BHASKARACHARYA NATIONAL INSTITUTE FOR SPACE APPLICATIONS AND GEO-INFORMATICS

WEEKLY PROGRESS REPORT (06/03/2023 – 12/03/2023)

WEEK

PROJECT NAME	MALWARE DETECTION USING ML

PROJECT DESCRIPTION :	**DESIGN AND IMPLEMENT ML MODEL TO DETECT MALWARE IN SYSTEM**
GROUP MEMBER :	**PRATHAM PATEL, Shubham Patel, Shashank Sharma, Yash Soni**
GROUP ID :	**I2**
GROUP GUIDE :	**HARSH KIRATSATA**
GROUP COORDINATOR :	**SIDHDHARTH PATEL**
COLLEGE GUIDE NAME :	**MAYANK PATEL, RITIKA LADHA, KINJAL CHAUDHARI**
COLLEGE GUIDE No. :	**+91 9033280445,8866438698, 9723173567**
COLLEGE GUIDE EMAIL. :	**mayank.patel@adaniuni.ac.in, ritika.ladha@adaniuni.ac.in, kinjal.chaudhari@adaniuni.ac.in**
COLLEGE NAME :	**ADANI INSTITUTE OF INFRASTRUCTURE AND ENGINEERING**

06/03/2023 TILL 12/03/2023 (7 DAYS)

Using sk learn and linux.

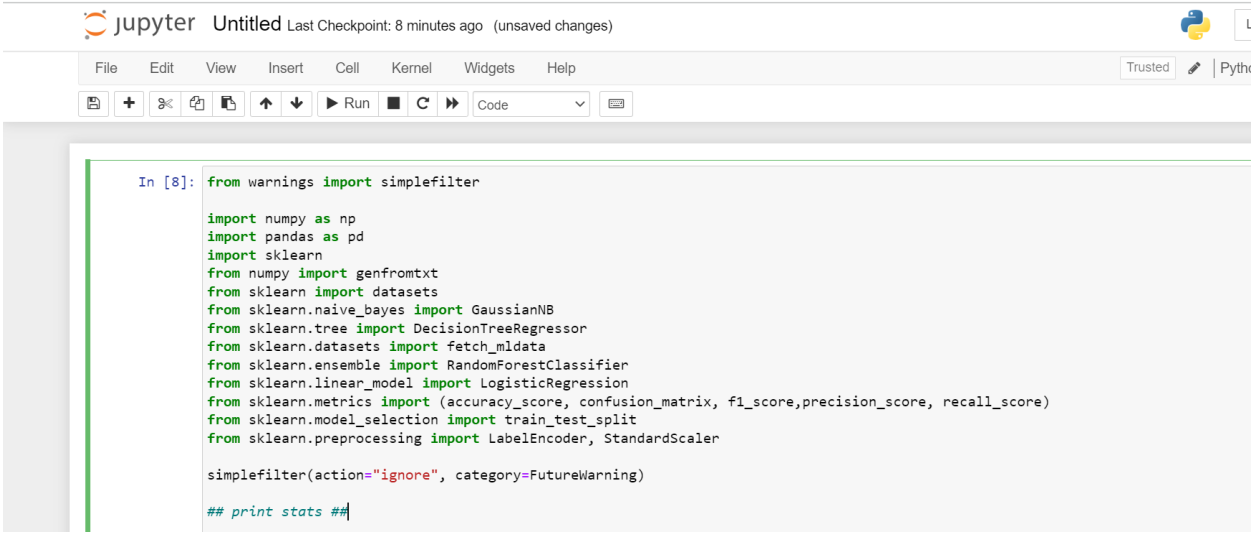
06/03/2023	**Exploring kali Linux and Jupiter in kali os.**
07/03/2023	**Using Simpelfilter for warnings and errors handling.**
08/03/2023	**Holiday (Dhureti).**
09/03/2023	**Writing code for Sk learn and statistics.**
10/03/2023	**Writing code for Logistic Regression and random Forest.**
11/03/2023	**Holiday (2nd Saturday).**
12/03/2023	**Holiday (Sunday).**

WEEK 8(PLAN)	**We are planning to implement our project in TensorFlow.**

REFERENCE :

- https://www.youtube.com/watch?v=fXtJekoqBeY&list=PL74sw1ohGx7FE-DI18bOfi2X61zRE-wMd&index=7
- https://www.tensorflow.org/
- https://www.geeksforgeeks.org/understanding-logistic-regression/
- https://www.javatpoint.com/machine-learning-random-forest-algorithm

Screenshots :




```
## print stats ##

def print_stats_metrics(y_test, y_pred):
    print('Accuracy: %.2f' % accuracy_score(y_test,y_pred) )
    confmat = confusion_matrix(y_true=y_test, y_pred=y_pred)
    print ("confusion matrix")
    print(confmat)
    print (pd.crosstab(y_test, y_pred, rownames=['True'], colnames=['Pred$
    print('Precision: %.3f' % precision_score(y_true=y_test, y_pred=y_pre$
    print('Recall: %.3f' % recall_score(y_true=y_test, y_pred=y_pred))
    print('Fl-measure: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
```

```
    print('F1-measure: %.3f' % f1_score(y_true=y_test, y_pred=y_pred))
```

In [12]:
```
feature = genfromtxt('log_file_features.csv', delimiter=',', \
                     usecols=(1 for 1 in range(1, 1001)), dtype=int, \
                       skip_header=1)

target =genfromtxt('log_file_features.csv',delimiter=',' dtype=int,
                    skip_header=1)

labels = LabelEncoder().fit_transform(target)

feature_normalized = StandardScaler().fit_transform(feature)
x_train, x_test, y_train, y_test = train_test_split(feature_normalized, \
                                                    labels, test_size=0.25, random_state=0)
```

In [13]:
```
#### Logisic reg ###
print('Logistic regression results:')
LogReg1 = LogisticRegression()
logReg1.fit(x_train, y_train)
predictions = logReg1.predict(x_test)
print_stats_metrics (y_test, predictions)
```

In [ ]:

```
Logistic regression results:
Accuracy: 0.97
confusion matrix
[[23  1]
 [ 0 12]]
Predicted   0   1  All
True
0          23   1   24
1           0  12   12
All        23  13   36
Precision: 0.923
Recall: 1.000
F1-measure: 0.960
```

```
In [14]: #### Random Forest ###
         print('Random Forest')
         Forest = RandomForestClassifier()
         Forest.fit(x_train,y_train)
         predictions = Forest.predict(x_test)
         print_stats_metrics(y_train.predictions)
```

```
Random Forest:
Accuracy: 1.00
confusion matrix
[[24  0]
 [ 0 12]]
Predicted   0   1  All
True
0          24   0   24
1           0  12   12
All        24  12   36
Precision: 1.000
Recall: 1.000
F1-measure: 1.000
```