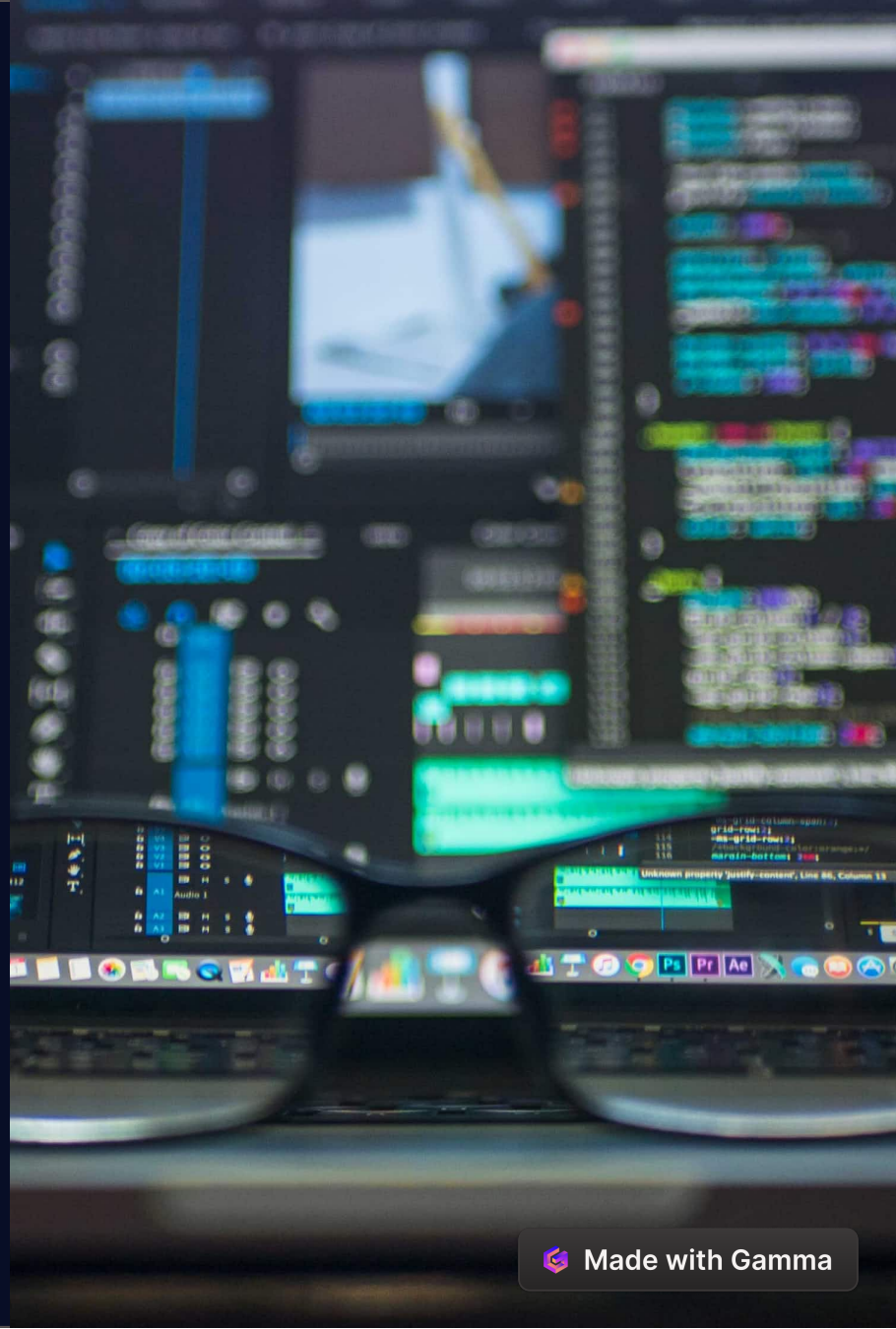
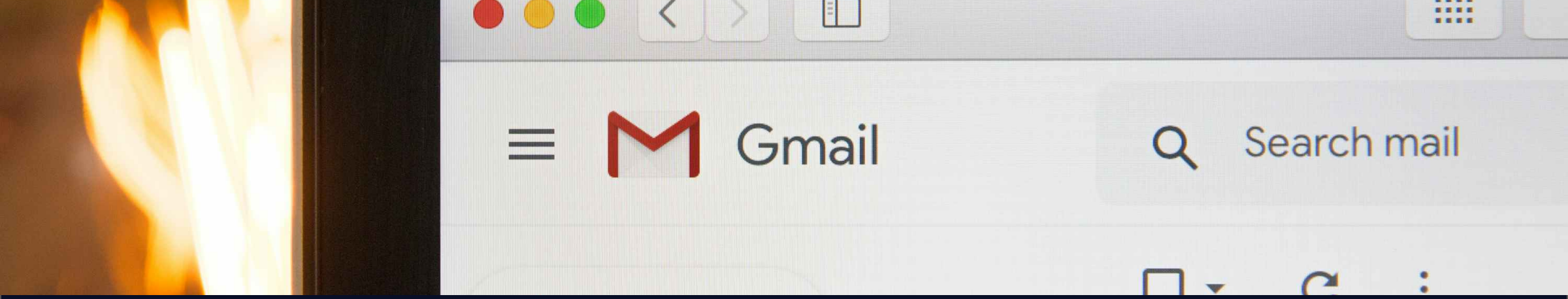
SMS Spam Filter Classifier

Building an Efficient Text Classification Model

by Kacem Benbrahim

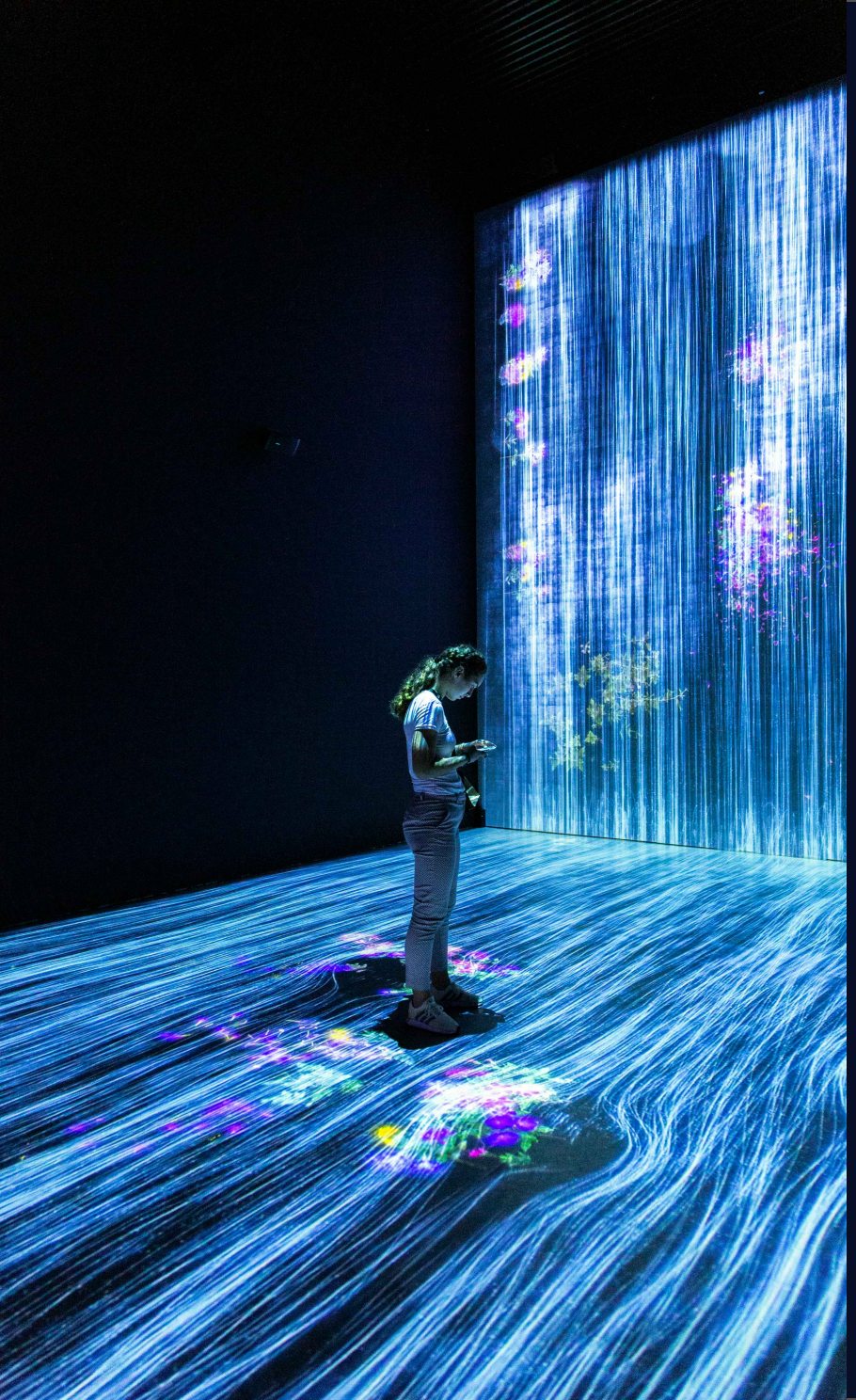




Introduction

Overview:

- The project focuses on creating a classifier to automatically distinguish between spam and legitimate (ham) messages



1-Project Components

1. Data Loading and Preprocessing:

- Utilized `pandas` to load the SMS dataset

```
df = pd.read_csv('data/sms-spam-collection-dataset/spam.csv', encoding="latin-1")
df.head()
```

	v1	v2	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	ham	Go until jurong point, crazy.. Available only ...	NaN	NaN	NaN
1	ham	Ok lar... Joking wif u oni...	NaN	NaN	NaN
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...	NaN	NaN	NaN
3	ham	U dun say so early hor... U c already then say...	NaN	NaN	NaN
4	ham	Nah I don't think he goes to usf, he lives aro...	NaN	NaN	NaN

- Removed unnecessary columns: `df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)`

remove column 2, 3 and 4 as they have no useful information

```
df.drop(['Unnamed: 2', 'Unnamed: 3', 'Unnamed: 4'], axis=1, inplace=True)
df['SMS'] = df['v2']
df['label'] = df['v1'].map({'ham': 0, 'spam': 1})
df.drop(['v1', 'v2'], axis=1, inplace=True)
train_data = df[:4400]
test_data = df[4400:]
df.head()
```

	SMS	label
0	Go until jurong point, crazy.. Available only ...	0
1	Ok lar... Joking wif u oni...	0
2	Free entry in 2 a wkly comp to win FA Cup fina...	1
3	U dun say so early hor... U c already then say...	0
4	Nah I don't think he goes to usf, he lives aro...	0

2-Model Selection

a-Machine Learning Models:

- Imported classifiers from scikit-learn: `from sklearn.naive_bayes import *`, `from sklearn.ensemble import *`.

```
from sklearn.naive_bayes import *
from sklearn.ensemble import *
from sklearn.neighbors import *
from sklearn.svm import *
```

list of various classifiers we are going to use

```
classifiers = [
        BernoulliNB(),
        RandomForestClassifier(n_estimators=100, n_jobs=-1),
        AdaBoostClassifier(),
        BaggingClassifier(),
        ExtraTreesClassifier(),
        GradientBoostingClassifier(),
        DecisionTreeClassifier(),
        CalibratedClassifierCV(),
        DummyClassifier(),
        PassiveAggressiveClassifier(),
        RidgeClassifier(),
        RidgeClassifierCV(),
        SGDClassifier(),
        OneVsRestClassifier(SVC(kernel='linear')),
        OneVsRestClassifier(LogisticRegression()),
        KNeighborsClassifier()
    ]
```

list of various vectorizers we are going to use

```
vectorizers = [
        CountVectorizer(),
        TfidfVectorizer(),
        HashingVectorizer()
    ]
```

3-Model Training

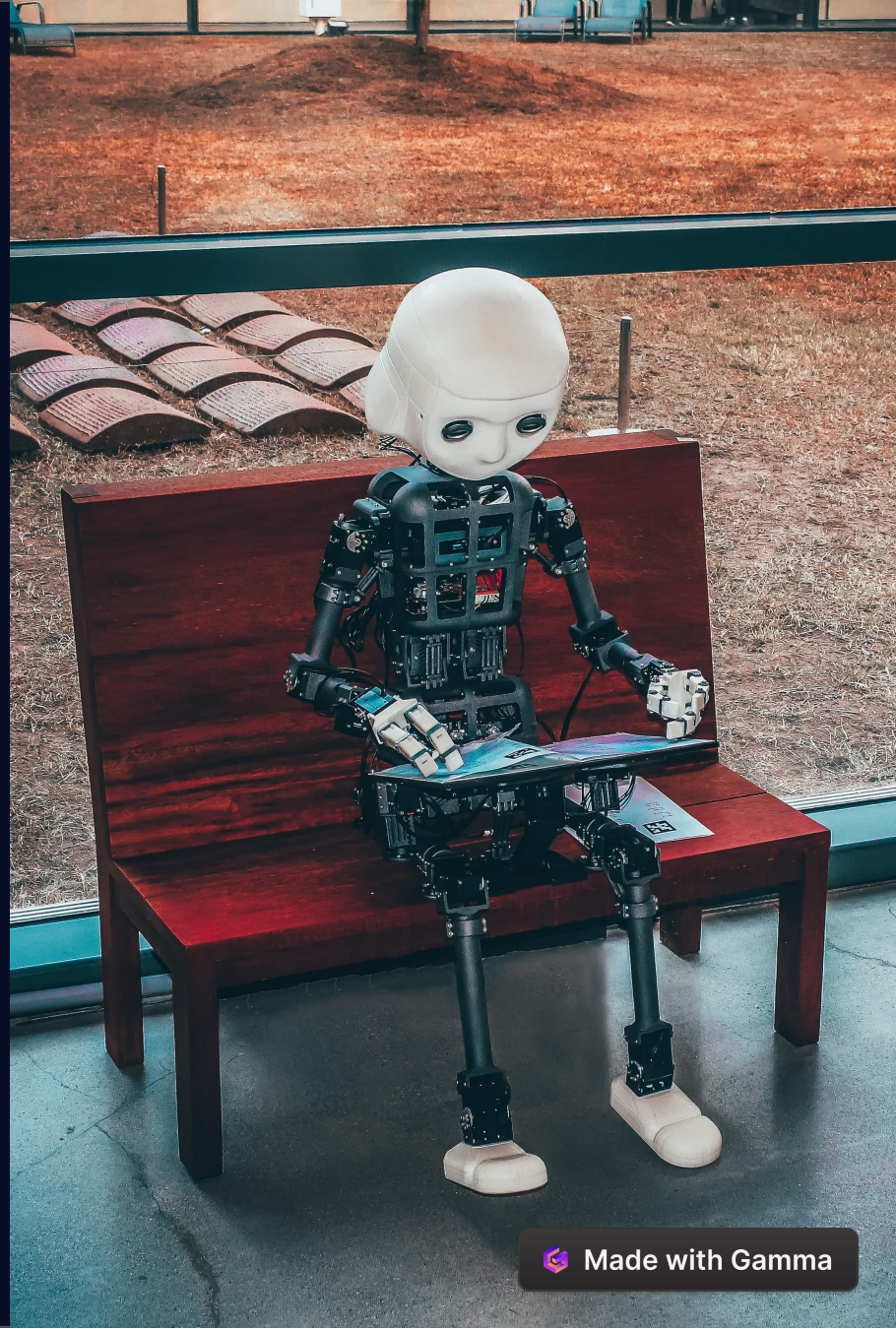
a-Training the Classifier:

- Split data into training and test sets.

```
train_data = df[:4400]
test_data = df[4400:]
df.head()
```

b-Chose the best-performing classifier and trained it.

```
# train the classifier with best accuracy
Classifier = OneVsRestClassifier(SVC(kernel='linear', probability=True))
Vectorizer = TfidfVectorizer()
vectorize_text = Vectorizer.fit_transform(train_data.SMS)
Classifier.fit(vectorize_text, train_data.label)
```



Model Evaluation

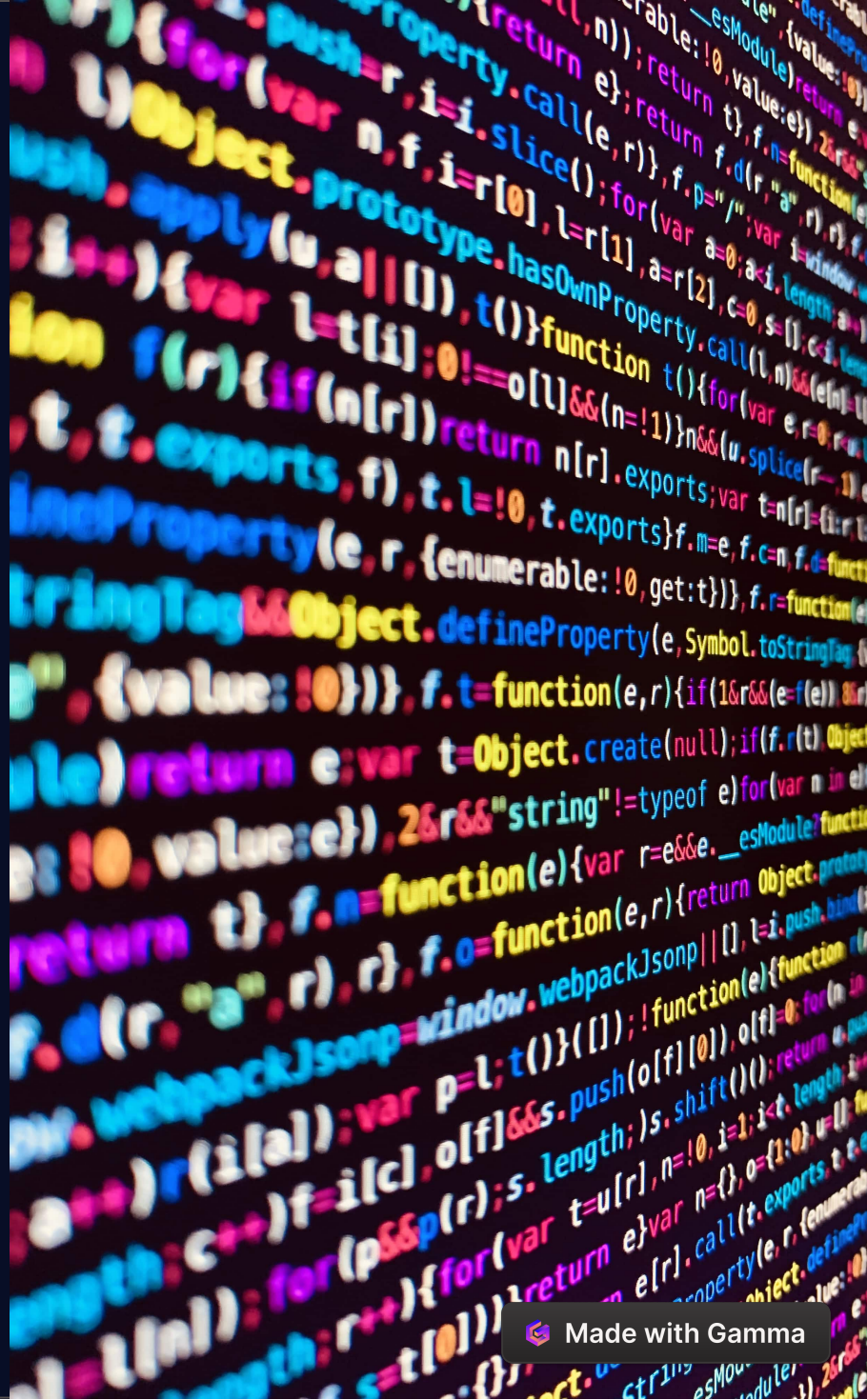
Performance Evaluation:

- Created a function (perform) to evaluate various combinations of classifiers and vectorizers.

```
def perform(classifiers, vectorizers, train_data, test_data):
    max_score = 0
    max_name = 0
    for classifier in classifiers:
        for vectorizer in vectorizers:

            # train
            vectorize_text = vectorizer.fit_transform(train_data.SMS)
            classifier.fit(vectorize_text, train_data.label)

            # score
            vectorize_text = vectorizer.transform(test_data.SMS)
            score = classifier.score(vectorize_text, test_data.label)
            name = classifier.__class__.__name__ + ' with ' + vectorizer.__class__.__name__
            print(name, score)
        if score > max_score:
            max_score = score
            max_name = name
    print ('============================================')
    print ('============================================')
    print (max_name, max_score)
    print ('============================================')
    print ('============================================')
```



Flask Web Application

Building the Web App:

- Created a Flask app with routes ('/' and '/predict').

```
from flask import Flask,render_template,url_for,request, jsonify
import pandas as pd
import pickle
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.multiclass import *
from sklearn.svm import *


app = Flask(__name__)
```

```
@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
```

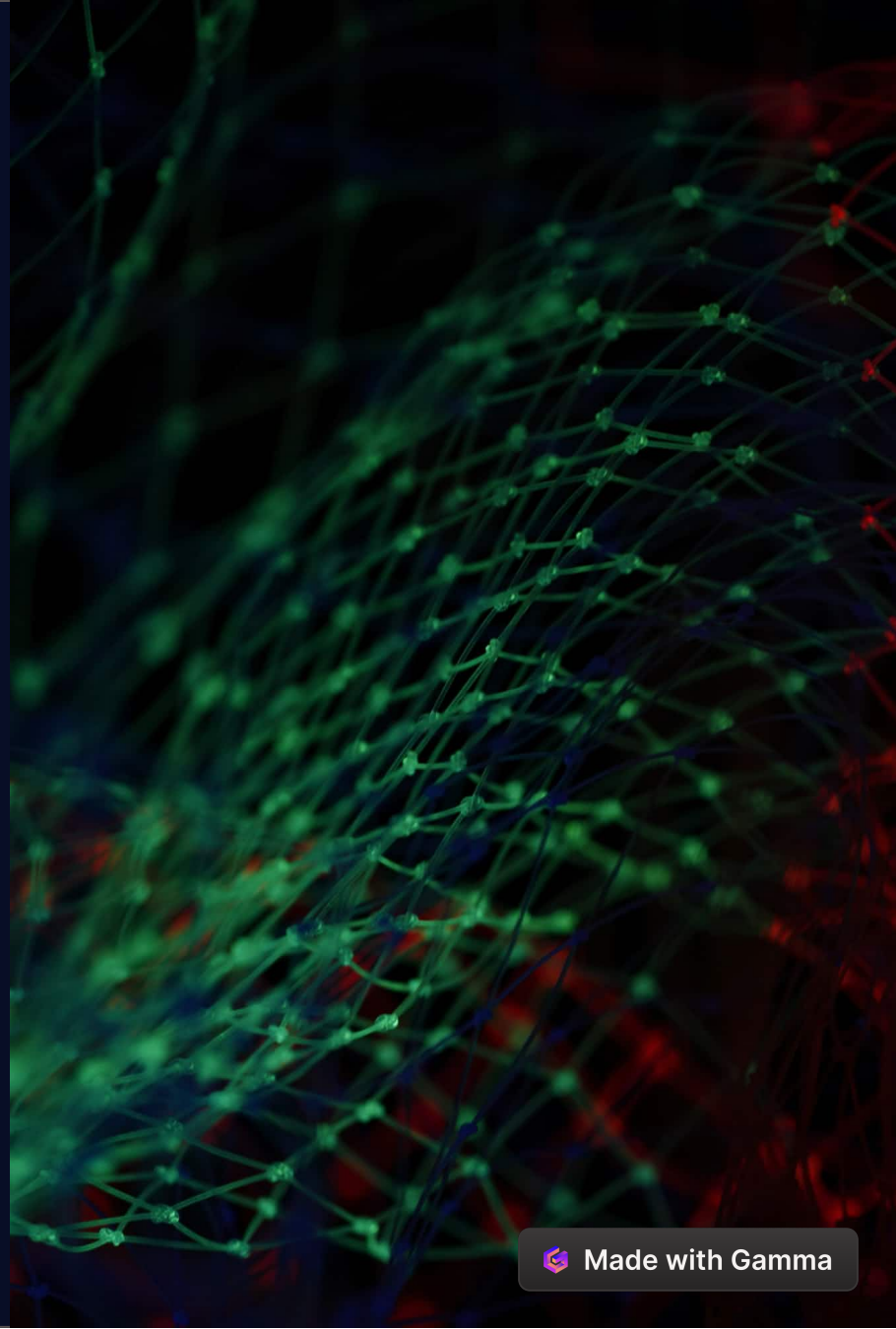
Results

Prediction Results:

```
[21]
SMS = ' won a 1 week FREE membership in our $100,000 Prize Jackpot! Txt the word: C'
vectorize_message = Vectorizer.transform([SMS])
predict = Classifier.predict(vectorize_message)[0]
```

```
[22]
if predict == 0:
    print ('ham')
else:
    print ('spam')
```

```
spam
```



App Interface

SMS Spam Classifier

Enter your SMS here

Predict

Examples	SMS
SPAM	URGENT! You have won a 1 week FREE membership in our $100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
HAM	Pls go ahead with watts. I just wanted to be sure. Do have a great weekend. Abiola
SPAM	Thanks for your subscription to Ringtone UK your mobile will be charged $5/month Please confirm by replying YES or NO. If you reply NO you will not be charged
HAM	U don't know how stubborn I am. I didn't even want to go to the hospital. I kept telling Mark I'm not a weak sucker. Hospitals are for weak suckers.

SMS Spam Classifier

Enter your SMS here

Predict

Ham

SMS Spam Classifier

BOX 4403LDNW1A7RW18

Predict

Spam

Thank You!

Any questions or feedback are highly appreciated