Project 2 :

Implementing a Big Data Application

Case study : Predicting malicious URLs

อินเทอร์เน็ตน่าจะเป็นหนึ่งในสิ่งที่ยิ่งใหญ่ที่สุดในสมัยใหม่ มันช่วยเพิ่มการพัฒนาของมนุษยชาติ หลายบริษัท เช่น Google พยายามปกป้องเรา จากการฉ้อโกง ด้วยการตรวจจับเว็บไซต์ที่เป็นอันตรายสำหรับเรา การทำเช่นนี้ไม่ใช่เรื่องง่ายเพราะอินเทอร์เน็ตมีหน้าเว็บไซต์หลายพันล้าน

Acquiring the URL data

ข้อมูลที่นำมาใช้ทดสอบ(URL Data set) มีทั้งหมด 420,465 URL

ซึ่งข้อมูลชุดนี้มี 2 คุณสมบัติ คือ url และ label ดังรูป

```
url,label
diaryofagameaddict.com,bad
espdesign.com.au,bad
iamagameaddict.com,bad
kalantzis.net,bad
slightlyoffcenter.net,bad
toddscarwash.com,bad
tubemoviez.com,bad
ipl.hk,bad
crackspider.us/toolbar/install.php?pack=exe,bad
pos-kupang.com/,bad
rupor.info,bad
svision-online.de/mgfi/administrator/components/com_babackup/classes/fx29id1.txt,bad
officeon.ch.ma/office.js?google_ad_format=728x90_as,bad
sn-gzzx.com,bad
sunlux.net/company/about.html,bad
outporn.com,bad
timothycopus.aimoo.com,bad
xindalawyer.com,bad
freeserials.spb.ru/key/68703.htm,bad
deletespyware-adware.com,bad
orbowlada.strefa.pl/text396.htm,bad
ruiyangcn.com,bad
zkic.com,bad
adserving.favorit-network.com/eas?camp=19320;cre=mu&grpid=1738&tag_id=618&nums=FGApbjFAAA,bad
cracks.vg/d1.php,bad
juicypussyclips.com,bad
```

Needed software Tools/Concepts

Logistic Regression, Stochastic Gradient Descent Classifier, Passive Aggressive Classifier, SciKit-learn, Python

Machine Learning Packages

“

```
# Machine Learning Packages
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.linear_model import SGDClassifier
from sklearn.linear_model import PassiveAggressiveClassifier
from sklearn.metrics import classification_report
```

”

Load URL Data Set

```
# Load Url Data
urls_data = pd.read_csv("./dataset.csv")


type(urls_data)

urls_data.head()


def maketokens(f):
    tkns_byslash = str(f.encode('utf-8')).split('/')    # make tokens after splitting by slash
    total_tokens = []
    for i in tkns_byslash:
        tokens = str(i).split('-')  # make tokens after splitting by dash
        tkns_bydot = []
        for j in range(0, len(tokens)):
            temp_tokens = str(tokens[j]).split('.') # make tokens after splitting by dot
            tkns_bydot = tkns_bydot + temp_tokens
        total_tokens = total_tokens + tokens + tkns_bydot
    total_tokens = list(set(total_tokens))  #remove redundant tokens
    if 'com' in total_tokens:
        total_tokens.remove('com')  #removing .com since it occurs a lot of times and it should not be
    return total_tokens
```

Split Test

แบ่งข้อมูลออกเป็น 2 ชุด

```
# Store vectors into X variable as Our X Features
X = vectorizer.fit_transform(url_list)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Training data สำหรับสร้างโมเดล และ

Testing data สำหรับทดสอบโมเดล

Using Logistic Regression

เพื่อทำนายว่า จะเกิดเหตุการณ์หนึ่งขึ้นหรือไม่หรือมี โอกาสเกิดขึ้นมากน้อยเพียงใด โดยมีการกำหนดค่าตัวแปรตัวหนึ่งหรือหลายตัวที่คาดว่าจะส่งผลต่อการเกิดเหตุการณ์นั้นๆ ใช้ลักษณะของตัวแปรตามเป็นตัวกำหนด

ซึ่งลักษณะของตัวแปรตาม (y) มีเพียงสองกลุ่ม คือ Bad / Good

```
#using logistic regression
logit = LogisticRegression()
logit.fit(X_train, y_train)
```

Using Stochastic Gradient Descent Classifier

วิธีการเคลื่อนลงตามความชันแบบสุ่ม เป็นวิธีการสำคัญที่ใช้ในปรับค่าพารามิเตอร์ ในการเคลื่อนลงตามความชันแบบธรรมดานั้น การเปลี่ยนแปลงค่าของพารามิเตอร์จะขึ้นกับอัตราการเรียนรู้ ซึ่งจะคงที่ตลอดไม่ว่าจะวนซ้ำเพื่อฝึกไปกี่ครั้ง ต่อมากมีคนพยายามหาวิธี ที่ทำให้การปรับค่าพารามิเตอร์ขึ้นกับปัจจัยต่างๆมากขึ้น

```
#using Stochastic Gradient Descent Classifier
SGD = SGDClassifier()
SGD.fit(X_train, y_train)
```

Using Passive Aggressive Classifier

เป็นการจำแนกโดยสมมติให้ dataset:

$$\begin{cases} X = \{\bar{x}_0, \bar{x}_1, \dots, \bar{x}_t, \dots\} \; where \; \bar{x}_i \in \mathbb{R}^n \\ Y = \{y_0, y_1, \dots, y_t, \dots\} \; where \; y_i \in \{-1, +1\} \end{cases}$$

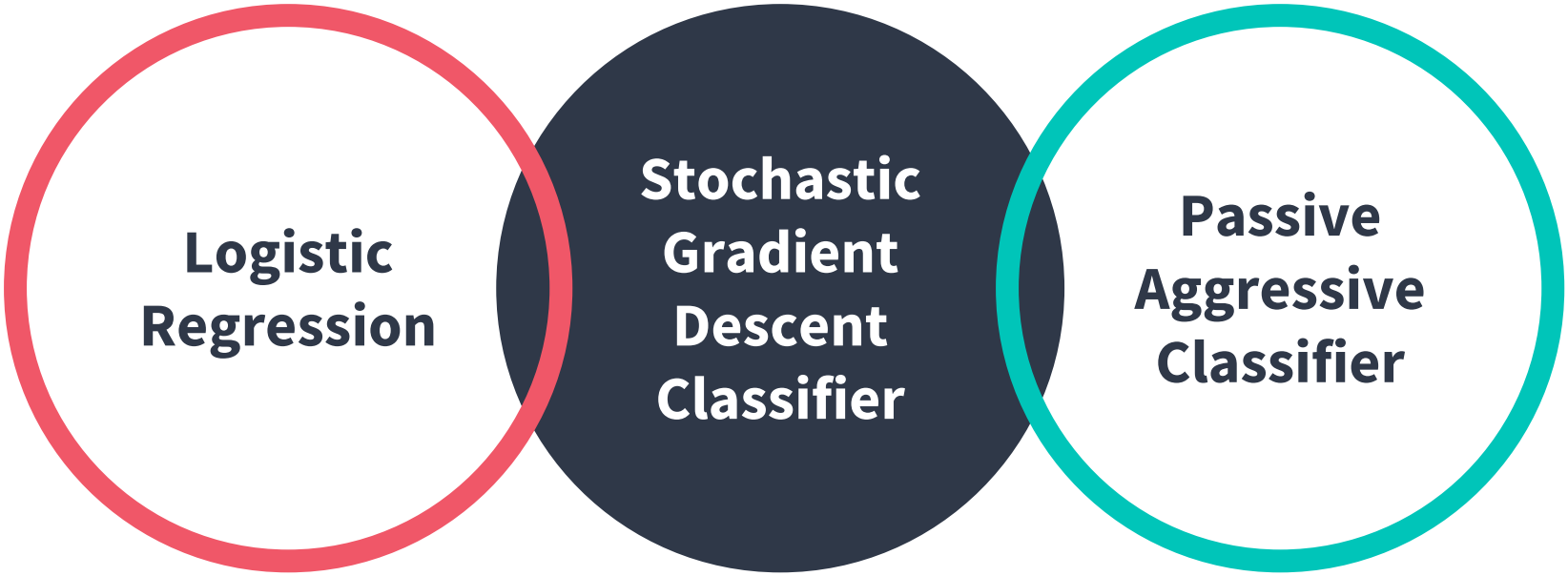
โดยมี t เป็นเวลา หรือ รอบที่มีการจำแนกข้อมูล และเปลี่ยนแปลงน้ำหนัก(เวกเตอร์ w) ซึ่งจะมีผลต่อการจำแนกและน้ำหนัก ในรอบที่ t+1 ดังนั้นจุดที่มีผลลัพธ์ในแง่ลบจะถูกจัดให้อยู่ในกลุ่ม -1 และผลลัพธ์ในแง่บวกจะถูกจัดให้อยู่ในกลุ่ม +1

```
#using Passive Aggressive Classifier
PAC = PassiveAggressiveClassifier()
PAC.fit(X_train, y_train)
```

Accuracy of Our Model

```
# Accuracy of Our Model
print(" LogisticRegression Accuracy ")
print(classification_report(logit.predict(X_test), y_test))
print(logit.predict(X_predict))
print(" SGDClassifier Accuracy ")
print(classification_report(SGD.predict(X_test), y_test))
print(SGD.predict(X_predict))
print(" PassiveAggressiveClassifier Accuracy ")
print(classification_report(PAC.predict(X_test), y_test))
print(PAC.predict(X_predict))
```

Accuracy of Our Model



Logistic Regression

	Precision	recall	f1-score	support
bad	**0.81**	**0.98**	**0.89**	**12432**
good	**1.00**	**0.96**	**0.98**	**71661**
avg. / total	**0.97**	**0.96**	**0.96**	**84093**

Stochastic Gradient Descent Classifier

	Precision	recall	f1-score	support
bad	**0.52**	**0.98**	**0.68**	**8071**
good	**1.00**	**0.90**	**0.95**	**76022**
avg. / total	**0.95**	**0.91**	**0.92**	**84093**

Passive Aggressive Classifier

	Precision	recall	f1-score	support
bad	**0.92**	**0.98**	**0.95**	**14161**
good	**1.00**	**0.98**	**0.99**	**69932**
avg. / total	**0.98**	**0.98**	**0.98**	**84093**

Accuracy of Our Model

```
 LogisticRegression
             precision    recall  f1-score   support

        bad       0.81      0.98      0.89     12432
       good       1.00      0.96      0.98     71661

avg / total       0.97      0.96      0.96     84093

 SGDClassifier
             precision    recall  f1-score   support

        bad       0.52      0.98      0.68      8071
       good       1.00      0.90      0.95     76022

avg / total       0.95      0.91      0.92     84093

 PassiveAggressiveClassifier
             precision    recall  f1-score   support

        bad       0.92      0.98      0.95     14161
       good       1.00      0.98      0.99     69932

avg / total       0.98      0.98      0.98     84093
```

Presenter

Thanisorn Carpholdee
57050070

Thanapon Kosallvitr
57050065

Thanakan Jeangdee
57050063

Thanks you!

for your attention