to reduce the noise. After preparation on the input of features, explore more other machine learning algorithm. Find the most fit algorithm model to dataset.