TAGSUGGEST: A STACK OVERFLOW TAG RECOMMENDATION SYSTEM USING ONE-VS-REST MULTI-LABEL CLASSIFICATION

Presented by Trixia Belleza
Adviser: Concepcion Khan

Background of the Study

Stack Overflow

- Question and Answer forum for programmers
- Questions are organized in a form of tags

Tags

- Describe questions content
- For filter-searching of questions

STATEMENT OF THE PROBLEM

- Tagging of questions are done manually by users

- Users find it difficult to identify which category their question belongs to

STATEMENT OF THE PROBLEM

- Prone to human error
- Forms a disarray of information that could overturn its purpose

A TAG RECOMMENDATION SYSTEM
IS USEFUL BECAUSE...

Significance of the Study

It will save the user's time from thinking and assessing the tags that will actually fit their question.

A TAG RECOMMENDATION SYSTEM IS USEFUL BECAUSE...

It will increase the number of properly tagged questions which would improve the Stack Overflow's user-experience.

Significance of the Study

Significance of the Study

THE PERFORMANCE EVALUATION OF SVM, LR, AND MNB IS SIGNIFICANT BECAUSE...

It will assess which of the algorithms is most efficient and most accurate to use in a multi-label classification problem

OBJECTIVES OF THE STUDY

Generally, the study aims to develop an application that recommends tags to Stack Overflow questions using a multi-label classifier.

OBJECTIVES OF THE STUDY

1. To train and test 20K data using Support Vector Machine(SVM), Logistic Regression(LR), and Multinomial Naive Bayes(MNB)

2. To evaluate the performance of the classifiers by computing accuracy, precision, recall, and f1 score with a 60% f1 score threshold in classifying questions.

OBJECTIVES OF THE STUDY

3 To develop a chrome extension that recommends tags to Stack Overflow questions using the model with the highest overall performance among SVM, LR, and MNB

4 To evaluate the application to at least 30 users using a System Usability Scale and obtain an overall above average score (68 and above)

SCOPE AND LIMITATION OF THE STUDY

- Should be in English
- Actual Data Labels used are limited to:

Python, JavaScript, Java, C, R, MySQL, HTML, CSS, for-loop, while-loop, & if-statement

REVIEW OF RELATED LITERATURE

KRESTEL & FRANKHAUSTER

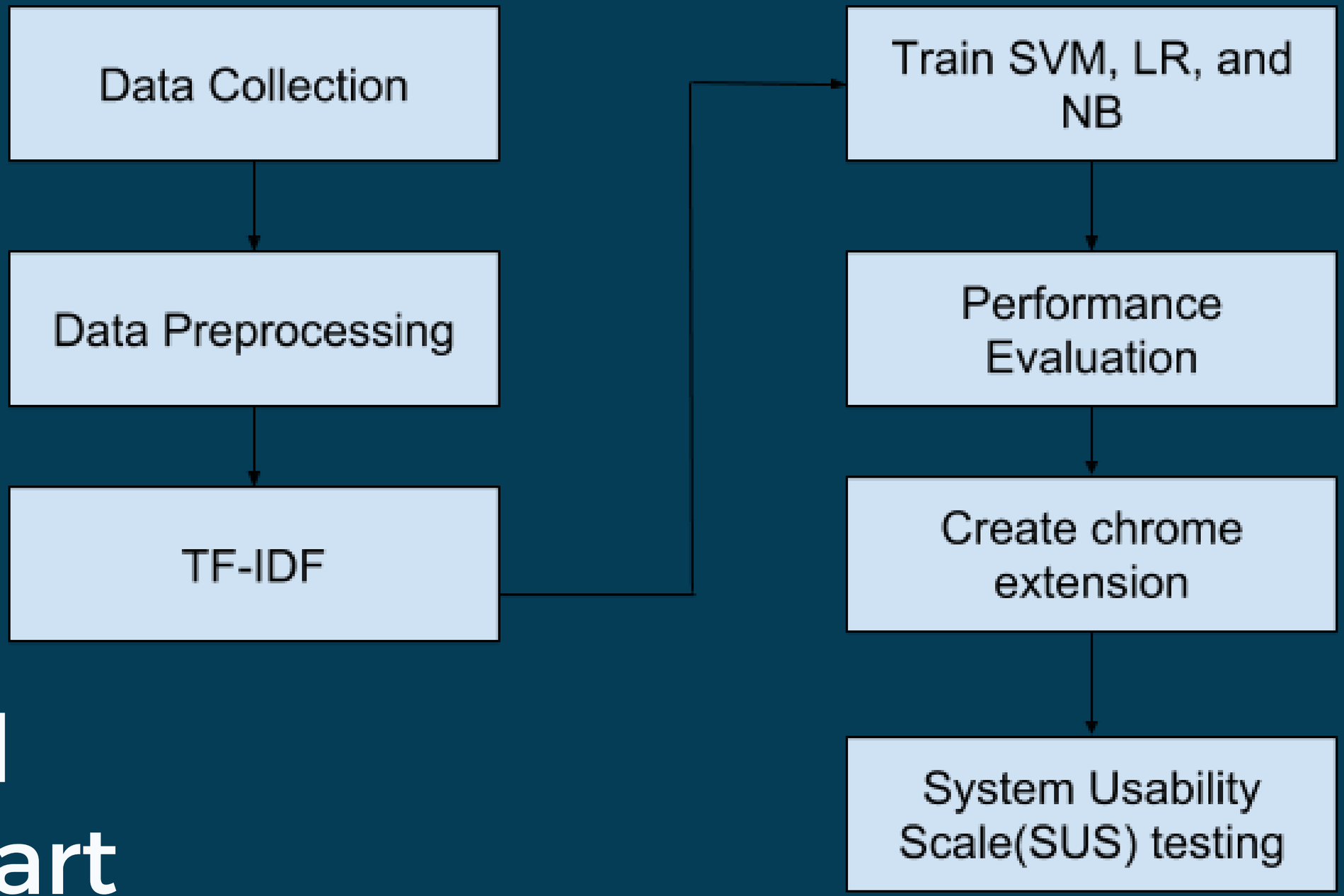
Tag recommendation is based on tagging history of the user.

SCHUSTER, ZHU, & CHENG

Computed a feature vector for each document-label pair which resulted to a very slow performance.

Materials & Methods

General Flowchart



DATA COLLECTION

- Web Scraped using the StackAPI
- Data was also manually provided with correct labels.

DATA PREPROCESSING

HTML Tags were removed

Punctuation marks such as [? | ! | , | ~ | ^] were removed

Stemming

USING NATURAL LANGUAGE PROCESSING TOOLKIT

TF

TERM FREQUENCY

If a word appears frequently in a document, then it's important. Give the word a high score.

IDF

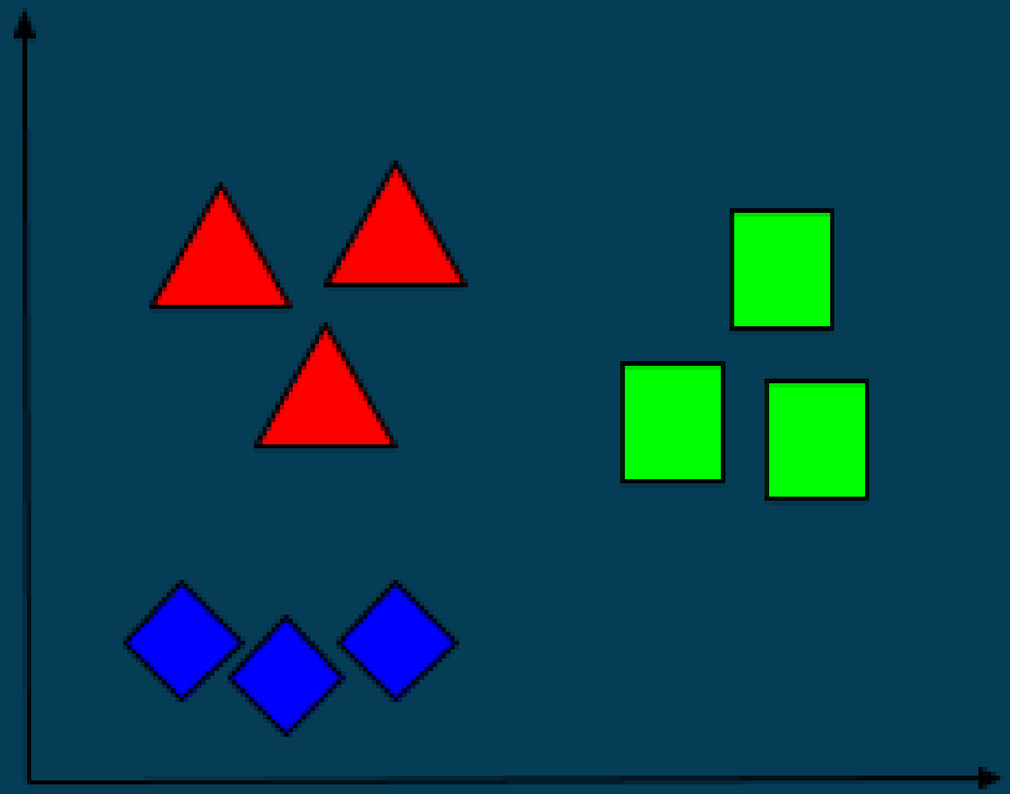
INVERSE DOCUMENT FREQUENCY

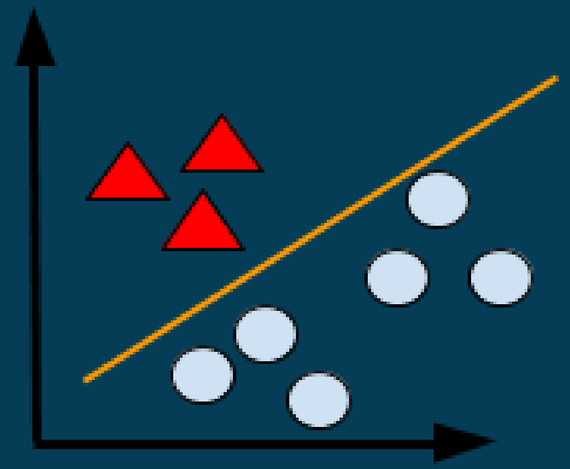
If a word appears in many documents, then it's not a unique identifier. Give the word a low score.

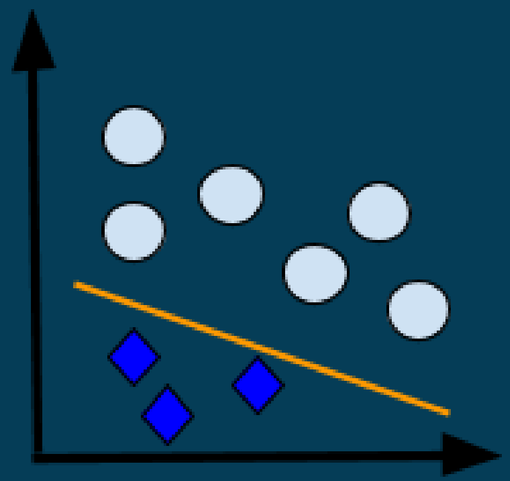
TRAINING THE MODELS

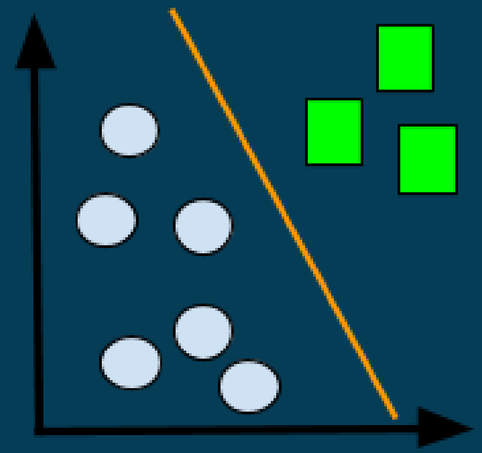
Support Vector Machine, Logistic Regression and Multinomial Naive Bayes

One-vs-Rest (One-vs-All) Approach



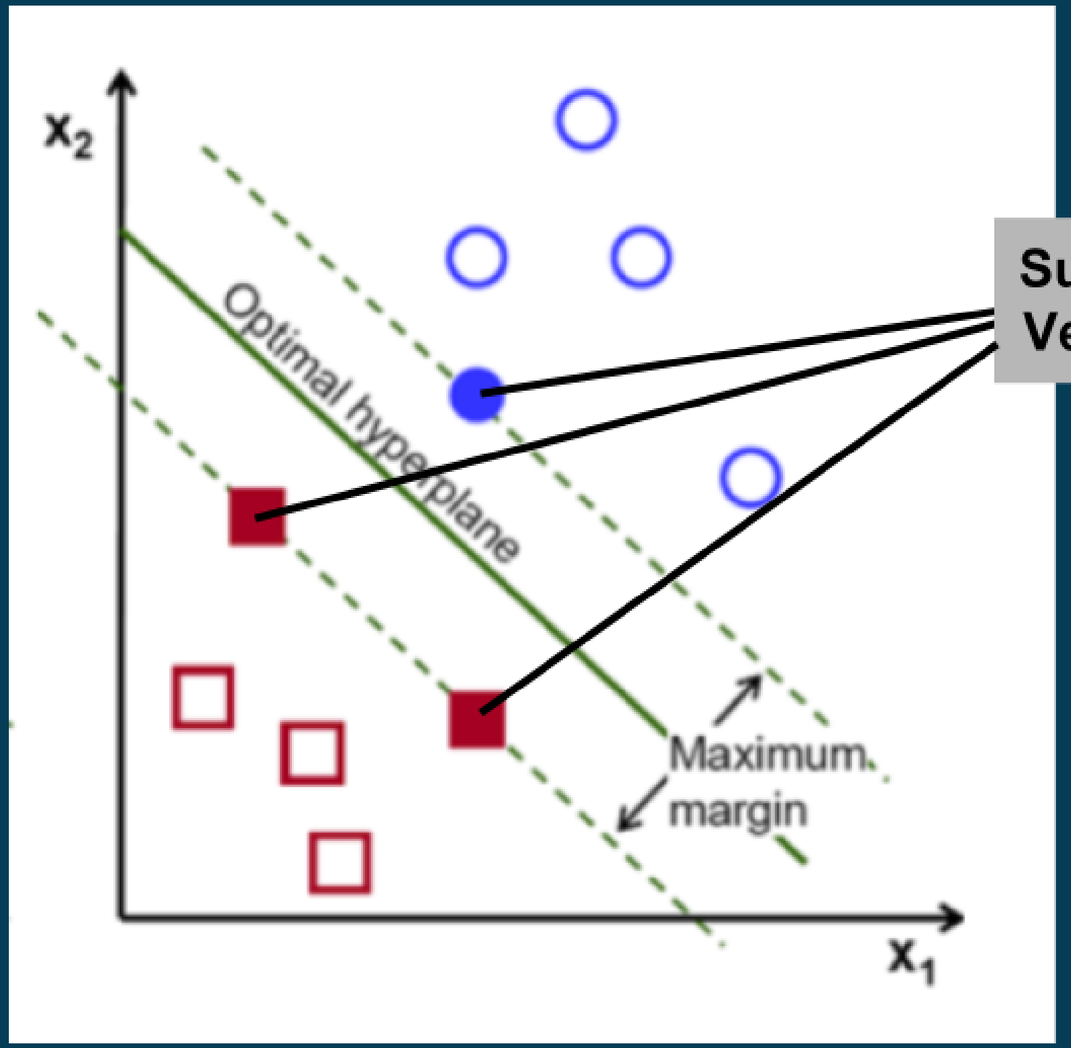




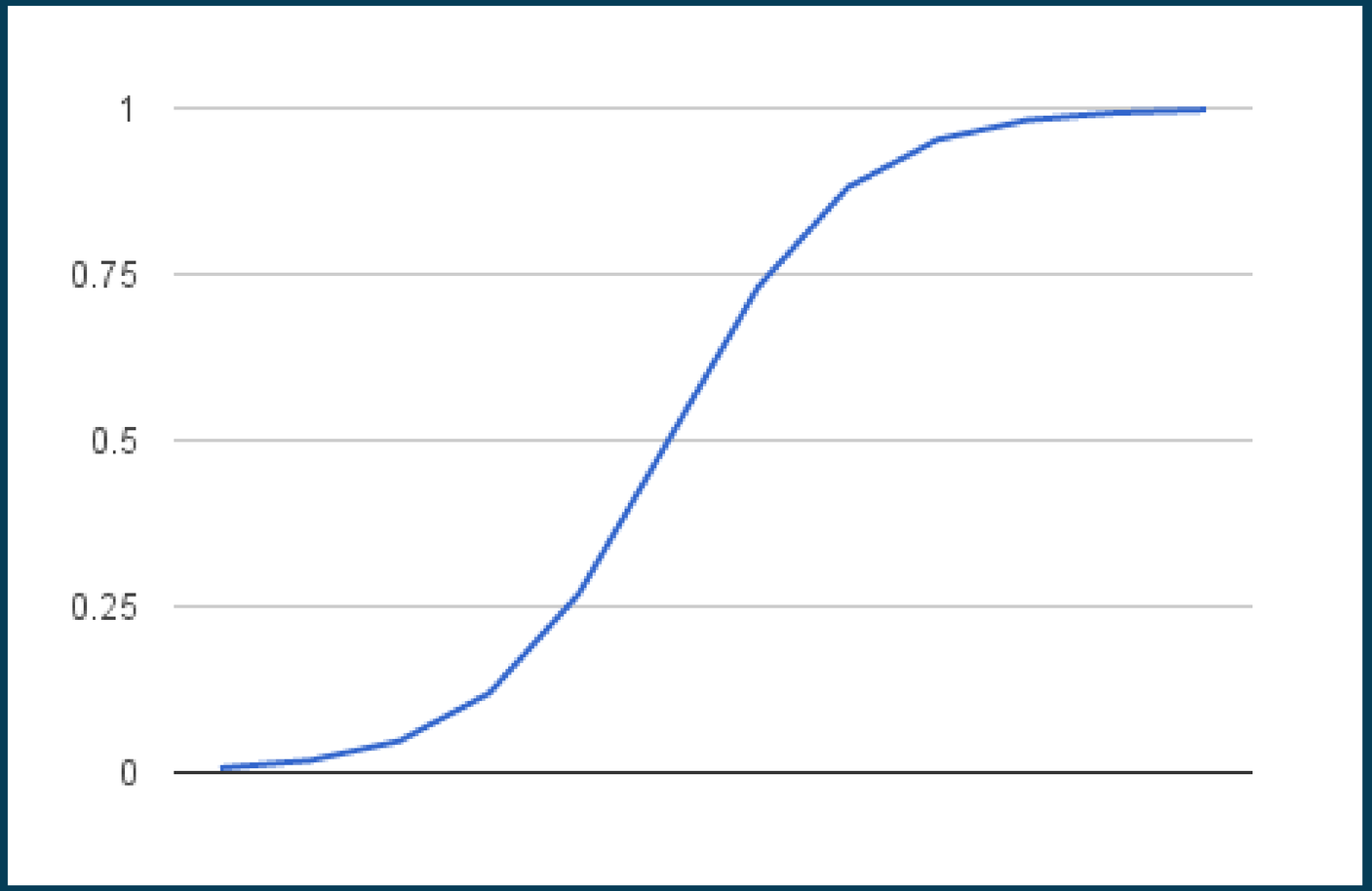


Support Vector Machine

- Based on decision planes
- SVM maximizes the margin
- The support vectors far from the decision hyperplane provides high level of certainty.



Logistic Regression

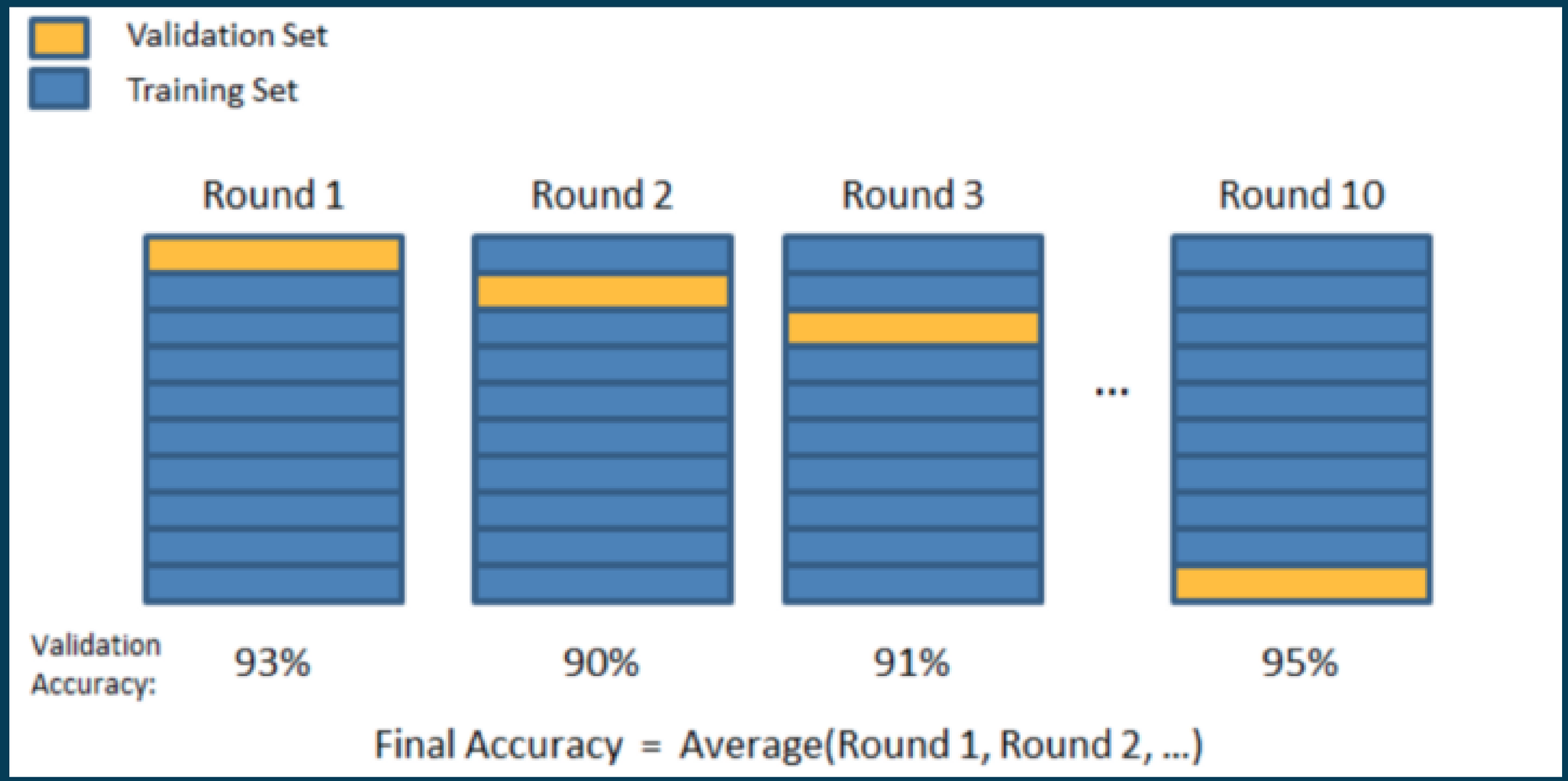


- Forms a logistic curve that limits the values from 0 to 1
- The curve is formed using the natural logarithm of the odds of the target labels

NAIVE BAYES

Naive Bayes algorithm is a classification technique that uses the probability theory called Bayes' theorem.

$$P(c|d) = \frac{P(d|c) * P(c)}{P(d)}$$



K-Fold Cross Validation

Split data into k folds where k = 10

Hyperparameter Tuning

Using Grid Search Algorithm

HYPERPARAMETER C RANGE OF VALUES: (FOR SVM AND LR)

[1,2,3,4,5]

HYPERPARAMETER TOL RANGE OF VALUES: (FOR SVM AND LR)

[1, 0.01, 0.001, 0.0001, 0.00000001]

ALPHA RANGE OF VALUES (FOR MNB)

[1, 0.01, 0.001]

Performance Evaluation

ACCURACY

Set of labels of a sample that exactly match the true corresponding set of labels

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

PRECISION

The ratio of correctly classified samples to the total classified positive samples

$$Precision = \frac{TP}{TP + FP}$$

Performance Evaluation

RECALL

The ratio of correctly classified samples to the total samples in the actual class

$$Recall = \frac{TP}{TP + FN}$$

F1 SCORE

The balance of recall and precision

$$Fscore = \frac{2 * Precision * Recall}{Precision + Recall}$$

Flask

Developing the chrome extension app

STATEMENT 1
I would like to use this system frequently when asking questions to SO

STATEMENT 2
The system is unnecessarily complex.

STATEMENT 3
The system was easy to use.

STATEMENT 4
I would need a technical person to use the system.

STATEMENT 5
The various functions in this system were well integrated.

STATEMENT 6
There were too much inconsistency in this system.

STATEMENT 7
Most people would learn to use this system very quickly.

STATEMENT 8
I think the system is very cumbersome to use.

STATEMENT 9
I felt very confident using the system.

STATEMENT 9
I needed to learn a lot of things before I could use this system.

SYSTEM USABILITY SCALE

RESULTS AND DISCUSSION

Training Evaluation (SVM)

Table I. Tuned Parameters of SVM for Each Label

Labels	C	tol	Training F1 Score
Python	1	0.01	85.72980
JS	2	0.01	83.28237
Java	1	1	89.63699
C	1	1	90.83980
R	2	0.01	89.38115
MySQL	1	0.01	89.26343
HTML	1	0.01	81.06965
If	1	0.01	75.93209
While	1	1	72.83676
For	1	1	71.83868
CSS	1	0.001	85.31148

C influences the length of the margin

tol (tolerance) or stopping criteria

Training Evaluation (LR)

Table II. Tuned Parameters of LR for Each Label

Labels	C	tol	Training F1 Score
Python	4	0.001	84.01987
JS	4	1	82.77860
Java	4	1	89.28364
C	4	1	90.79341
R	4	0.01	88.75614
MySQL	3	1	88.46617
HTML	3	0.001	80.32727
If	4	1	75.52289
While	3	1	71.50742
For	4	1	70.12207
CSS	4	0.001	84.26368

C is the regularization parameter to avoid overfitting

tol (tolerance) or stopping criteria

Training Evaluation (MNB)

Table III. Tuned Parameters of MNB for Each Label

Labels	Alpha	Training F1 Score
Python	0.1	83.59637
JS	0.01	75.92564
Java	0.1	88.00554
C	0.1	90.24161
R	0.01	88.55784
MySQL	0.1	87.92793
HTML	0.1	81.47754
If	0.01	69.01341
While	0.1	72.25000
For	0.1	67.90615
CSS	0.01	84.63210

Alpha is the laplace smoothing to catch zero frequency probability.

Testing Evaluation (Accuracy)

SVM

LR

MNB

Table IV. Accuracy of SVM, LR, and MNB for Each Fold

Folds	SVM	LR	MNB
Fold1	93.25153	93.11210	91.05782
Fold2	92.54508	90.85332	89.46830
Fold3	93.72560	91.41104	89.89589
Fold4	93.47462	92.72170	90.91839
Fold5	95.89143	95.64045	94.72950
Fold6	95.34300	95.64045	94.72950
Fold7	94.42276	93.35378	92.04313
Fold8	92.65842	92.00707	91.16032
Fold9	93.97972	93.27254	92.76077
Fold10	92.93756	92.22108	91.14171
Average	93.82297	92.96386	91.73476

SVM

HAS THE HIGHEST PERCENTAGE (93.8%) OF GETTING CORRECT PREDICTIONS

Accuracy

Testing Evaluation (Precision)

SVM

LR

MNB

Table V. Precision of SVM, LR, and MNB for Each Fold

Folds	SVM	LR	MNB
Fold1	88.40211	88.11907	82.39526
Fold2	86.41691	83.22667	78.38197
Fold3	86.97669	83.27971	77.65770
Fold4	85.66157	84.02380	78.76094
Fold5	81.00311	79.75806	77.91692
Fold6	79.44429	79.20743	77.04935
Fold7	89.53365	86.53630	82.57825
Fold8	78.58053	77.07480	75.87907
Fold9	73.94950	70.99401	72.46094
Fold10	87.88426	86.85786	83.61169
Average	83.78526	81.90777	78.66921

SVM

CORRECT 83.8%
OF THE TIME.

Precision

Testing Evaluation (Recall)

SVM

MNB

LR

Table VI. Recall of SVM, LR, and MNB for Each Fold

Folds	SVM	LR	MNB
Fold1	82.96112	81.49346	79.92735
Fold2	81.94898	78.93873	80.52252
Fold3	84.50890	80.10360	82.30886
Fold4	84.10196	80.65574	80.37382
Fold5	82.82266	79.85886	82.41797
Fold6	81.39430	80.44719	81.24740
Fold7	86.19803	81.34462	82.06336
Fold8	80.88833	78.14957	79.53993
Fold9	85.02173	81.08630	84.70296
Fold10	82.49878	79.19586	79.70251
Average	83.23448	80.12739	81.28067

SVM

CORRECTLY IDENTIFIES 83.2% OUT OF ITS ACTUAL POSITIVE LABELS FROM THE DATA

Recall

Testing Evaluation (F1 Score)

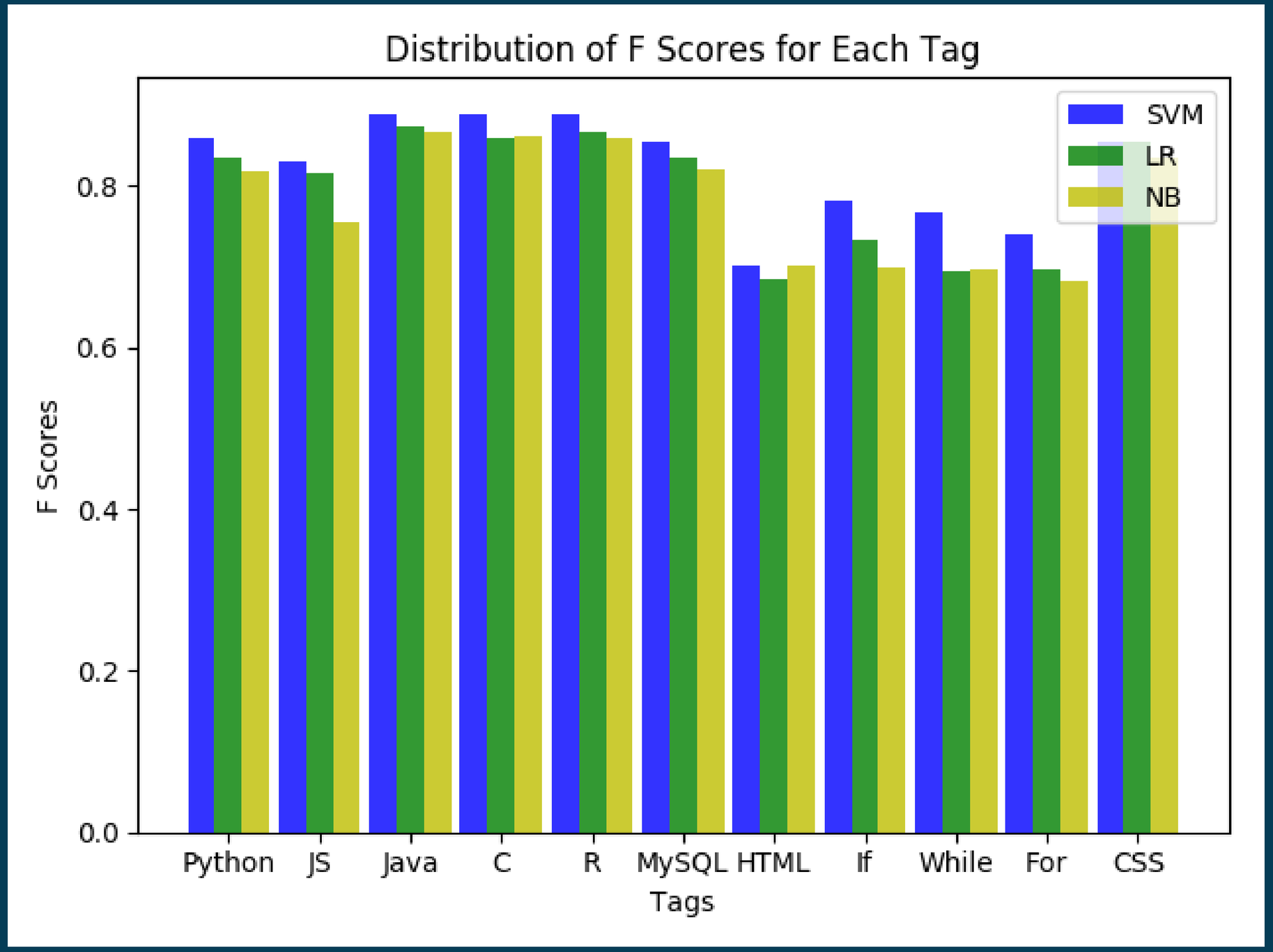
SVM

LR

MNB

Table VII. F1 score of SVM, LR, and MNB for Each Fold

Folds	SVM	LR	MNB
Fold1	85.13426	83.95301	80.71291
Fold2	83.32870	79.83060	78.31452
Fold3	84.96781	80.67052	78.10313
Fold4	83.81702	81.26988	78.03223
Fold5	80.47635	78.97241	77.33961
Fold6	78.41130	77.20139	76.00511
Fold7	87.49552	83.34496	81.72504
Fold8	78.70157	76.29201	76.09234
Fold9	75.62881	71.78486	74.10138
Fold10	84.61435	82.00414	80.93097
Average	82.25757	79.53238	78.13572


Distribution of F Scores for Each Tag
SVM
LR
NB
F Scores
0.0
0.2
0.4
0.6
0.8
Python
JS
Java
C
R
MySQL
HTML
If
While
For
CSS
Tags

VISUALIZATION OF THE APP

CHROME EXTENSION
APPLICATION

Title

Responsive width and height for a group of squares

Links Images Styling/Headers Lists Blockquotes Code HTML advanced help »

```
I'm creating a grid of 1:1 squares. The user can keep adding squares
and I want the size of the squares to be maintained at their aspect
ratio but resize accordingly. The tricky part is I want the square to
always be visible on the page - that is to say that there is no
scrolling and the webpage would be responsive with width and height.

I have created an example that adds a square every second while
testing this. However, I am unable to get it working with the height
part. I have been able to get it working with the width.

    <div class="square-container">
```

I'm creating a grid of 1:1 squares. The user can keep adding squares and I want the size of the squares to be maintained at their aspect ratio but resize accordingly. The tricky part is I want the square to always be visible on the page - that is to say that there is no scrolling and the webpage would be responsive with width and height.

I have created an example that adds a square every second while testing this. However, I am unable to get it working with the height part. I have been able to get it working with the width.

```
<div class="square-container">
  <div class="square"></div>
</div>
setInterval(()=> {
  console.log(document.getElementsByClassName('square-container')[0]);
  document.getElementsByClassName('square-container')[0].innerHTML += ("<div cla
}, 1000);
```

```
    position: relative;
    flex-basis: calc(33.333% - 10px);
    margin: 5px;
    box-sizing: border-box;
    background: red;
    transition: background 1s;
  }

  .square::before {
    content: "";
    display: block;
    padding-top: 100%;
  }
```

I'm not using any ui libraries like bootstrap, just vanilla html, css and javascript.

Tags

e.g. (xcode asp.net css)

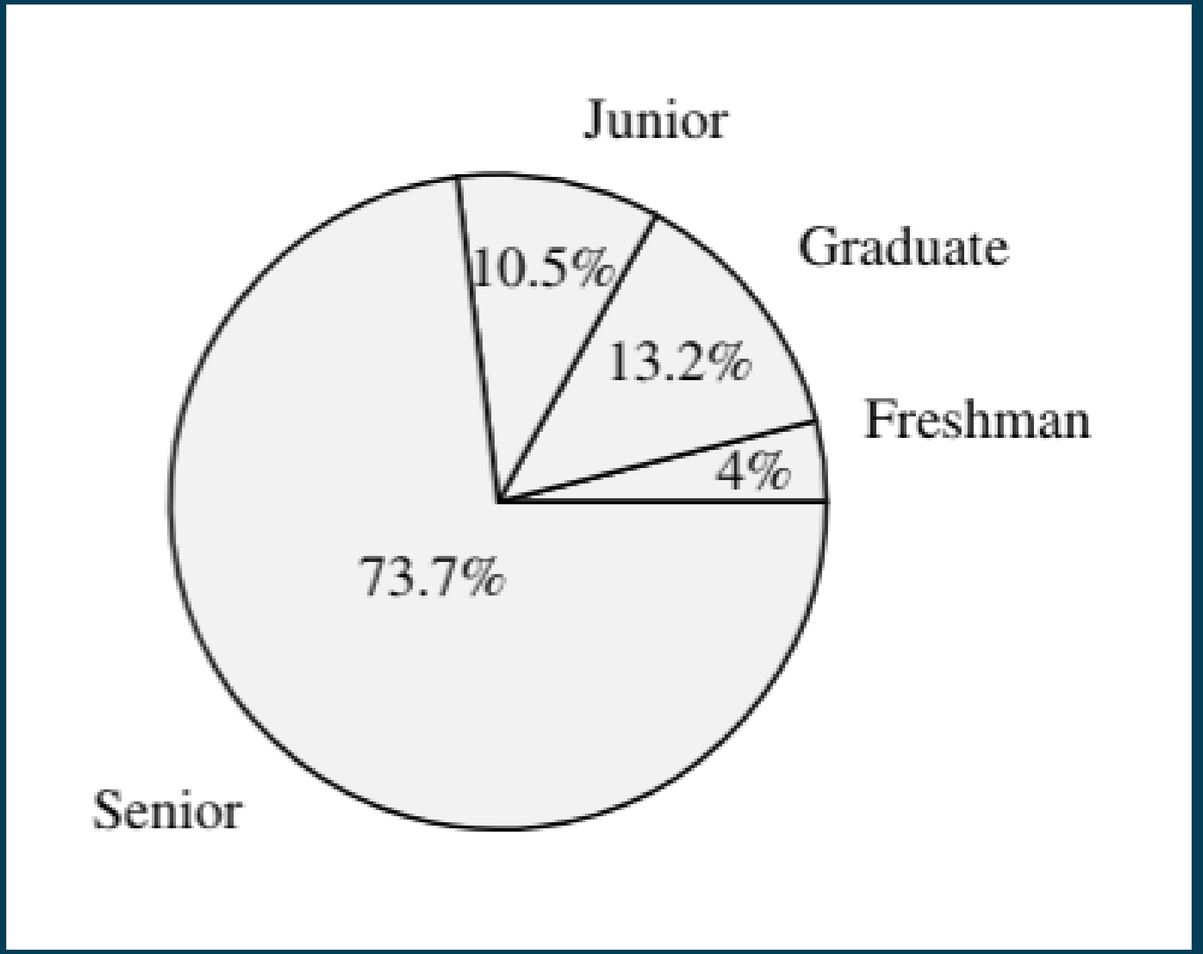
Tag Suggestions

javascript

html

css

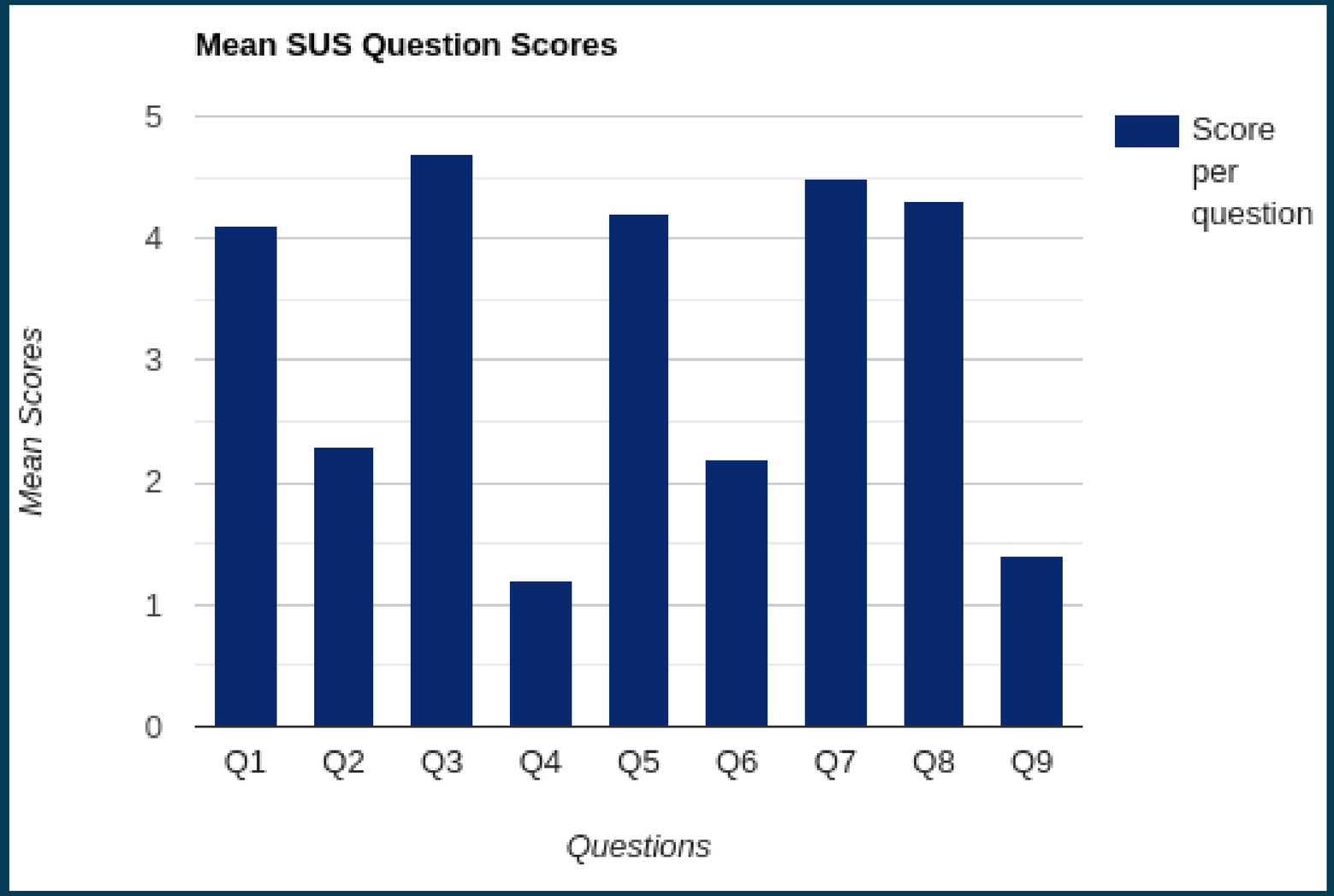
SYSTEM USABILITY SCALE



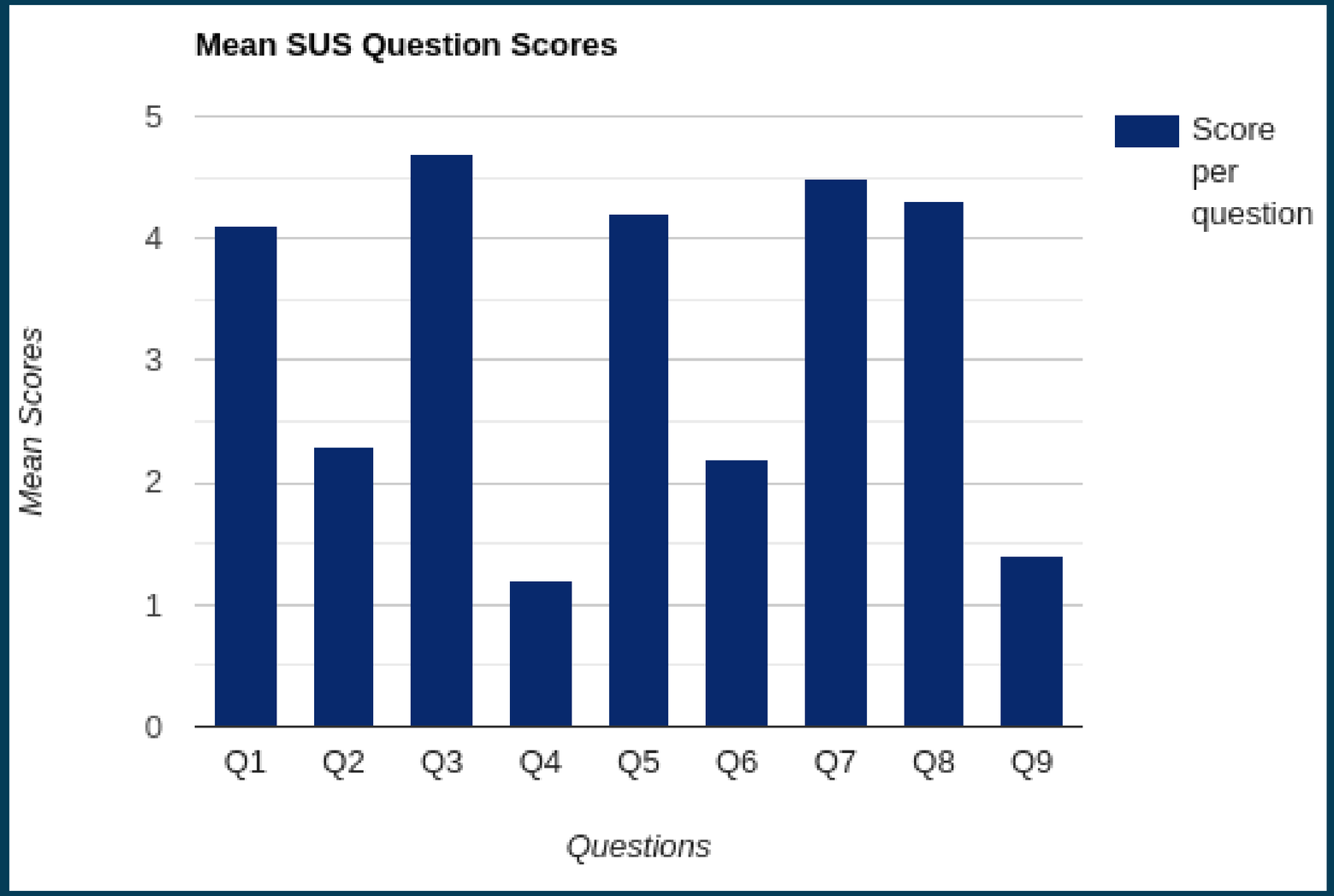
40 Evaluators

Mean SUS Scores

Q4 & 9 - Learnability
1 Easiest to learn, 5 Most Difficult to learn



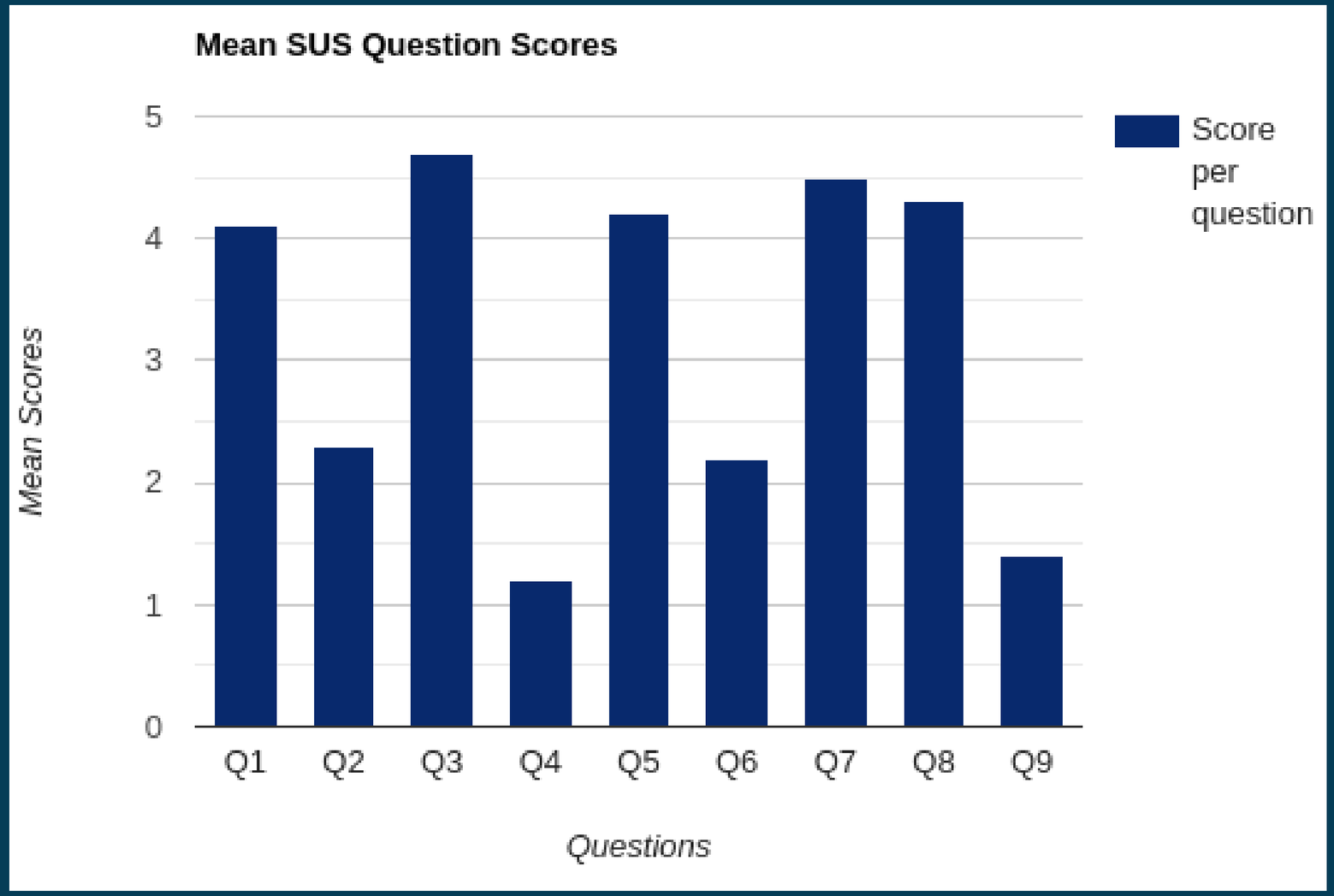
Mean SUS Scores



Q4 & 9 - Learnability
1 Easiest to learn, 5 Most Difficult to learn

Q2 & 6 - Complexity & Consistency
1 Least Complex, Least Inconsistent, 5 Most Complex and Most Inconsistent

Mean SUS Scores



Q4 & 9 - Learnability
1 Easiest to learn, 5 Most Difficult to learn

Q2 & 6 - Complexity & Consistency
1 Least Complex, Least Inconsistent, 5 Most Complex and Most Inconsistent

Q1,3,5,7,8- Usability
1 Least Usable, 5 Most Usable

OVERALL TAGSUGGEST PERFORMANCE

SCORE OF

74.36

ABOVE AVERAGE!