Name: Anjali Sharma

Email address: anjaleana@gmail.com

Contact number: 7607219812

Anydesk address:

Years of Work Experience: **3.6yrs**

Date: **29th Dec 2021**

Self Case Study -2: SQL Injection Dataset

"After you have completed the document, please submit it in the classroom in the pdf format."

Please check this video before you get started:
https://www.youtube.com/watch?time_continue=1&v=LBGU1_JO3kg

Overview

*** Write an overview of the case study that you are working on. ***(MINIMUM 200 words)*** ***

Introduction

SQL injection attacks are the most common attack on the web applications. It is among the top 10 security risks on the web applications. SQL injection is an attack in which attacker gives SQL tokens as an input when such input is used to form query, it changes the structure of the query and allows the attacker to get access to confidential data or modify database. SQL injection exploits confidentiality and integrity of the application

Most of the applications that we use every day are web- based application. With the growth of internet, we are performing various kind of transaction. Data are entered by users during transaction on web application or websites is stored in some kind of database. Relational database can be communicated with a language called Structure Query Language(SQL). Using SQL to launch attacks on databases and manipulate them to do what the user wants is a form of a web hacking technique called SQL Injection Attack.

Sql injection attack considered as the one among foremost vulnerability that exploits in terms of privacy exposure also as money loss. Sql injection attack is known SQLI, is a common attack vector that use malicious SQL code for manipulating the backend database to access information that is not intended to be displayed. The information may include bank details, company data, private data or any sensitive data.

Fig. 1 - The SQL Injection Attacks

SQL injection attacks can be broadly classified into following three categories

1. Union Based SQL Injection
2. Error Based SQL Injection
3. Blind SQL Injection

Union Based SQL Injection:

In union-based SQL injection, attacker uses the Union operator. This is achieved by another query in place of plain text and using union at beginning of query.

Let’s take a example of Students database, We want to search Students name from database. Following query is found

SQL Query : Select * from Students where name = ‘ABC’

However, a malicious user might enter the following SQL query to exploit the database

SQL Query : select * from Students where name = ‘ABC’ Union drop table Students

Using this approach, the second part of the query can be used to perform any desired unauthorized action on the database. This might end up in deleting the entire Students table.

Error Based SQL Injection

This is achieved by forcing the database to perform an action that can triggered an error in database. The user can get error generated output and use those error to get information on how to further manipulate the database by exploiting SQL query.

Blind SQL Injection

Blind SQL injection attack does not reveal any data directly from the database. Rather, the attacker closely examines indirect clues in behavior. Details within HTTP responses, blank webpages for certain user input, and how long it takes database to respond to certain user input are all things that can be clues depending on the goal of the attacker.

Source of data:

We have dataset with 30920 number of rows. It has labelled data 0 and 1.

Research-Papers/Solutions/Architectures/Kernels

*** Mention the urls of existing research-papers/solutions/kernels on your problem statement and in your own words write a detailed summary for each one of them. If needed you can include images or explain with your own diagrams. it is mandatory to write a brief description about that paper. Without understanding of the resource please don't mention it***

Research Paper URL

https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1727&context=etd_projects

file:///C:/Users/Admin/Downloads/ojsadmin,+1939.pdf

https://www.imperva.com/learn/application-security/sql-injection-sqli/

https://www.interscience.in/cgi/viewcontent.cgi?article=1026&context=ijcsi

Research Work

1. A number of solutions are proposed to prevent SQL injection attacks. Almost all the solution requires either the developer to check the injection parameters during development phase of the application or the application is modified to prevent SQL injection attacks. SQLrand protects from SQL injection by randomizing the SQL Statement, creating instances of the language that are unpredictable to attacker. It is implemented as database server proxy.

There are different -2 tool found in research paper as follows

WAVES Tool : It identifies all the input points where SQL injection attacks are possible . It uses machine learning algorithm to test SQL injection vulnerabilities.

CANDID Tool : It intended query by dynamically evaluating runs over benign candidate and detects attacks by comparing it against the structure of the actual query issued.

AMNESIA Tool : It combines static analysis and runtime monitoring. In static phase it builds the query model and in dynamic phase it checks the dynamic query with the static model. Our

approach is also based on static and dynamic analysis. AMNESIA tool is developed to prevent SQL injection attacks on Java. It uses Java String Analyzer for static analysis.

2. Program contains the structure of the query. Approach is to extract query structure from the programs using static analysis and at the runtime capture the dynamic query and validate it against the static query model. SQL injection attack will add tokens to the user input and hence change the query structure. If the dynamic query contains malicious code than it will not match the static query model then it will be rejected.

First Cut Approach

*** Explain in steps about how you want to approach this problem and the initial experiments that you want to do. ***(MINIMUM 200 words)*** ***

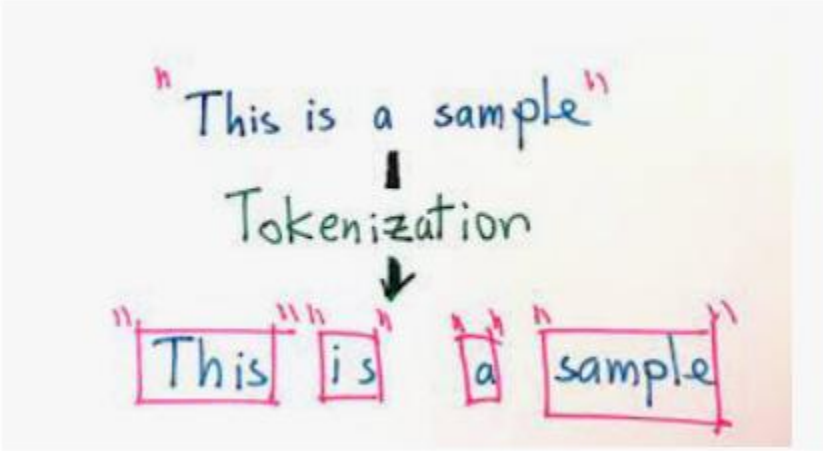
*** When you are doing the basic EDA and building the First Cut Approach you should not refer any blogs or papers ***

The dataset consists of plain-text query and has around 30920 rows. The dataset consists numerical data, special characters, spaces and textual data

Query	Label
" or pg_sleep (__TIME__) --	1
create user name identified by pass123 temporary tablespace temp default tablespace users;	1
AND 1 = utl_inaddr.get_host_address ((SELECT DISTINCT (table_name) FROM (SELECT DISTINCT (table_name) , ROWNUM AS LIMIT FROM sys.all_tables) WHERE LIMIT = 5))	1
select * from users where id = '1' or @ @1 = 1 union select 1,version ( ) -- 1'	1
select * from users where id = 1 or 1#" (union select 1,version ( ) -- 1	1
select name from syscolumns where id = (select id from sysobjects where name = tablename') --	1
select * from users where id = 1 +$+ or 1 = 1 -- 1	1
1; (load_file (char (47,101,116,99,47,112,97,115,115,119,100))) ,1,1,1;	1
select * from users where id = '1' or ||/1 = 1 union select 1,version ( ) -- 1'	1
select * from users where id = '1' or \.<\ union select 1,@@VERSION -- 1'	1
? or 1 = 1 --	1
) or ('a' = 'a	1
admin' or 1 = 1#	1
select * from users where id = 1 or " ( ]" or 1 = 1 -- 1	1
or 1 = 1 --	1
AND 1 = utl_inaddr.get_host_address ((SELECT DISTINCT (column_name) FROM (SELECT DISTINCT (column_name) , ROWNUM AS LIMIT FROM all_tab_columns) WHERE LIMIT = 5	1
select * from users where id = '1' %!<@ union select 1,version ( ) -- 1'	1
select * from users where id = 1 or "& (" or 1 = 1 -- 1	1
select * from users where id = 1 or "? (" or 1 = 1 -- 1	1
distinct	1
select * from users where id = '1' * (\ ) or 1 = 1 -- 1'	1
1 and ascii (lower (substring ((select top 1 name from sysobjects where xtype = 'u') , 1, 1))) > 116	1
select * from users where id = 1 or \ <\ or 1 = 1 -- 1	1

First, we need to do text pre- processing as follows:

1. **Tokenization** :
   In tokenization, a sequence of characters are broken into small pieces called ‘tokens’. Tokenization also includes removing certain characters sometimes



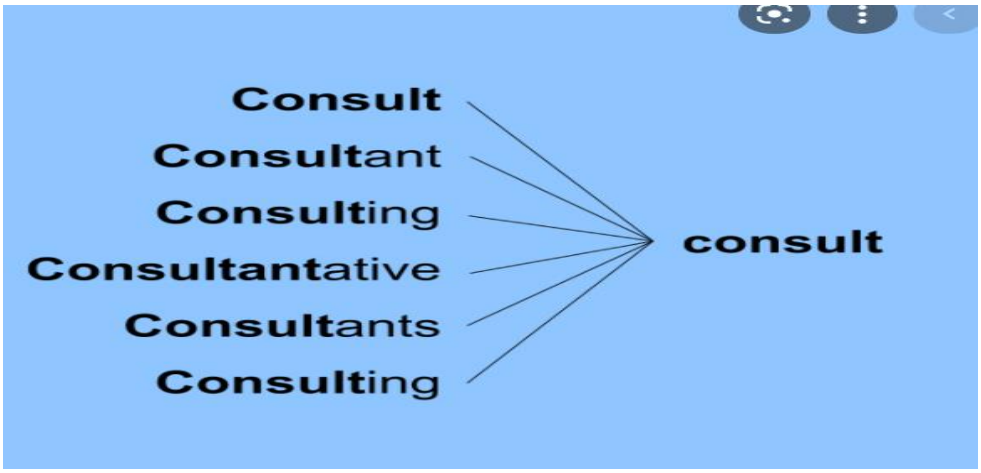
2. **Remove Punctuation:**
   We remove all the special characters e.g how are you? - > how are you

3. **Remove stop words:**
   Stop words are common words that will likely appear in any text. Which we need to remove.
   E.g. : silver or lead is fine for me -> silver, lead, fine.

Preprocessing Data :

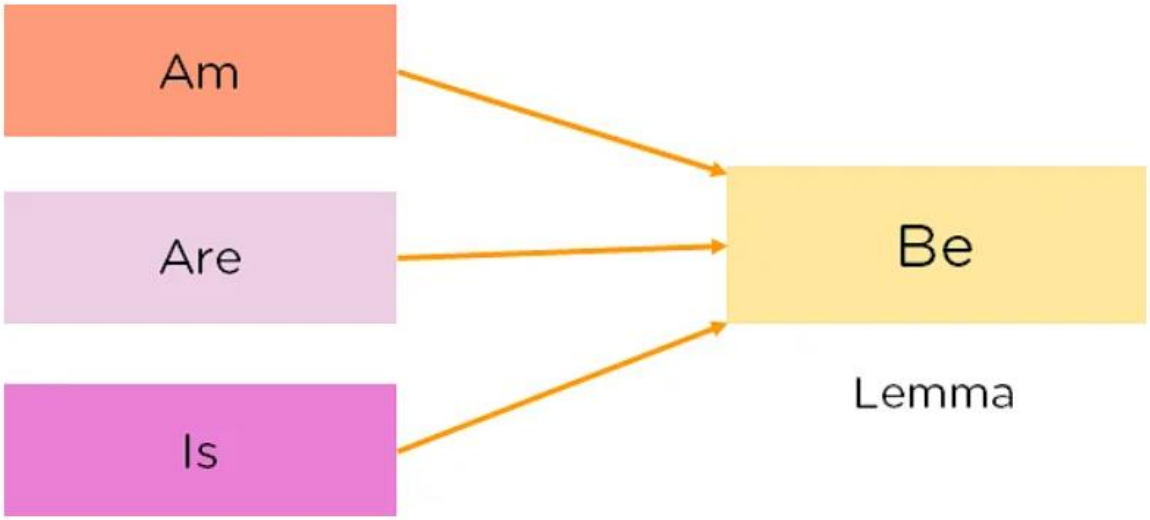
Stemming

Stemming helps reduce a word to its stem form. It removes the suffix, like “ing”, “ly”,”s” , etc by a simple rule based approah. E.g. Entitled, Entitling->Entitl



Lemmatizing :

Lemmatization also reduce the word but it takes dictionary-based approach. i.e a morphological analysis to the root word.eg: Entitling, Entitled->Entitle



Vectorizing Data:

Vectorizing is the process of encoding text as integers i.e. numeric form to create feature vectors so that machine learning algorithm can understand our data

Bag-of-Words:

Bag-of-words(BOW) or Count Vectorizer describes the presence of words within the text data. It gives a result of 1 if word present in text or 0 if not present

N- Grams :

N-grams are simply all combinations of adjacent words or letters of length n that we can find in our source text. Ngrams with n = 1 are called as unigrams. Similarly, bigrams(n=2), trigrams(n=3) and so on also can be used

N-Grams

"plata o plomo means silver or lead"

n	Name	Tokens
2	bigram	["plata o", "o plomo", "plomo means", "means silver", "silver or","or lead"]
3	trigram	["plata o plomo", "o plomo means", "plomo means silver", "means silver or ", "silver or lead"]

TF – IDF Vectorizer

TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm which covert text to numeric data which is used to fit machine learning algorithm for prediction.

TF – IDF = TF(d,t) + lDF(t)

TF(d,t) - >Number of times term t appears in document d

IDF(t) = log(1+n)/(1+df(d,t))+1

Where n = number of documents, df(d,t)-> Document frequency where term t appears.

We have used 7 model using hyper-parameters

1. Naïve Bayes Algorithm(Linear)

2. Logistic Regression(Linear)

3. Support -Vector Machine(Non-Linear)

4. Decision Tree(Non-Linear)

5. Gradient Boosting Algorithm(Non-linear)

6. Convolutional Neural Network

7. Stacking Classifier

We can also use forward Feature Selection to get important features

For feature selection, we have four types

1. Filter Methods
2. Wrapper Methods
3. Embedded Methods
4. Hybrid Methods

The Wrapper methods usually result in better predictive accuracy than filter method.

We used forward feature selection and Exhaustive Feature selection.

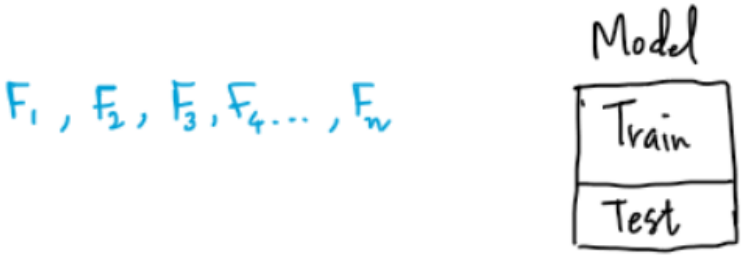
Feature Selection:

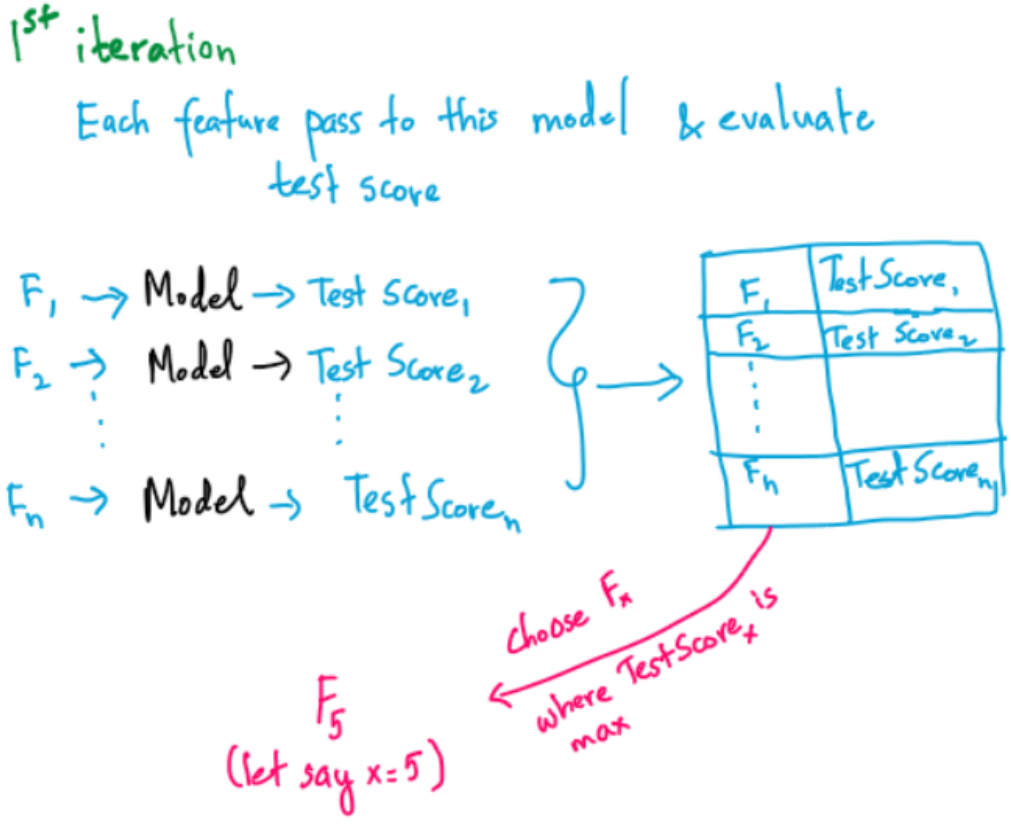
To find top 'n' importance features, First we need to train the model. At the same model, we used to find best features that can give more importance to that model. We used **forward feature selection.**

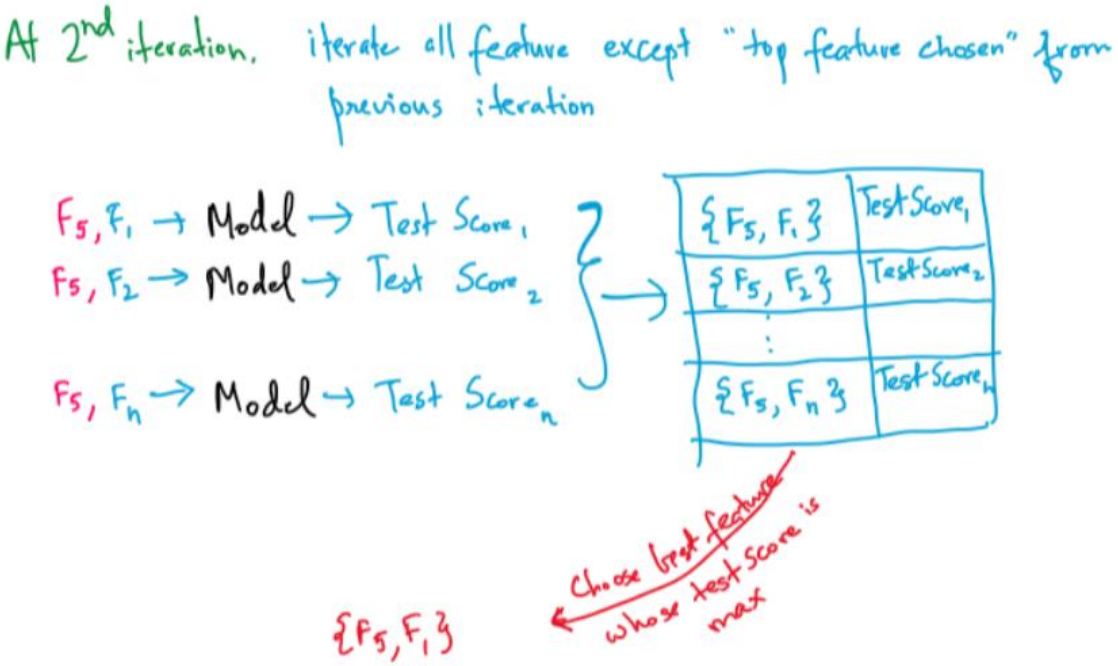
The idea is simple:

1. We need to initialize the number of top importance features of your desire.

2. Iterate over all the features in train data.
3. Stop when condition of your desire is fulfilled or cross-validation score stopped improving after adding new features.







Notes when you build your final notebook:

1. You should not train any model either it can be a ML model or DL model or Countvectorizer or even simple StandardScalar

2. You should not read train data files
3. The function1 takes only one argument "X" (a single data points i.e 1*d feature) and the inside the function you will preprocess data point similar to the process you did while you featurize your train data
    a. Ex: consider you are doing taxi demand prediction case study (problem definition: given a time and location predict the number of pickups that can happen)
    b. so in your final notebook, you need to pass only those two values
    c. def final(X):
        preprocess data i.e data cleaning, filling missing values etc
        compute features based on this X
        use pre trained model
        return predicted outputs
    final([time, location])
    d. in the instructions, we have mentioned two functions one with original values and one without it
    e. final([time, location]) # in this function you need to return the predictions, no need to compute the metric
    f. final(set of [time, location] values, corresponding Y values) # when you pass the Y values, we can compute the error metric(Y, y_predict)
4. After you have preprocessed the data point you will featurize it, with the help of trained vectorizers or methods you have followed for your train data
5. Assume this function is like you are productionizing the best model you have built, you need to measure the time for predicting and report the time. Make sure you keep the time as low as possible
6. Check this live session: https://www.appliedaicourse.com/lecture/11/applied-machine-learning-online-course/4148/hands-on-live-session-deploy-an-ml-model-using-apis-on-aws/5/module-5-feature-engineering-productionization-and-deployment-of-ml-models