Support Vector Machine Kernel Functions Comparison

Bishoy Abd-ElMassieh Aiad, Karim Basem Zarif, and Zeyad Mahmoud Gadallah
Arab Academy for Science Technology and Maritime Transport, Aswan, b.a.aboalsaad@student.aast.edu,

karimbasem42@gmail.com, zeyadmahmoud00@gmail.com

Supervisor: College of Computing and Information Technology, Hadeel Abd EL-kareem, TA
Arab Academy for Technology, Information and Maritime Transport, Aswan, hadeelsaleh@aast.edu

Abstract – This paper conducts a comparative analysis between several kernel functions of support vector machine learning classifier for (Surveillance), to determine which kernel function is better in determining the best one that can be used on medical diagnosis datasets for the best accuracy and performance. Its goal is to find the best kernel for the best accuracy and performance for medical diagnosis datasets and doing it through comparing 3 different kernels which are: "Radial Basis Function", "Linear Function", "Polynomial Function". We used Support Vector Machine (SVM) algorithm in our training and testing while K folding (Cross-Validation) in our research to find the best accuracy. And we tested our kernel functions on a dataset called "lungcancer.csv" from Kaggle and experimented on the dataset achieving good results in performance measures: accuracy, precision, sensitivity (Recall) and specificity. The dataset briefly talks about categories of COVID-19 surveillance case like Person without Symptoms (PWS) refers to people who show no symptoms, Person in Monitoring (PIM) refers to people under observation (suspected person) and Patient under Supervision (PUS) refers to patients under surveillance.

Keywords — Support Vector Machine, Radial Basis Function, Linear Function, Polynomial Function, Cross-Validation, COVID-19 Surveillance case.

I. Introduction

The COVID-19 pandemic, also known as the coronavirus pandemic, is an ongoing global pandemic of coronavirus disease 2019 (COVID-19), which is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The virus was first identified in December 2019 in Wuhan, China. The World Health Organization declared a Public Health Emergency of International Concern on 30 January 2020, and later declared a pandemic on 11 March 2020. As of 3 July 2021, more than 183 million cases have been confirmed, with more than 3.96 million confirmed deaths attributed to COVID-19, making it one of the deadliest pandemics in history [1].

Data mining explores large data sets to extract critical decision-making data from past collections for future analysis. The medical field contains extensive patient information. This data needs to be mined by various machine learning algorithms [2]. Healthcare professionals analyse this data to achieve an effective diagnostic decision by health professionals. Mining medical data using classification algorithms provide clinical assistance with analysis. It examines the classification algorithms to predict coronavirus in patients.

II. Methodology

The methodology for predicting coronavirus was done using this algorithm (Support Vector Machine Classifier) and comparing its three major kernel functions with each other through their experimental results:

1. RBF (Radial Basis Function).
2. Linear Function.
3. Polynomial Function.

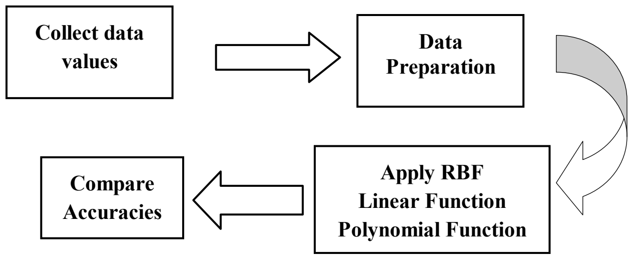
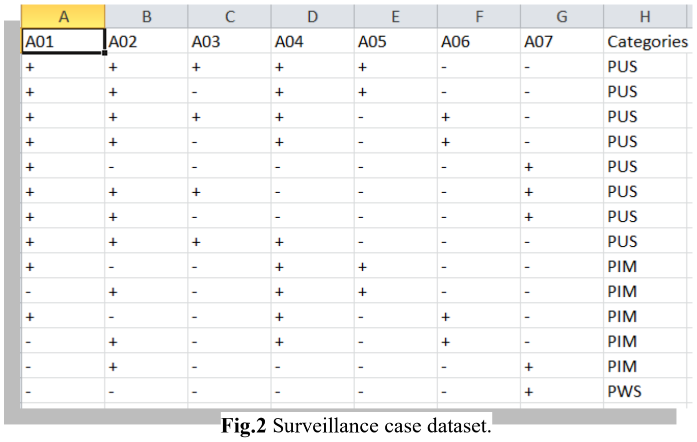


Fig.1 Methodology to predict corona virus

III. Dataset

There are 7 features that represent symptoms of COVID-19 (any version), and the categories represent the three we talked about such as PIM (Person in Monitoring), PUS (Patient under Supervision) and PWS (Person without Symptoms), In Surveillance case dataset "Fig 2" [6].



A	B	C	D	E	F	G	H
A01	A02	A03	A04	A05	A06	A07	Categories
+	+	+	+	+	-	-	PUS
+	+	-	+	+	-	-	PUS
+	+	+	+	-	+	-	PUS
+	+	-	+	-	+	-	PUS
+	-	-	-	-	-	+	PUS
+	+	+	-	-	-	+	PUS
+	+	-	-	-	-	+	PUS
+	+	+	+	-	-	-	PUS
+	-	-	+	+	-	-	PIM
-	+	-	+	+	-	-	PIM
+	-	-	+	-	+	-	PIM
-	+	-	+	-	+	-	PIM
-	+	-	-	-	-	+	PIM
-	-	-	-	-	-	+	PWS

Fig.2 Surveillance case dataset.

IV. Support Vector Machine

SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line.

In SVM, we plot each data item in the dataset in an N-dimensional space, where N is the number of features/attributes in the data. Next, find the optimal hyperplane to separate the data.

We use kernelized SVM for non-linearly separable data.

We can transform this data into two-dimensions and the data will become linearly separable in two dimensions as in the figure below "Fig 3".

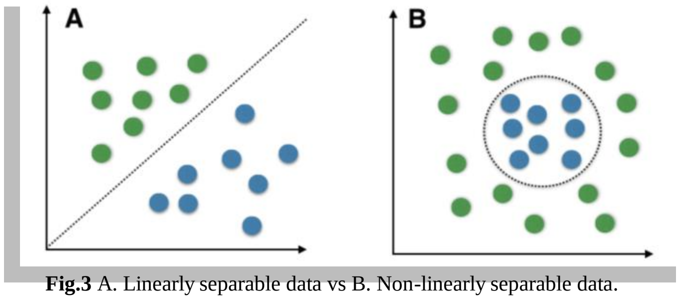


Fig.3 A. Linearly separable data vs B. Non-linearly separable data.

This is done by mapping each 1-D data point to a corresponding 2-D ordered pair.
There are various kernel functions available, but we choose the three most popular ones:

A. Radial Basis Function.
B. Linear Function.
C. Polynomial Function.

A. Radial Basis Function

Radial Basis Kernel is a kernel function that is used in machine learning to find a non-linear classifier or regression line.
KRBF (x, x`) = exp [$-\gamma$ ||x – x`||^2]

Where γ is a parameter that sets the "spread" of the kernel and *x, x'* are vector point in any fixed dimensional space.

The main motive of the kernel is to do calculations in any d-dimensional space where *d > 1*, so that we can get a quadratic, cubic or any polynomial equation of large degree for our classification/regression line. Since Radial basis kernel uses exponent and as we know the expansion of e^x gives a polynomial equation of infinite power, so using this kernel, we make our regression/classification line infinitely powerful too [3].

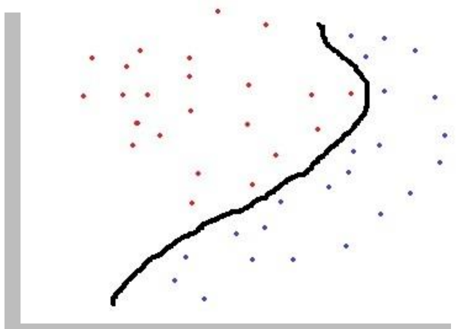
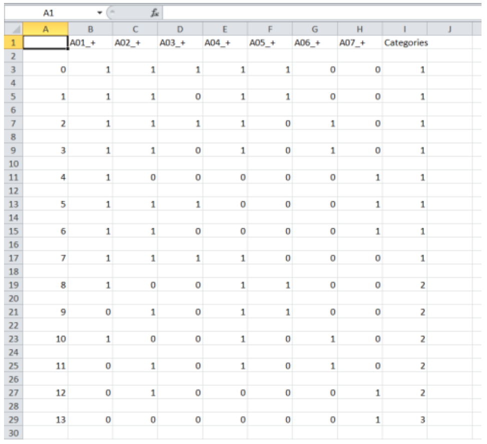


Fig.4 Example for Radial Fitted Curve on some complex dataset.

Implementation

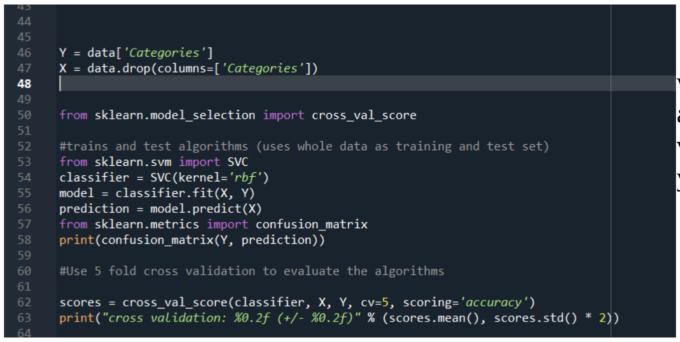
We read the dataset file and converted the dataset data from nominal values to binary values which can be recognized to the following figure "Fig 5".



	A01_+	A02_+	A03_+	A04_+	A05_+	A06_+	A07_+	Categories
0	1	1	1	1	1	0	0	1
1	1	1	0	1	1	0	0	1
2	1	1	1	1	0	1	0	1
3	1	1	0	1	0	1	0	1
4	1	0	0	0	0	0	1	1
5	1	1	1	0	0	0	1	1
6	1	1	0	0	0	0	1	1
7	1	1	1	1	0	0	0	1
8	1	0	0	1	1	0	0	2
9	0	1	0	1	1	0	0	2
10	1	0	0	1	0	1	0	2
11	0	1	0	1	0	1	0	2
12	0	1	0	0	0	0	1	2
13	0	0	0	0	0	0	1	3

Fig.5 Dataset Data Conversion from nominal values to binary values.

Then, we prepared the data before classification and then we classified using SVC function with RBF kernel function and for evaluating our algorithm that we chose, we used cross validation with K = 5 and we achieved accuracy for 83.09% as you can see in the figure below "Fig 6"



```python
Y = data['Categories']
X = data.drop(columns=['Categories'])


from sklearn.model_selection import cross_val_score

#trains and test algorithms (uses whole data as training and test set)
from sklearn.svm import SVC
classifier = SVC(kernel='rbf')
model = classifier.fit(X, Y)
prediction = model.predict(X)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(Y, prediction))

#Use 5 fold cross validation to evaluate the algorithms

scores = cross_val_score(classifier, X, Y, cv=5, scoring='accuracy')
print("cross validation: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

Fig.6 Implementation.

B. *Linear Function*

Linear Kernel is a kernel function that is used in machine learning to find a linear or a non-linear classifier or regression line.

Let us say that we have two vectors with name X1 and X2, and then the linear kernel is defined by the dot product of these two vectors [4]:

K (x1, x2) = X1. X2

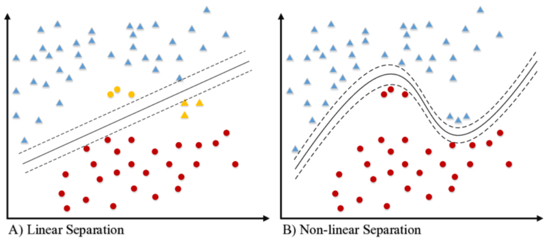
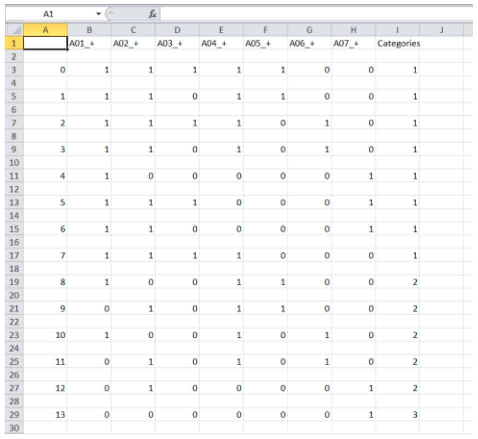


Fig.7 Example for Linear Fitted Curve.

Implementation

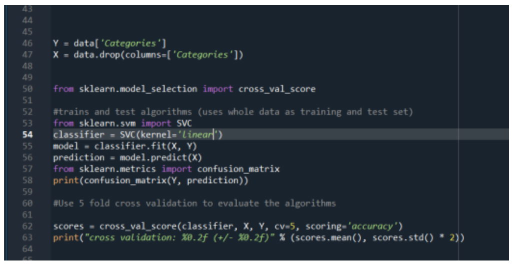
We read the dataset file and converted the dataset data from nominal values to binary values which can be recognized to the following figure "Fig 8"



	A01_+	A02_+	A03_+	A04_+	A05_+	A06_+	A07_+	Categories
0	1	1	1	1	1	0	0	1
1	1	1	0	1	1	0	0	1
2	1	1	1	1	0	1	0	1
3	1	1	0	1	0	1	0	1
4	1	0	0	0	0	0	1	1
5	1	1	1	0	0	0	1	1
6	1	1	0	0	0	0	1	1
7	1	1	1	1	0	0	0	1
8	1	0	0	1	1	0	0	2
9	0	1	0	1	1	0	0	2
10	1	0	0	1	0	1	0	2
11	0	1	0	1	0	1	0	2
12	0	1	0	0	0	0	1	2
13	0	0	0	0	0	0	1	3

Fig.8 Dataset Data Conversion from nominal values to binary values

Then, we prepared the data before classification and then we classified using SVC function with Linear kernel function and for evaluating our algorithm that we chose, we used cross validation with K = 5 and we achieved accuracy for 73% as you can see in the figure below "Fig 9"



```python
Y = data['Categories']
X = data.drop(columns=['Categories'])


from sklearn.model_selection import cross_val_score

#trains and test algorithms (uses whole data as training and test set)
from sklearn.svm import SVC
classifier = SVC(kernel='linear')
model = classifier.fit(X, Y)
prediction = model.predict(X)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(Y, prediction))

#Use 5 fold cross validation to evaluate the algorithms

scores = cross_val_score(classifier, X, Y, cv=5, scoring='accuracy')
print("cross validation: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

Fig.9 Implementation.

C. *Polynomial Function*

Polynomial Kernel is a kernel function that is used in machine learning to find a non-linear classifier or regression line and it is popular in image processing.

For degree-*d* polynomials:

$K(x, y) = (x^T y + c)^d$

Where x and y are vectors in *the input space* vectors of features computed from training or test samples and $c \geq 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial and T is a user-specified parameter [5].

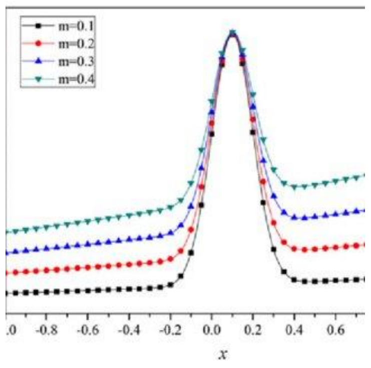
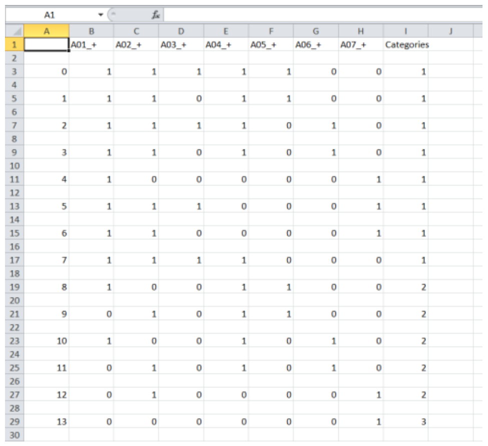


Fig.10 Example for Polynomial Fitted Curve.

Implementation

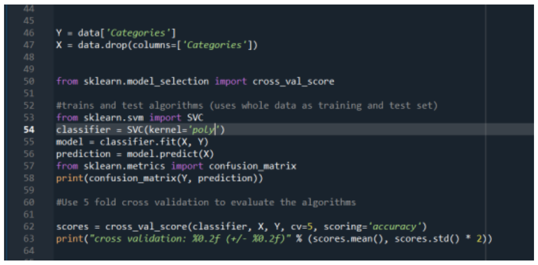
We read the dataset file and converted the dataset data from nominal values to binary values which can be recognized to the following figure "Fig 11".



	A01_+	A02_+	A03_+	A04_+	A05_+	A06_+	A07_+	Categories
0	1	1	1	1	1	0	0	1
1	1	1	0	1	1	0	0	1
2	1	1	1	1	0	1	0	1
3	1	1	0	1	0	1	0	1
4	1	0	0	0	0	0	1	1
5	1	1	1	0	0	0	1	1
6	1	1	0	0	0	0	1	1
7	1	1	1	1	0	0	0	1
8	1	0	0	1	1	0	0	2
9	0	1	0	1	1	0	0	2
10	1	0	0	1	0	1	0	2
11	0	1	0	1	0	1	0	2
12	0	1	0	0	0	0	1	2
13	0	0	0	0	0	0	1	3

Fig.11 Dataset Data Conversion from nominal values to binary values

Then, we did reprocess (Data Preparation) for the data before classification and then we classified it using SVC function with Linear kernel function and for evaluating our algorithm that we chose, in this case we used cross validation with K = 5 and we achieved accuracy for 77% as you can see in the figure below "Fig 12".



```
Y = data['Categories']
X = data.drop(columns=['Categories'])


from sklearn.model_selection import cross_val_score

#trains and test algorithms (uses whole data as training and test set)
from sklearn.svm import SVC
classifier = SVC(kernel='poly')
model = classifier.fit(X, Y)
prediction = model.predict(X)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(Y, prediction))

#Use 5 fold cross validation to evaluate the algorithms

scores = cross_val_score(classifier, X, Y, cv=5, scoring='accuracy')
print("cross validation: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))
```

Fig.12 Implementation

V. Results

Results from three major kernel functions with each other through their experimental results are analysed, and RBF (Radial Basis Function) has given the highest accuracy. Hence RBF is the best to be implemented in the proposed system.

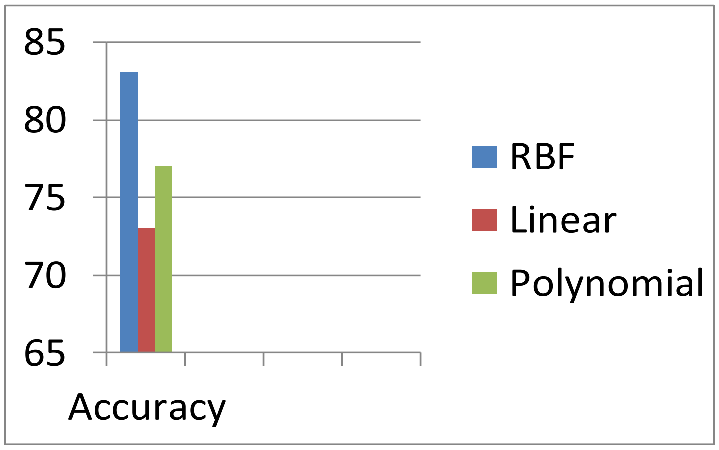


Fig.13 Graphical Representation of Accuracy.

VI. Conclusion

Now, we deduce that the best kernel function of them all without a doubt is Radial Basis Function with an accuracy score of 83.09% but that will not help that much in medical diagnosis because we need at least to accumulate an accuracy of 90% to say that we were successful with the experiment.

Coronavirus prediction which uses Machine learning algorithm provides users a prediction result if the user has Corona Virus. In this proposed method RBF was used because of its efficiency and accuracy.

REFERENCES

[1] Corona Virus Symptoms Centers for Disease Control and Prevention. https://www.cdc.gov/coronavirus/2019-ncov/symptoms-testing/symptoms.html

[2] Predictive Data Mining for Medical Diagnosis, International Journal of Computer Applications (0975 – 8887) Volume 17– No.8, March 2011, Ujma Ansari Professor Raipur Institute of Technology Raipur, Chhattisgarh, India

[3] Radial Basis Function Definition and Equation, https://www.geeksforgeeks.org/radial-basis-function-kernel-machine-learning/

[4] Linear Function Definition and Equation, https://www.educba.com/kernel-methods/

[5] Polynomial Function Definition and Equation, https://data-flair.training/blogs/svm-kernel-functions/

[6] Surveillance Test Dataset https://archive.ics.uci.edu/ml/datasets/COVID-19+Surveillance