

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in Information Technology

By
EVANGELIN.E
REG. NO. 39120030



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

NOVEMBER 2021



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC
(Established under Section 3 of UGC Act, 1956)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI
CHENNAI- 600119
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **EVANGELIN.E** (Reg. No: 3912003) who carried out the project entitled "**Calls Classification using KNN algorithm**" under my supervision from June 2021 to November 2021.

Internal Guide

Dr. SENDURU SRINIVASALU, M.E., Ph.D.,

Head of the Department

Dr. SUBHASHINI, M.E., Ph.D

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **EVANGELIN.E** hereby declare that the project report entitled **Calls Classifcatio System using KNN Algorithm** done by me under the guidance of **Dr. Senduru Srinivaalu** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE:

E.EVANGELIN

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D, Dean**, School of Computing and **DR.SUBHASHINI,M.E.,Ph.D., Heads of the Department of Information technology** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. SENDURU SRINIVASALU, M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE

CERTIFICATE OF COMPLETION



THIS IS TO ACKNOWLEDGE THAT

EVANGELIN.E

Is Certified in

MACHINE LEARNING USING PYTHON (B Code: 210921-ML-2)

And has successfully completed all requirements and criteria for said certification through examination administered by irms

This Certification Earned on

October 2021


Controller


Authorised Signature

TABLE OF CONTENTS

Chapter No.	TITLE	PAGE NO.
	ABSTRACT	I
	LIST OF FIGURES	II
	LIST OF ABBREVIATION	III
1.	INTRODUCTION 1.1.WHAT IS MACHINE LEARNING? 1.2.TYPES OF MACHINE LEARNING 1.3.KNN-ALGORITHM 1.4.REQUIREMENTS FOR ML MODEL	1 , 2
2.	CALLS CLASSIFICATION USING KNN ALGORITHM 2.1.OBJECTIVE 2.2.MODEL IMPLEMENTATION 2.3.METHODOLOGY	3 to 6
3.	RESULT AND CONCLUSION 3.1.PREDICTION AND RESULT 3.2.CONCLUSION	7
	REFERENCE	8
	APPENDIX SOURCE CODE	8 to 11

ABSTRACT

As the technology developed to its peak, security issues were also increasing with it . Nowadays almost everyone are using mobile phones to communicate with others and through these mobile phone we face many dangers ,one of these danger is spam calls .These spam callers are involving in fraudalants .To aviod these kind of spam calls ,I designed a supervised machine learning model using KNN algorithm to detect spam calls with there average calls per day and the total reports filed on the person .First we split the dataset into training and testing part,after this we analyze data and we implement the KNN algorithm to classify the calls , after implementing we predict the confusion matrix ,configuration report and accuracy rate .The designed model predicts the call using KNN algorithm and gives 100% accuracy rate and f1_score.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME
3.1	LOADING DATASET
3.2	DATA ANALYSIS
3.3	SPLITTING TRAIN AND TEST DATA
3.4	IMPLEMENTING KNN ALGORITHM AND PREDICTION
3.5	EVALUATE THE RESULT
3.6	CALCULATE THE ERROR RATE
3.7	PLOT THE ERROR RATE

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
KNN	K Nearest Neighbor
ML	Machine Learning
Sup.ML	Supervised machine learning
Acc.	According

1.INTRODUCTION

1.1. WHAT IS MACHINE LEARNING?

Arthur Samuel, a pioneer in the field of artificial intelligence and computer gaming, coined the term “**Machine Learning**”. He defined machine learning as – “**Field of study that gives computers the capability to learn without being explicitly programmed**”.

In a very layman manner, Machine Learning(ML) can be explained as automating and improving the learning process of computers based on their experiences without being actually programmed i.e. without any human assistance. The process starts with feeding good quality data and then training our machines(computers) by building machine learning models using the data and different algorithms. The choice of algorithms depends on what type of data do we have and what kind of task we are trying to automate.

1.2. TYPES OF MACHINE LEARNING

There are many ways to frame this idea, but largely there are three major recognized categories: supervised learning, unsupervised learning, and reinforcement learning.

Supervised machine learning :

Supervised learning is the most popular paradigm for machine learning. It is the easiest to understand and the simplest to implement. It is very similar to teaching a child with the use of flash cards.

Supervised learning is often described as task-oriented because of this. It is highly focused on a singular task, feeding more and more examples to the algorithm until it can accurately perform on that task.

Unsupervised machine learning :

Unsupervised learning is very much the opposite of supervised learning. It features no labels. Instead, our algorithm would be fed a lot of data and given the tools to understand the properties of the data. From there, it can learn to group, cluster, and/or organize the data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data.

Reinforcement machine learning :

Reinforcement learning is fairly different when compared to supervised and unsupervised learning. Where we can easily see the relationship between supervised and unsupervised (the presence or absence of labels), the relationship to reinforcement learning is a bit murkier. Some people try to tie reinforcement learning closer to the two by describing it as a type of learning that relies on a time-dependent sequence of labels, however, my opinion is that that simply makes things more confusing.

1.3. KNN-ALGORITHM :

This algorithm is an supervised machine learning algorithm and is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line.

Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to overfit the data and resulting in complex models.

1.4. Requirements for ML project :

- Install Anaconda3 .
- choose the right dataset.
- launch jupyter notebook and create the workspace.
- Import the required libraries.
- And initialize the classifier and implement KNN algorithm.
- Run the model and predict the results and evaluate it.

2.CALLS CLASSIFICATION SYSTEM USING KNN-ALGORITHM :

2.1.OBJECTIVE :

This project aims to predict the calls as SPAM or NOT A SPAM using KNN algorithm to avoid fraud calls .Our model will predict the result by splitting the dataset into train and test part .After splitting ,the model learns from the train part and predicts the result for the test part .

2.2. MODEL IMPLEMENTATION :

- Install Anaconda3.
- Launch Jupyter Notebook using Anaconda Navigator.
- Open the workspace and import the libraries such as pandas,numpy,seaborn,sklearn ect.
- Load the dataset using CSV reader.
- Analyze the data and implement the algorithm.
- Predict the Results and evaluate the predictions.

2.3.METHODOLOGY :

The first step is to launch the jupyter notebook using anaconda navigator and open a new work space .After creating a new workspace,firstly import the libraries and moduls such as pandas, numpy, seaborn ,sklearn ,matplotlib ect.

After importing the libraries,load the dataset (spreadsheet file saved as csv file) using csv reader .

```
In [2]: #Loading the dataset and printing it for a Look

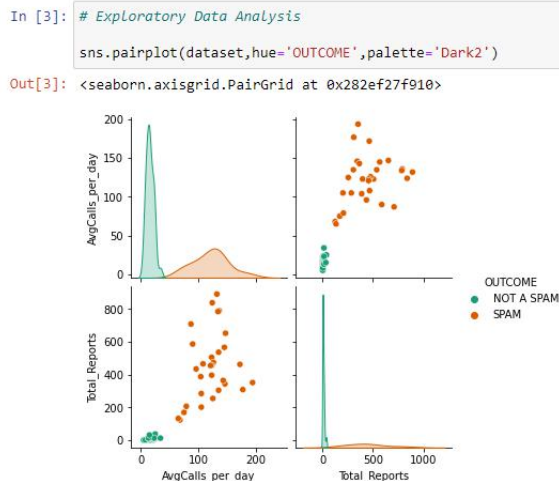
dataset = pd.read_csv('C:\\Users\\evang.LAPTOP-1BPRT0TT\\Downloads\\Calls Dataset - Sheet1.csv')

print( len(dataset))
print(dataset)

66
   AvgCalls_per_day  Total_Reports  OUTCOME
0                10              1  NOT A SPAM
1                 15              0  NOT A SPAM
2               105             202      SPAM
3                75             170      SPAM
4                 8              2  NOT A SPAM
..              ...             ...      ...
61              121             456      SPAM
62              134             785      SPAM
63               13              14  NOT A SPAM
64               15              32  NOT A SPAM
65               65             133      SPAM

[66 rows x 3 columns]
```

After loading the dataset , analyze the datas by plotting and using group by functions.



Now split the dataset into train and test part using train_test_split function and scale the train and test part to fit using StandardScaler().

```

In [18]: # printing the actual outcome values

print(dataset.groupby("OUTCOME").size())

OUTCOME
NOT A SPAM    36
SPAM          30
dtype: int64

In [4]: #Splitting the dataset for training and testing

X = dataset.iloc[:,[0,1]]
Y = dataset.iloc[:,-1]
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.25,random_state=0)

In [7]: #Feature Scaling

sc_X = StandardScaler()
X_train = sc_X.fit_transform(X_train)
X_test = sc_X.transform(X_test)

```

After splitting,use a variable classifier with implenting the Built KNN algorithm function with k value as 3. Now pedict the results using the predict function and assign the predicted results in the variable called Y_pred.

```

In [9]: #define the model: Init K-NN
# n_neighbors takes value from above calculation as 3

classifier = KNeighborsClassifier(n_neighbors=3, p=2, metric='euclidean')

In [10]: # Fit Model

classifier.fit(X_train,Y_train)

Out[10]: KNeighborsClassifier(metric='euclidean', n_neighbors=3)

In [11]: # Predict the test set results
# y_pred holds the predicted values

Y_pred = classifier.predict(X_test)

In [12]: #the predicted results are..

print(Y_pred)

['SPAM' 'NOT A SPAM' 'NOT A SPAM' 'SPAM' 'NOT A SPAM' 'NOT A SPAM' 'SPAM'
 'NOT A SPAM' 'SPAM' 'NOT A SPAM' 'NOT A SPAM' 'SPAM' 'SPAM' 'SPAM'
 'SPAM' 'SPAM']

```

Evaluating the predicted results using confusion matrix,classification report and accuracy rate

```

In [13]: # Evaluate the model using Confusion Matrix

cm = confusion_matrix(Y_test,Y_pred)
print(cm)

[[ 7  0]
 [ 0 10]]

In [14]: # Evaluate the model using classification report

print(classification_report(Y_test,Y_pred))

              precision    recall  f1-score   support

 NOT A SPAM      1.00      1.00      1.00         7
      SPAM       1.00      1.00      1.00        10

 accuracy              1.00              17
 macro avg           1.00           1.00           17
 weighted avg        1.00           1.00           17

In [15]: # Calculate the accuracy of this model

print(accuracy_score(Y_test,Y_pred))

1.0

```

Check the model's error rate using a set of code and plot the graph and evaluate the results.

The set of code :

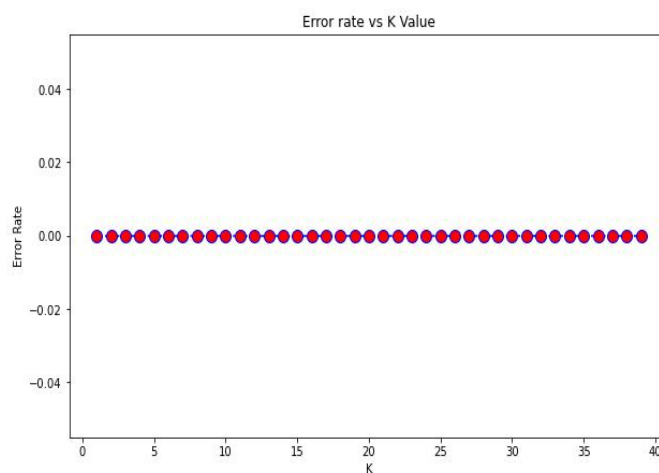
```
In [16]: error_rate = []

for i in range(1,40):
    classifier = KNeighborsClassifier(n_neighbors=3)
    classifier.fit(X_train,Y_train)
    pred_i = classifier.predict(X_test)
    error_rate.append(np.mean(pred_i != Y_test))

In [17]: plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue',linestyle='dashed',marker='o',markerfacecolor='red',markersize=10)
plt.title('Error rate vs K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

Plot the graph :

Out[17]: Text(0, 0.5, 'Error Rate')



Supervised Machine Learning model to classify calls using KNearestNeighbor algorithm is successfully predicted the results with 100% accuracy rate

3 .RESULT AND CONCLUSION :

3.1.PREDICTION AND RESULT

- The Calls Classification model learns and predict the results with 100% accuracy rate and 0% error rate .
- The confusion matrix gives 7 actual true and predicted true value and 0 actual false and predicted false with 100% f1_score .
- This supervised machine learning model produce 100% accurate prediction .

3.2.CONCLUSION

- Using the KNN algorithm the calls has been classified and made prediction with 100% accuracy rate.
- As mentioned in the objective this model classify the call whether it is a SPAM or NOT A SPAM.
- In the future work,My goal is to predict whether the call is an local or international call.

REFERENCE :

[1] Ruchit Modi and Saifuddin Hitawala,"Email-Classification Using Supervised Learning Algorithms Project Demo",Machine Learning,springer 2017.

APPENDIX :

SOURCE CODE :

```
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import f1_score
from sklearn.metrics import accuracy_score
from pandas.plotting import scatter_matrix
from sklearn.metrics import classification_report
import seaborn as sns
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib','inline')

#loading the dataset and printing it for a look
dataset=pd.read_csv('C:\\Users\\evang.LAPTOP1BPRT0TT\\Downloads\\Calls Dataset - Sheet1.csv')
```

```
print( len(dataset))
```

```
print(dataset)
```

```
# Exploratory Data Analysis
```

```
sns.pairplot(dataset,hue='OUTCOME',palette='Dark2')
```

```
# printing the actual outcome values
```

```
print(dataset.groupby("OUTCOME").size())
```

```
#Splitting the dataset for training and testing
```

```
X = dataset.iloc[:,[0,1]]
```

```
Y = dataset.iloc[:,-1]
```

```
X_train,X_test,Y_train,Y_test =
```

```
train_test_split(X,Y,test_size=0.25,random_state=0)
```

```
#Feature Scaling
```

```
sc_X = StandardScaler()
```

```
X_train = sc_X.fit_transform(X_train)
```

```
X_test = sc_X.transform(X_test)
```

```
#calculate the K value
```

```
import math
```

```
math.sqrt(len(Y_test))
```

```
#define the model: Init K-NN
# n_neighbors takes value from above calculation as 3

classifier =KNeighborsClassifier(n_neighbors=3,p=2,metric ='euclidean')

# Fit Model

classifier.fit(X_train,Y_train)

# Predict the test set results
# y_pred holds the predicted values

Y_pred = classifier.predict(X_test)

#the predicted results are..

print(Y_pred)

# Evaluate the model using Confusion Matrix

cm = confusion_matrix(Y_test,Y_pred)
print(cm)

# Evaluate the model using classification report

print(classification_report(Y_test,Y_pred))

# Calculate the accuracy of this model
```

```
print(accuracy_score(Y_test,Y_pred))

error_rate = []

for i in range(1,40):
    classifier = KNeighborsClassifier(n_neighbors=3)
    classifier.fit(X_train,Y_train)
    pred_i = classifier.predict(X_test)
    error_rate.append(np.mean(pred_i != Y_test))

plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue',linestyle='dashed',marker='o',
markerfacecolor='red',markersize=10)
plt.title('Error rate vs K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```