Detection of SQL Injection with a Machine Learning Approach

Presenter: Urmi Patel

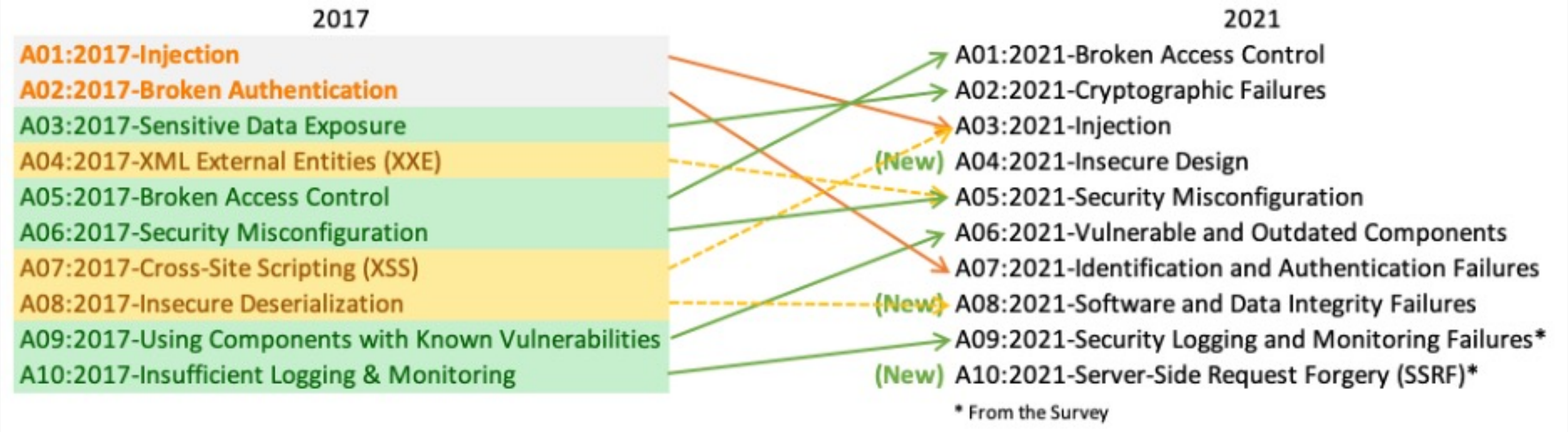
Subject: Cyber Security (CP8320)

Final Project Presentation

Table of Content

- SQL Injection
- Goal/ Motivation
- Dataset description
- Machine learning process
- Models / Algorithms
- Comparisons of various models
- Detailed Analysis
- Experiment and results
- Conclusion
- Future work and challenges

SQL Injection



2017

A01:2017-Injection
A02:2017-Broken Authentication
A03:2017-Sensitive Data Exposure
A04:2017-XML External Entities (XXE)
A05:2017-Broken Access Control
A06:2017-Security Misconfiguration
A07:2017-Cross-Site Scripting (XSS)
A08:2017-Insecure Deserialization
A09:2017-Using Components with Known Vulnerabilities
A10:2017-Insufficient Logging & Monitoring

2021

A01:2021-Broken Access Control
A02:2021-Cryptographic Failures
A03:2021-Injection
(New) A04:2021-Insecure Design
A05:2021-Security Misconfiguration
A06:2021-Vulnerable and Outdated Components
A07:2021-Identification and Authentication Failures
(New) A08:2021-Software and Data Integrity Failures
A09:2021-Security Logging and Monitoring Failures*
(New) A10:2021-Server-Side Request Forgery (SSRF)*

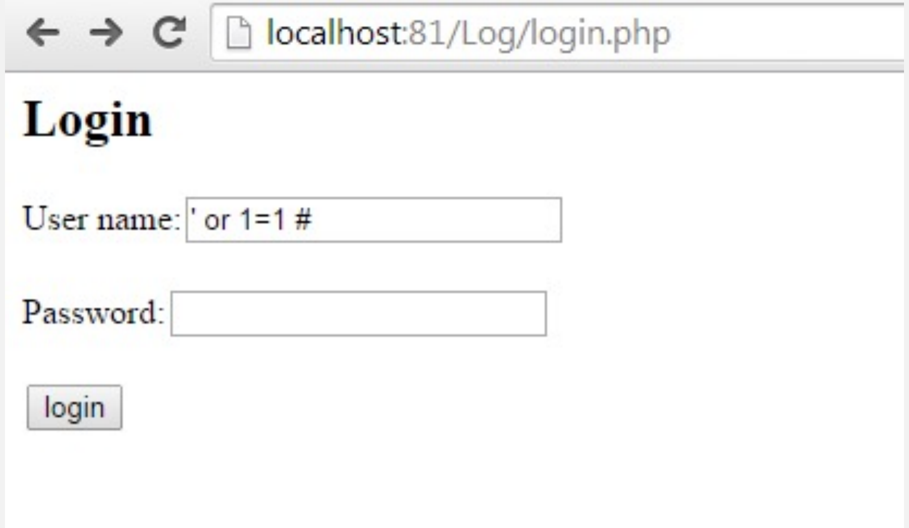
* From the Survey

- SQLI is a common type of attack that uses malicious SQL code for manipulating the database to access the data that was not intended to be exposed.

- Recent vulnerability reports found that web-based systems can receive up to 26 attacks/min.

Goal

- Develop a machine learning (ML) based classifier using supervised learning methods to identify whether the inputted data by user contains SQLI vulnerabilities or not!!!
- Tried various models and the best was chosen based on model accuracy.



localhost:81/Log/login.php

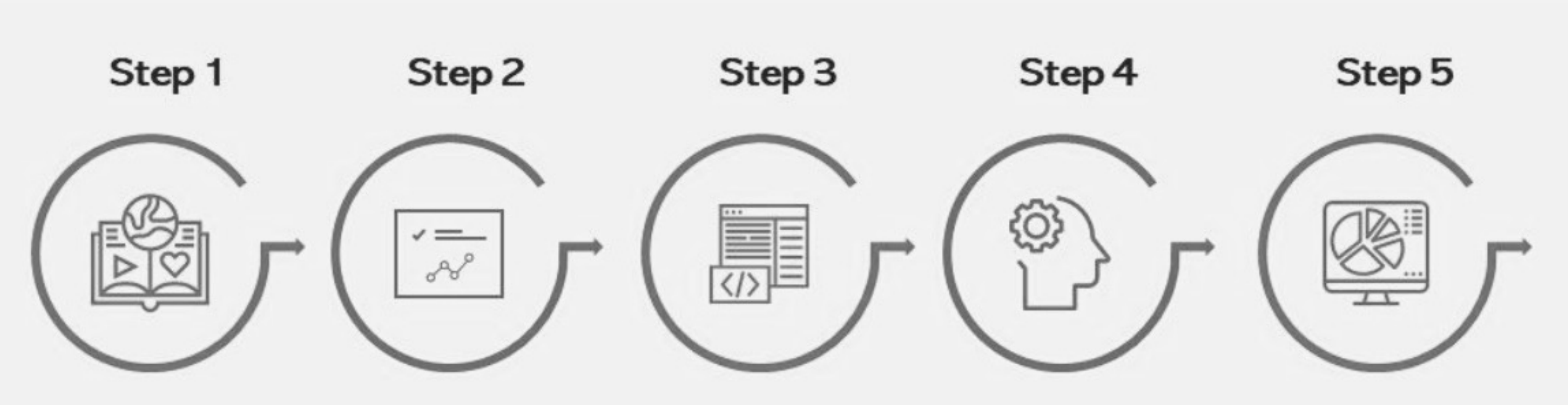
Login

User name: ' or 1=1 #

Password:

?

Machine Learning Process



Gathering Data

Preprocessing Data

Model building / selecting right method

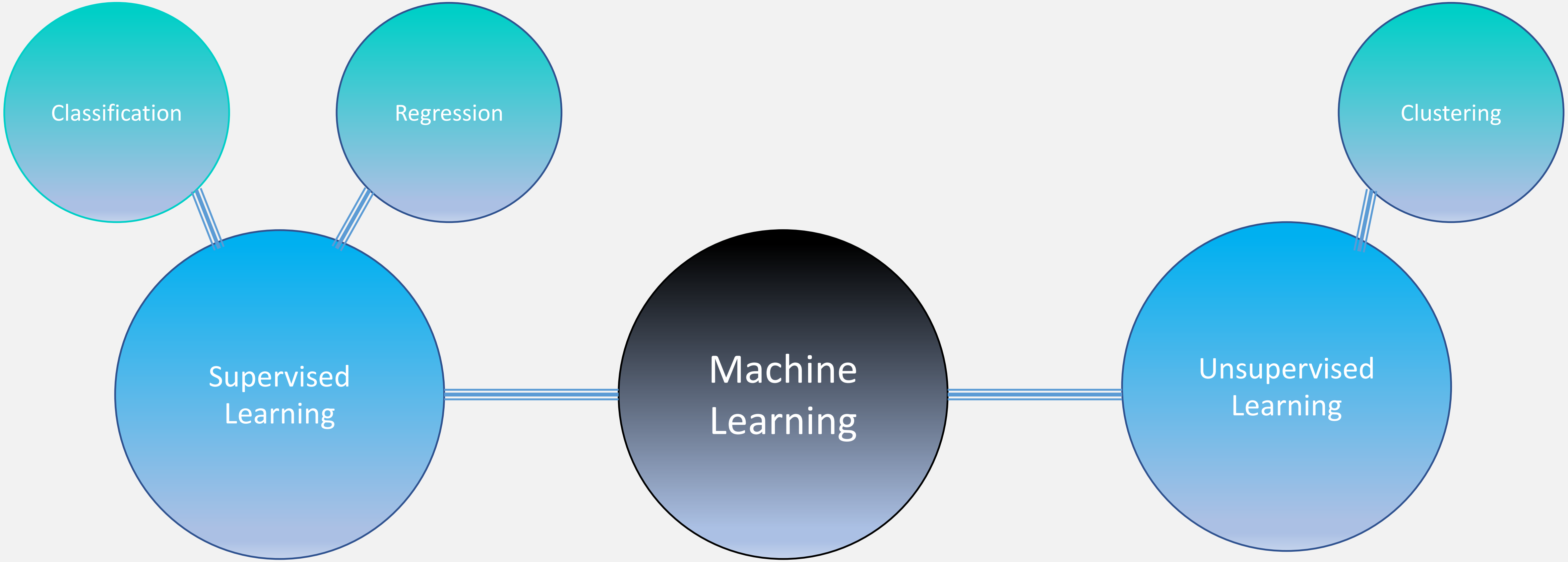
Model performance and Result

Data Visualization Graphs/ Figures

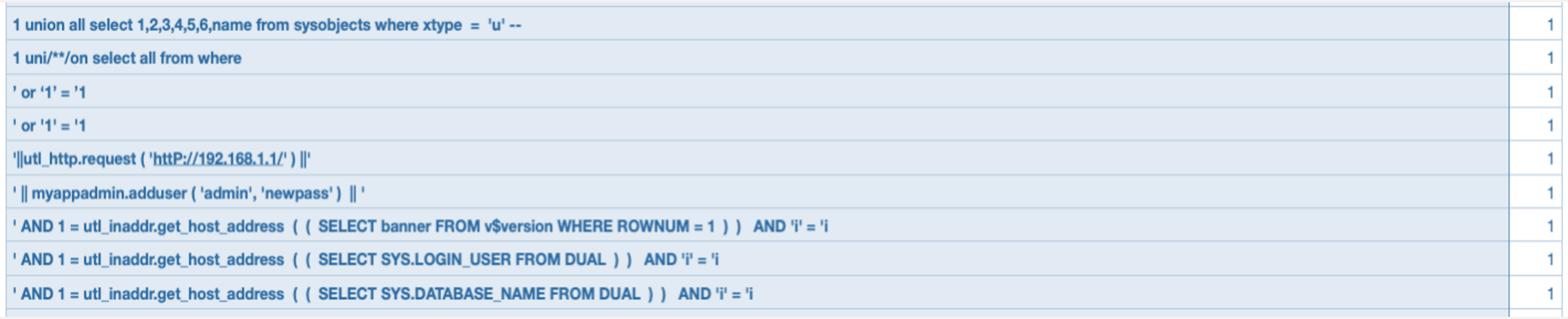
Tool : Google Colab

Language : Python

Machine Learning Approach



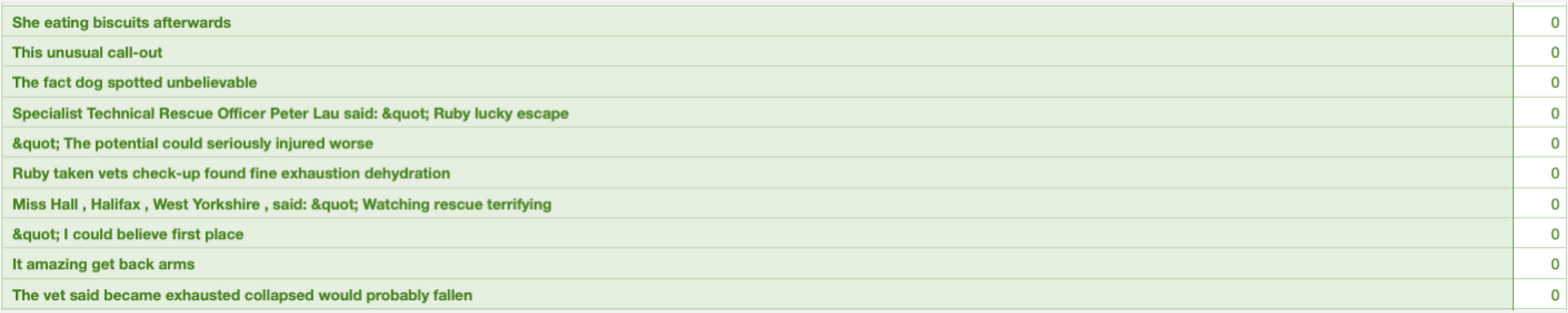
Dataset Description



1 union all select 1,2,3,4,5,6,name from sysobjects where xtype = 'u' --	1
1 uni/**/on select all from where	1
' or ‘1’ = ’1	1
' or '1' = '1	1
'||utl_http.request ('httP://192.168.1.1/') ||'	1
' || myappadmin.adduser ('admin', 'newpass') || '	1
' AND 1 = utl_inaddr.get_host_address ((SELECT banner FROM v$version WHERE ROWNUM = 1)) AND 'i' = 'i	1
' AND 1 = utl_inaddr.get_host_address ((SELECT SYS.LOGIN_USER FROM DUAL)) AND 'i' = 'i	1
' AND 1 = utl_inaddr.get_host_address ((SELECT SYS.DATABASE_NAME FROM DUAL)) AND 'i' = 'i	1

SQL Code – 1

Non-SQL Code - 0



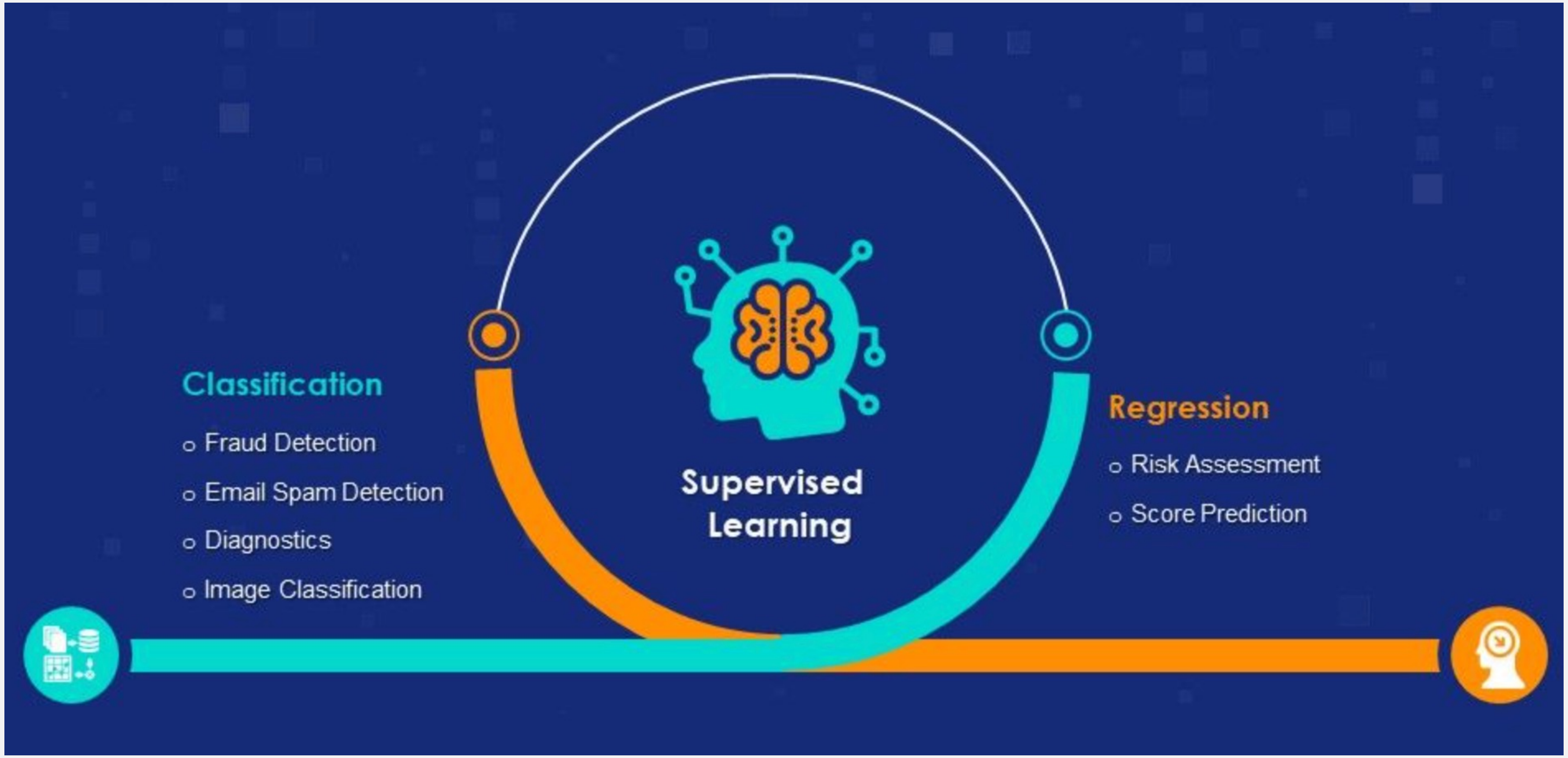
She eating biscuits afterwards	0
This unusual call-out	0
The fact dog spotted unbelievable	0
Specialist Technical Rescue Officer Peter Lau said: &quot; Ruby lucky escape	0
&quot; The potential could seriously injured worse	0
Ruby taken vets check-up found fine exhaustion dehydration	0
Miss Hall , Halifax , West Yorkshire , said: &quot; Watching rescue terrifying	0
&quot; I could believe first place	0
It amazing get back arms	0
The vet said became exhausted collapsed would probably fallen	0

Dataset Description

Sentences	Label
SQL or Non-SQL	1 or 0

- Total two columns and 4200 rows.
- Each row has SQL or Non-SQL sentences.
- Use 1 for SQL sentences and 0 for Non-SQL sentences.

Supervised Learning Method



Preprocessing Data

Removing NULL values from the dataset

Removing duplicate sentences from the dataset

Perform vectorization

Used scikit-learn library

Transform text into a vector on the basis of frequency of each word

Split training and testing data

80% for training

20% for testing

Models

- Naïve Bayes
- SVM
- KNN
- Decision Tree
- Logistic Regression
- CNN

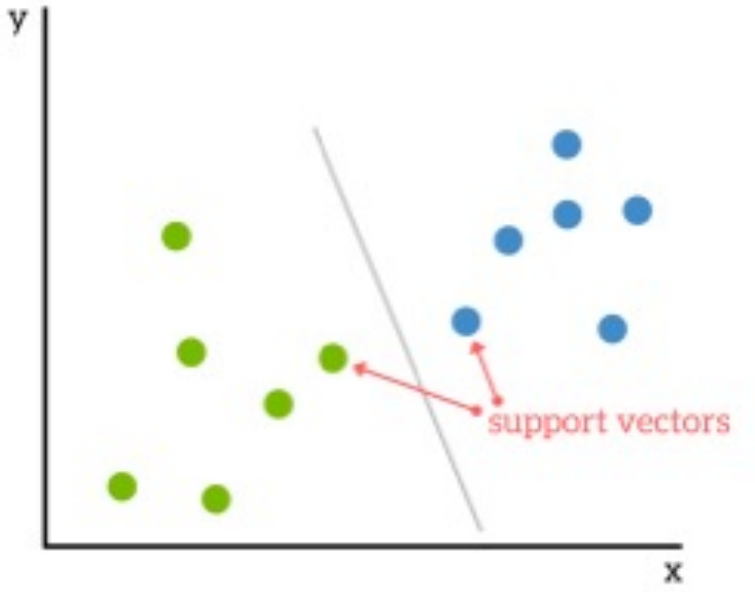
Models

Naïve Bayes

- Simple and most effective Classification algorithm.
- Building fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

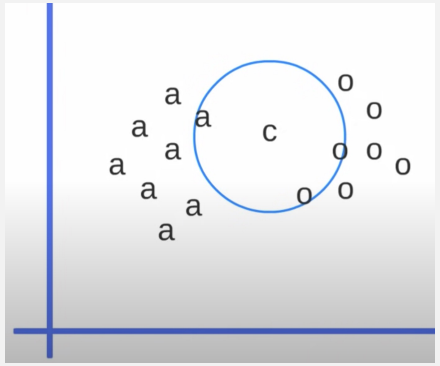
SVM

- SVM(support vector machine) is a margin-based classifier.
- SVM maps training examples points in a space and creates a line between them based on the calculation.
- New test element mapped into that same space and predicted to belong to a category based on which side they fall.



SVM Graph

Models



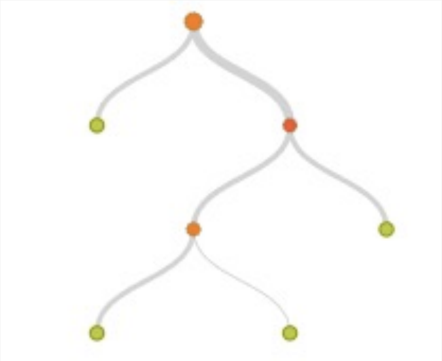
KNN Graph

KNN

- K nearest neighbor
- Identify K nearest neighbour of "c".
- Similar things are near to each other.
- Generally, neighbors share similar characteristics and behavior that's why they can be treated as they belong to the same group.

Decision Tree

- Decision tree builds classification or regression models in the form of a tree structure.
- It breaks down a dataset into smaller and smaller subsets.

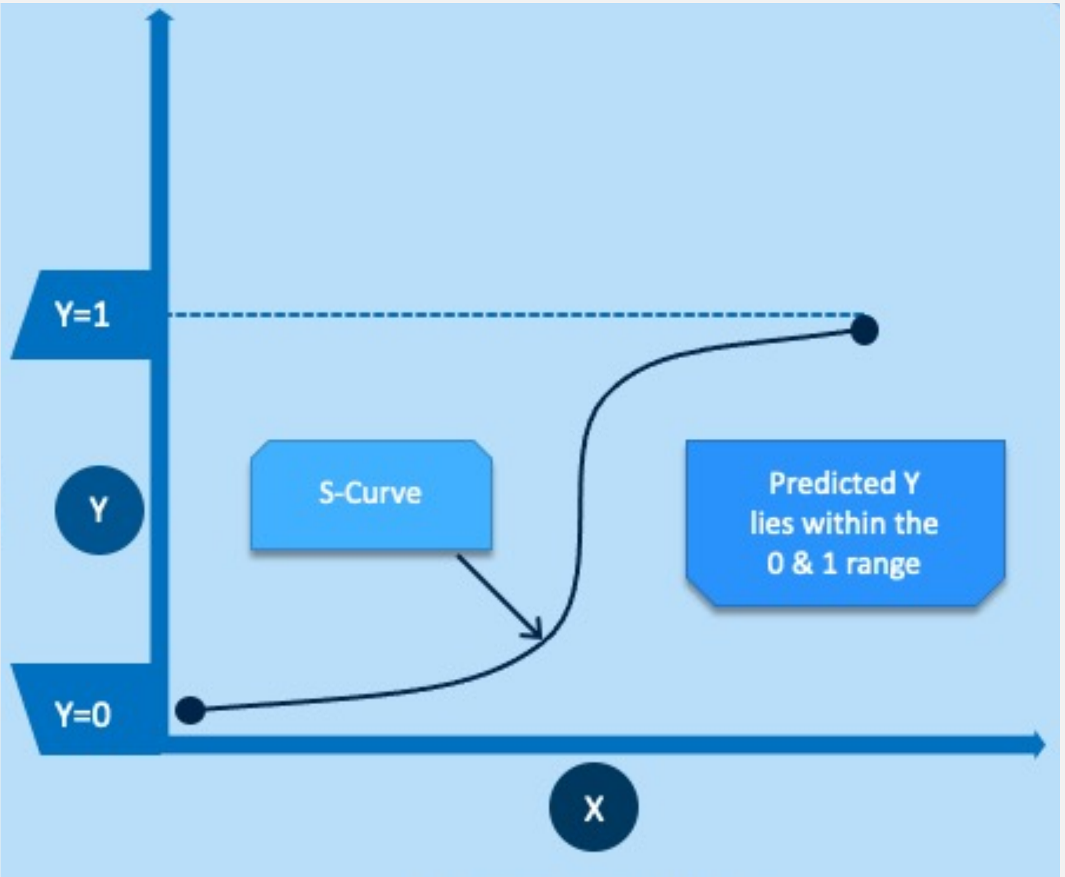


Tree Graph

Models

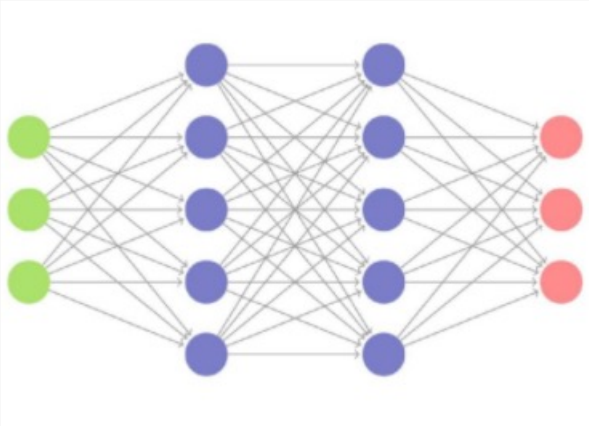
Logistic Regression

- The model is used for binary classification (1 or 0)
- Logistic regression models the data using the sigmoid function.
- There is a fixed threshold value pre-decided for each scenario.
- If the probability is greater than 0.5, the predictions will be classified as class 1. Otherwise, class 0 will be assigned.



Models - CNN (Convolutional Neural Network)

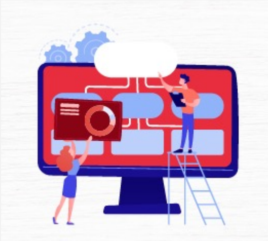
- Neural networks made with layers of neurons which are core processing units of the network.
- CNN has hidden layers called convolutional layers, it also has non – convolutional layers too.
- Neural networks takes a data as an input, trains them to recognize the pattern and then predicts the output for a new set of similar data.
- Neural networks may take hours or days to train the model.



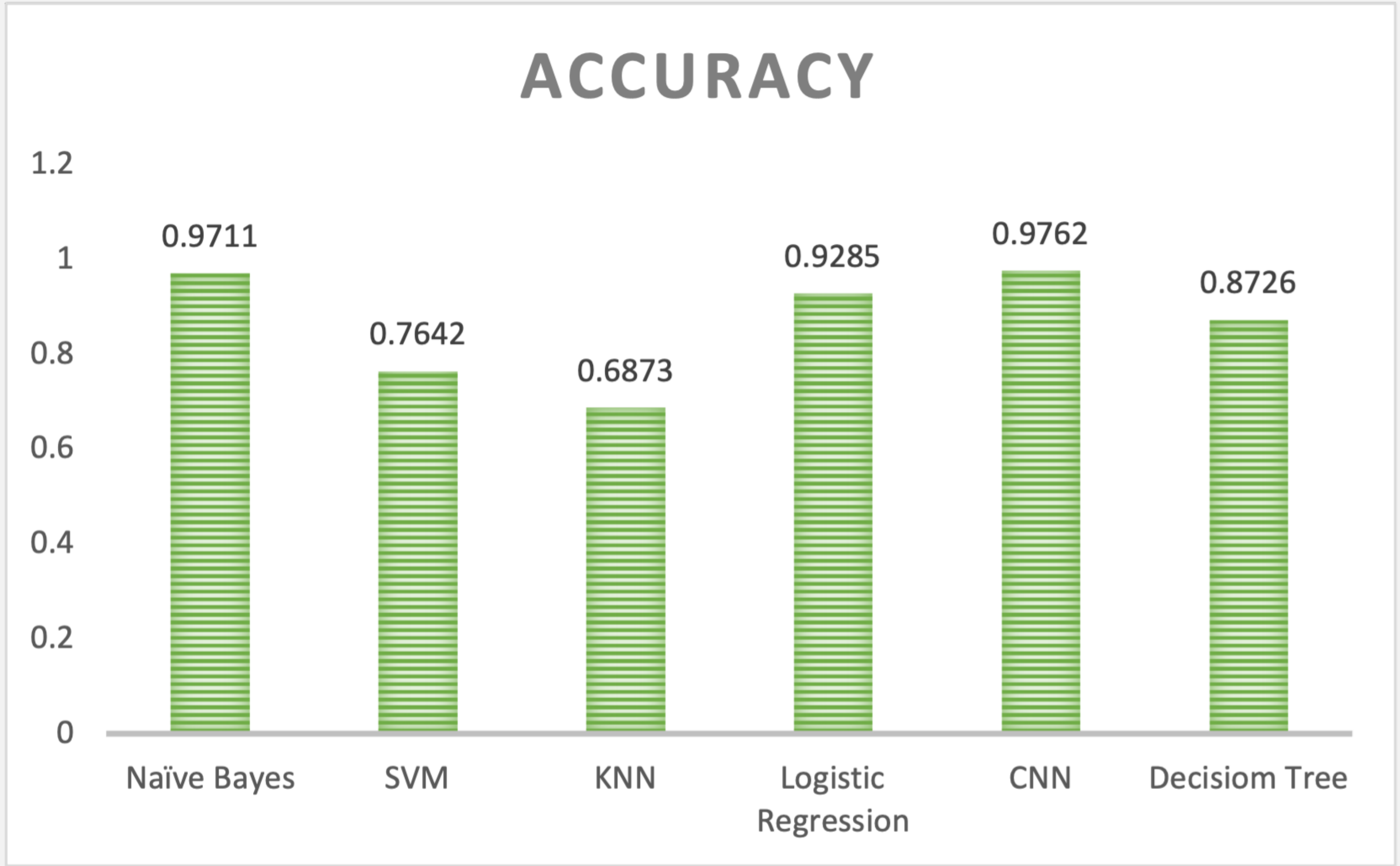
Hyper Parameter Tunning in CNN

Epoch	Batch Size	Accuracy
10	16	0.9642
10	32	0.9762
5	32	0.9533
10	44	0.9361
15	40	0.9702

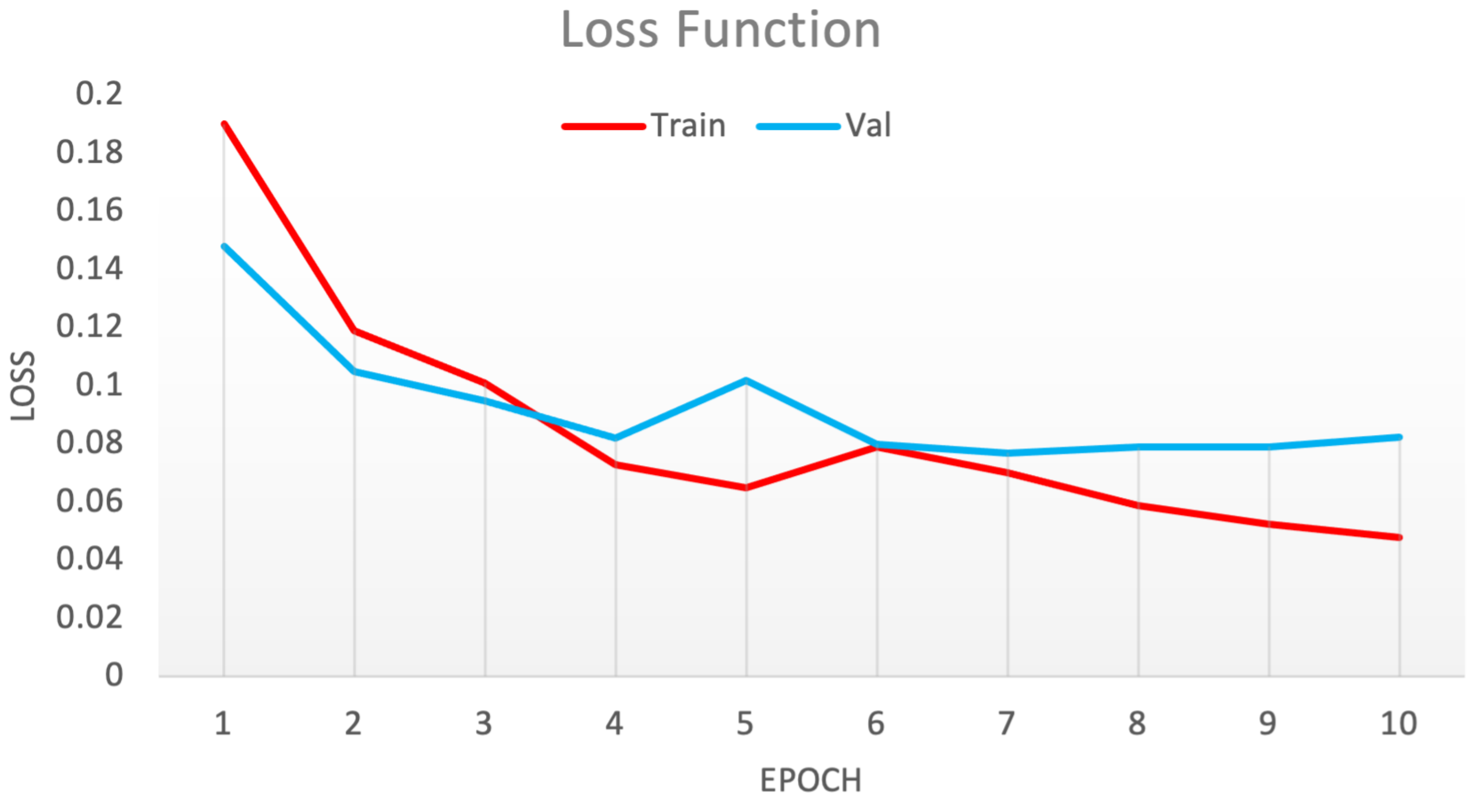
1. Dataset = 4000 rows
2. Split 80-20 = 3200(training) and 800 (testing)
3. Batch size = 32 then 3200 / 32 = 100 (each epoch take 100 rows)
4. Shuffled dataset every time
5. If epoch = 10 then (epoch 1 -> 100, epoch 2 -> 100, … up to epoch 10)



Results

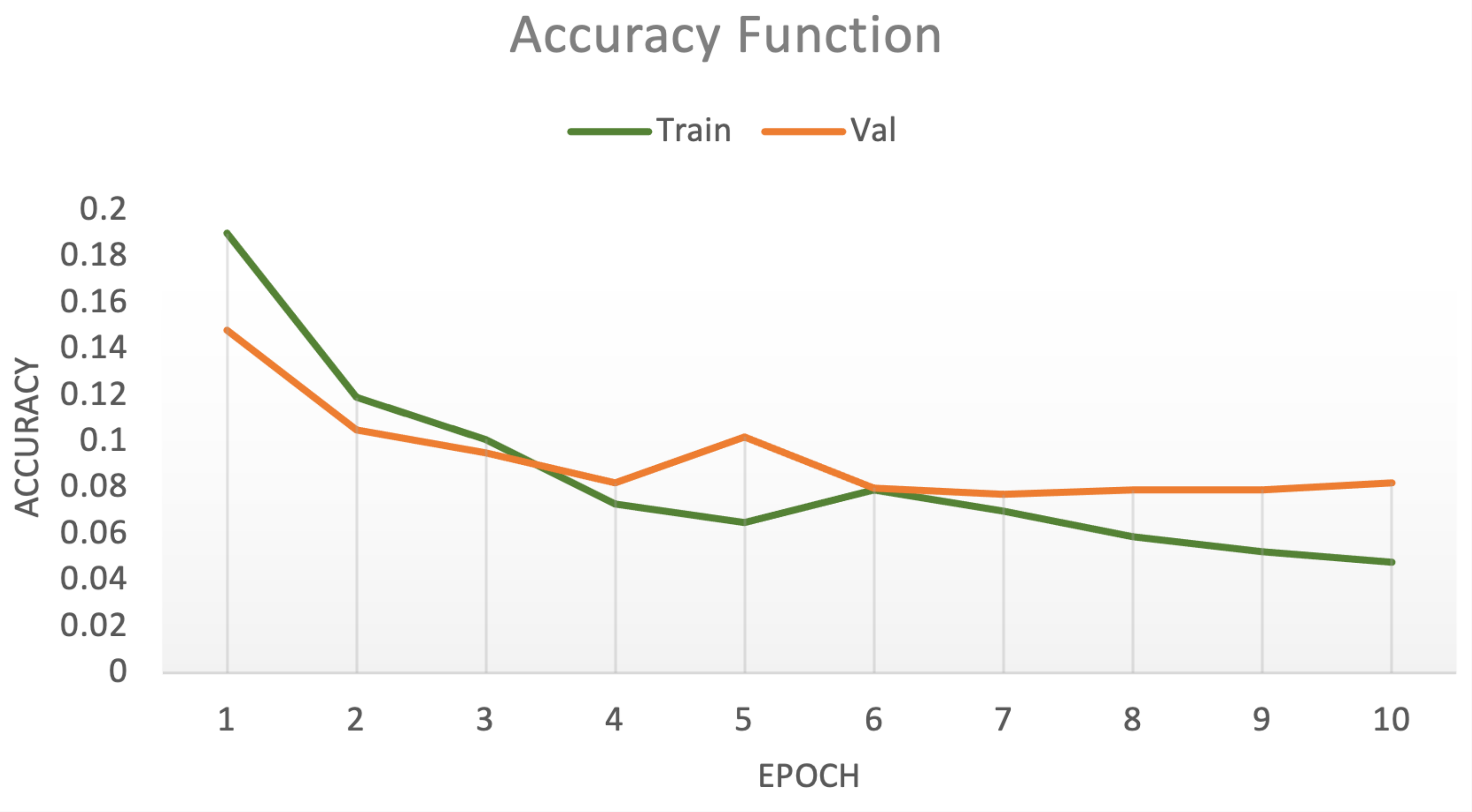

ACCURACY
1.2
1
0.8
0.6
0.4
0.2
0
0.9711
0.7642
0.6873
0.9285
0.9762
0.8726
Naïve Bayes
SVM
KNN
Logistic Regression
CNN
Decisiom Tree

Loss Curve for CNN

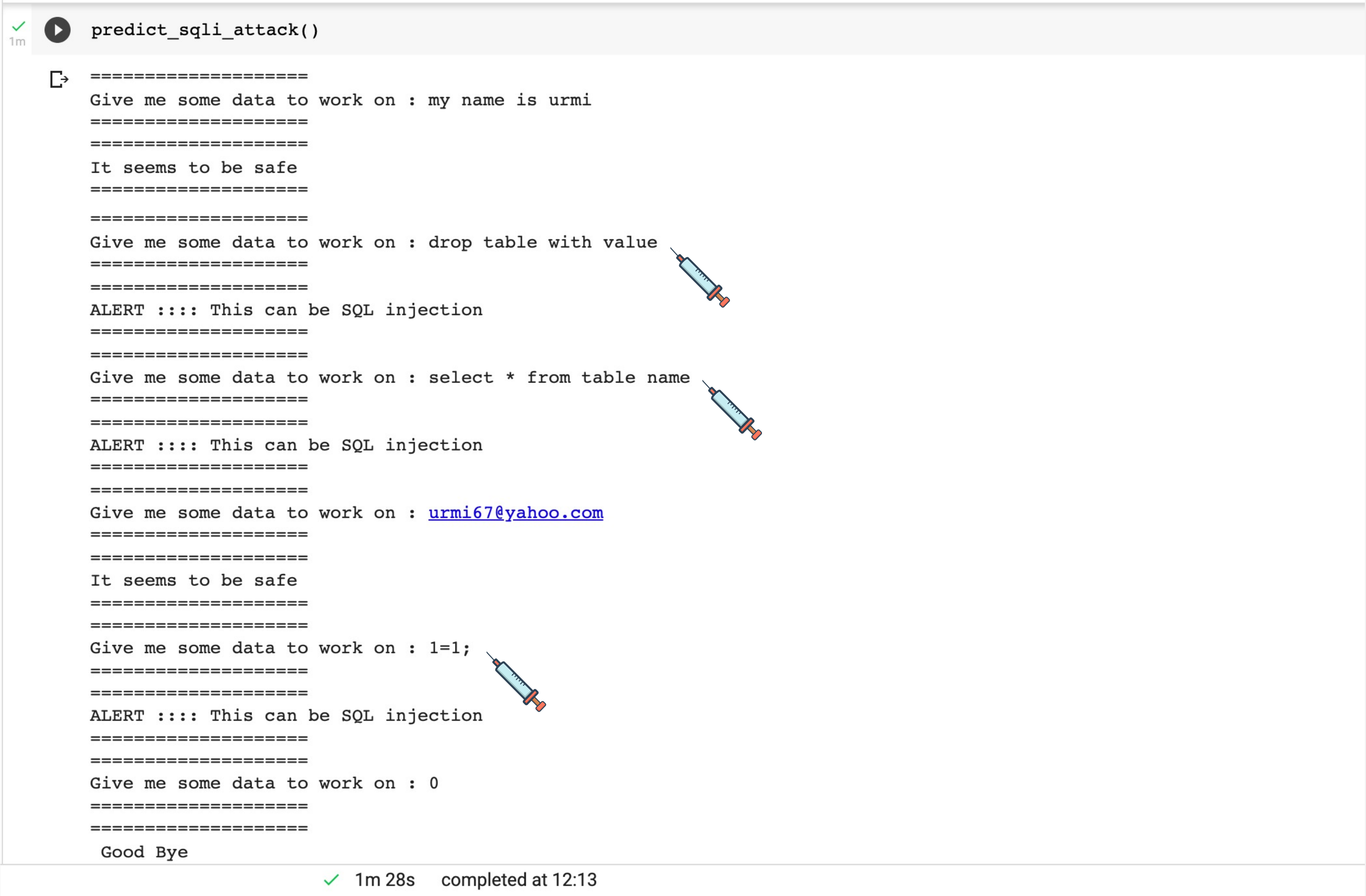


Epoch 1 -> 100 rows , calculate loss for each row and average them to calculate loss for 1 epoch

Accuracy Curve for CNN



Experiment



```
predict_sqli_attack()
```

```
====================
Give me some data to work on : my name is urmi
====================
====================
It seems to be safe
====================

====================
Give me some data to work on : drop table with value
====================
====================
ALERT :::: This can be SQL injection
====================
====================
Give me some data to work on : select * from table name
====================
====================
ALERT :::: This can be SQL injection
====================
====================
Give me some data to work on : urmi67@yahoo.com
====================
====================
It seems to be safe
====================
====================
Give me some data to work on : 1=1;
====================
====================
ALERT :::: This can be SQL injection
====================
====================
Give me some data to work on : 0
====================
====================
 Good Bye
```

✓ 1m 28s completed at 12:13

Conclusion

- Used Machine learning methods to identify whether the inputted data by user contained SQLI vulnerabilities or not!!!
- Performed various supervised learning algorithms and neural networks.
- CNN proved to be the best algorithm for used dataset, gave highest accuracy.
- Experiment proved successful, giving correct responses.

Future work

- More algorithm and complex neural networks can be applied.
- Try NLP based BERT model to identify SQL injection.

Challenge

Each time model's accuracy and loss gives slightly different output because of its nature of shuffling data.

Time consuming when doing hyper parameter tunning.

Thank You