WEATHER PREDICTION USING SUPPORT VECTOR MACHINES

ABSTRACT

This project is about Support Vector Machines and its application. We are using SVM to predict the weather using sklearn library in python. The code is drafted in an ipynb file. SVM classifier is used as the training algorithm using weather data collect from a resource. We have achieved an accuracy of 79.6%. Then we implemented our model to predict the current weather.

Index Terms – Support Vector Machine(SVM), sklearn, Support Vector Classification (SVC).

I. INTRODUCTION

A support vector machine (SVM) is a type of supervised machine learning algorithm that is used to solve two-group classification problems. It works by providing it with sets of labelled training data for each category and then using these data points to categorize new unseen data.

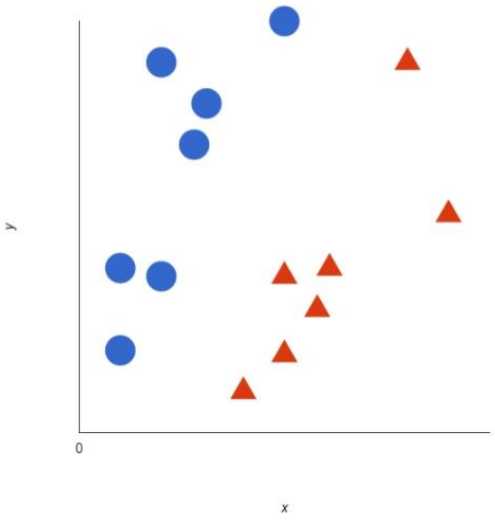
SVM algorithms are often faster than neural networks because they use relatively simple mathematical models that require fewer computations. Additionally, SVM algorithms are better suited to limited datasets because they are adept at finding patterns in small data and can be successfully trained on datasets with only a few thousand samples. This is because SVM algorithms can make use of kernel functions and other features to maximize their performance on small datasets.

The Support Vector Machine (SVM) algorithm is used to identify a hyperplane which clearly classifies a set of data points. The hyperplane's dimension is determined by the number of features; for two features, it is a line, for three features, it is a 2-D plane, and for more than three features, it is difficult to represent. The goal of SVM is to find an N-dimensional hyperplane to clearly classify the data points.

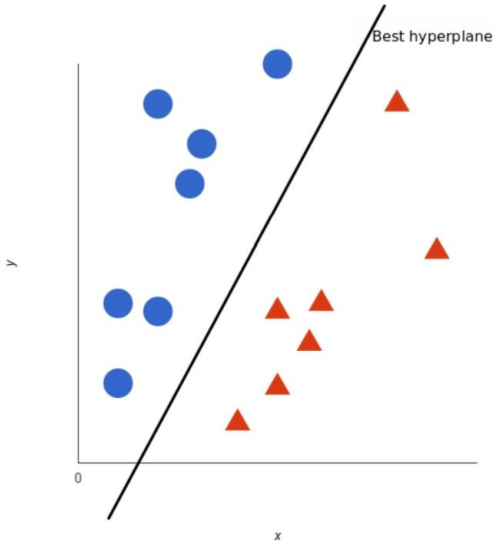
The method is utilised for a variety of applications including text categorization, face and handwriting recognition, gene identification, and more. A support vector machine (SVM) is a supervised ML algorithm that performs classification or regression tasks by constructing a divider that separates data in two categories. The optimal divider is the one which is in equal distance from the boundaries of each group.

LINEAR DATA

The basics of Support Vector Machines can be best understood by an example. Suppose we have two labels: red and blue, and our dataset has two features: x and y. We need a classifier that, when given a pair of (x, y) coordinates, will output whether it is red or blue. We can then plot our already labelled training data on a plane.



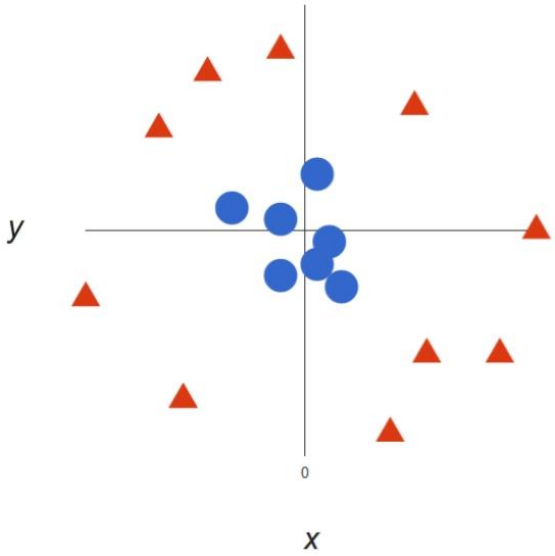
A support vector machine uses data points to generate a hyperplane that best divides the data into different categories. In two dimensions, this hyperplane is a line and serves as the decision boundary. Anything that is on one side of the line is classified as one colour (e.g., blue) and anything that is on the other side is classified as the other colour (e.g., red).



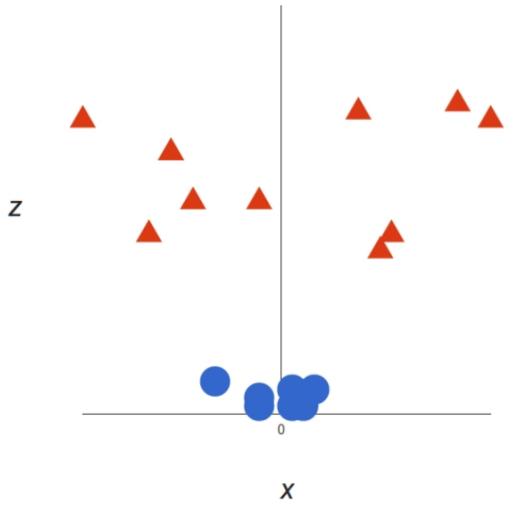
NON-LINEAR DATA

Classifying linear data was easy, since clearly the data was linearly separable — we could draw a straight line to separate *red* and *blue*. But take a look at this case.

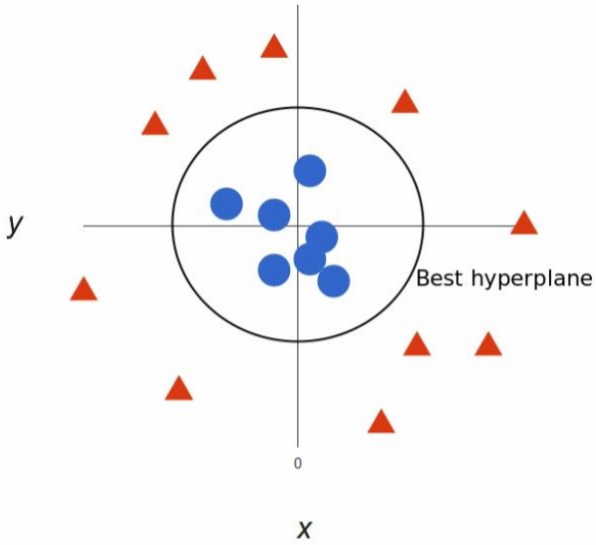
It's pretty clear that there's not a linear decision boundary (a single straight line that separates both tags). However, the vectors are very clearly segregated and it looks as though it should be easy to separate them.



We will be adding a third dimension. Up until now we had two dimensions: *x* and *y*. We create a new *z* dimension, and we rule that it be calculated a certain way that is convenient for us: $z = x^2 + y^2$. This will give us a three-dimensional space. Taking a slice of that space, it looks like this:



Note that since we are in three dimensions now, the hyperplane is a plane parallel to the *x* axis at a certain *z* (let's say $z = 1$).



Our decision boundary is a circumference of radius 1, which separates both tags using SVM.

II. WEATHER PREDICTION USING SVM

In this prediction, the Radial basis function is best. The choice of Kernel functions is not governed by a single model. The only way to define the kernel function is by experimental comparison, based on the traits of particular samples, for better parameters and better generalisation performance. While some kernel functions perform satisfactorily, others perform poorly. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Since the value of the RBF kernel decreases with distance and ranges between zero (in the limit) and one (when x = x'), it has a ready interpretation as a similarity measure. The feature space of the kernel has an infinite number of dimensions.

The proper kernel function can be chosen by utilizing the cross-validation method and selecting the one that has the lowest inductive error. With the correct parameters, improved accuracy can be obtained for all kernel functions, but the radial basis function (RBF) is usually the most effective option.

II.(A) Process of Prediction

The main steps of predicting the weather are as follows. First, we feed the data into the program, and a series of pre-processing is carried out to extract a set of feature vectors. Then these are sent to the training I/O of SVM to train the parameters and support vectors.

Here, the data is put under square root transformation. The square root transformation makes it easier to analyse the data by making it more spread out and easier to interpret. When data is concentrated in a small range, it can be difficult to distinguish patterns or to draw meaningful conclusions from it. Applying a square root transformation to the data makes it more spread out, which makes it easier to interpret the data and draw conclusions from it.

It changes the shape of the data by redistributing the points across the range in a different way. This makes it easier to identify patterns and draw conclusions from the data, even though the values themselves may appear to be smaller.

At last, some data is inputted from the user and used to predict the results by observing the patterns of the previous data which was made by the SVC.

II.(B) About the Data Used

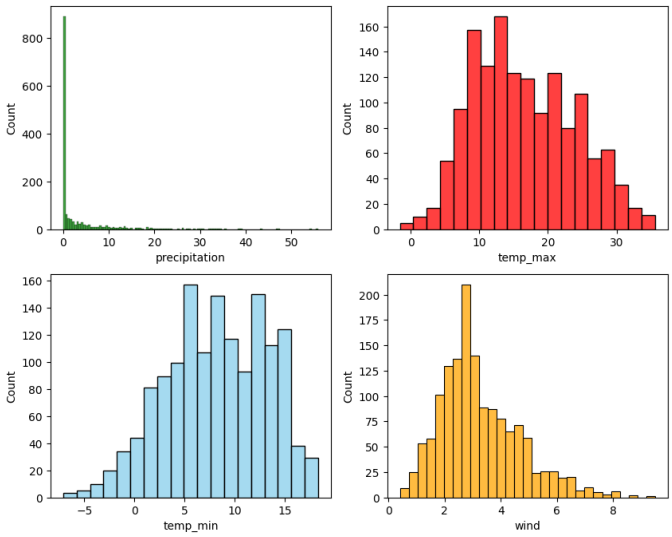
The data used contains 1461 cases with 6 variables namely Date (categorical), Precipitation (numerical), Maximum and Minimum Temperature (numerical), Wind (numerical) and Weather (categorical). While training the model, Date variable is excluded and Weather data is labelled using Label Encoder library in Python. Also, the data has the following attributes:

```
Percent of Rain:      43.87 %
Percent of Sun:       43.80 %
Percent of Drizzle:   3.62 %
Percent of Snow:      1.77 %
Percent of Fog:       6.91 %
```

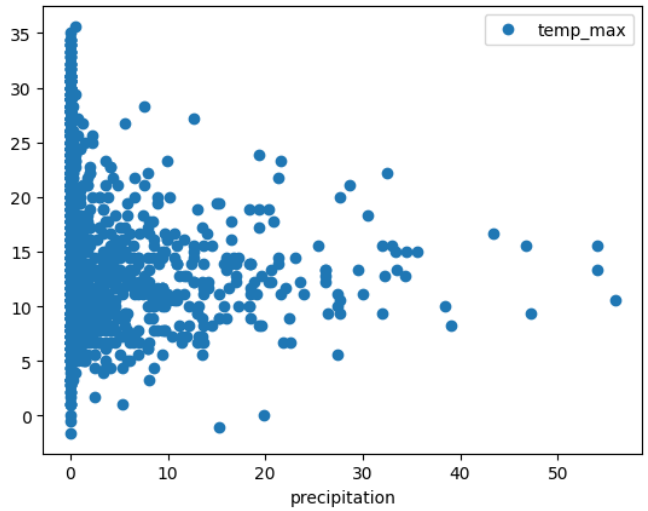
From the data, the set contains higher amount of data with the weather detail of *Rain and Sun* and it also have some additional like *drizzle, snow and fog*.

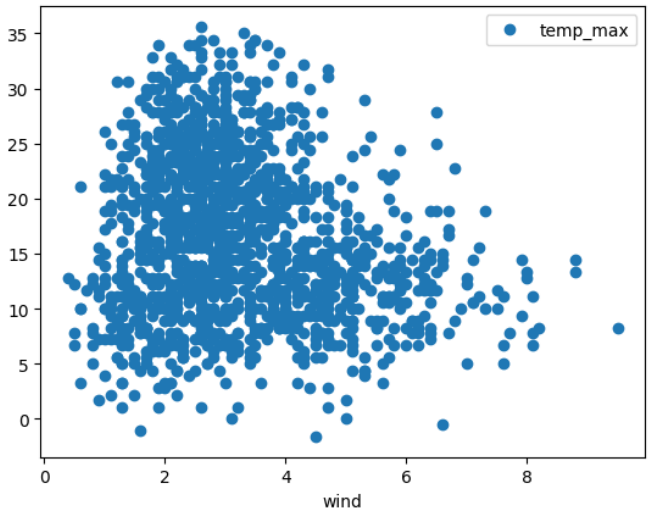
The description of the variable is as given below:

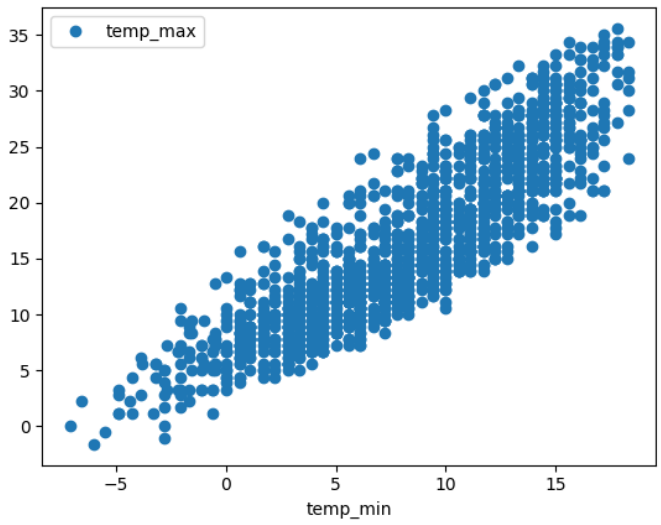
	precipitation	temp_max	temp_min	wind
count	1461.000000	1461.000000	1461.000000	1461.000000
mean	3.029432	16.439083	8.234771	3.241136
std	6.680194	7.349758	5.023004	1.437825
min	0.000000	-1.600000	-7.100000	0.400000
25%	0.000000	10.600000	4.400000	2.200000
50%	0.000000	15.600000	8.300000	3.000000
75%	2.800000	22.200000	12.200000	4.000000
max	55.900000	35.600000	18.300000	9.500000



Also, the scatter plot of Precipitation, Wind, and Minimum Temperature with respect to Maximum Temperature is as follows:







III. Python Program used for Prediction

The data is trained because it allows the model to learn from the data and make accurate predictions. Without training, the model would not be able to accurately predict the labels of unseen data. Training the data also helps the model to identify the most important features in the data and how they relate to the labels, which can help improve the accuracy of the predictions.

```
x=((df.loc[:,df.columns!="weather"]).astype(int))
.values[:,0:]
y=df["weather"].values
x_train,x_test,y_train,y_test=train_test_split(x,y
,test_size=0.1,random_state=2)
```

Next, the data is classified using Support Vector Classifier (SVC). It evaluates its performance on the test data, and calculates the accuracy of the model. The fit() function is used to train the model on the training data, and the predict() function is used to make predictions on the test data. The score() function is then used to calculate the accuracy of the model.

```
fun = SVC()
fun.fit(x_train, y_train)
y_pred = fun.predict(x_test)
fun.fit(x_train,y_train)
print("SVM
Accuracy:",(fun.score(x_test,y_test)*100),"%")
```

In the final step, user's input is taken and using SVC, the type of weather is being displayed.

```
precip = float(input("Enter Precipitaion"))
max_t = float(input("Enter Maximum Temperature"))
min_t = float(input("Enter Minimum Temperature"))
wind = float(input("Enter Wind Speed"))
dat = [[precip, max_t, min_t, wind]]
ot=fun.predict(dat)
print("The weather is:")
if(ot==0):
    print("Drizzle")
elif(ot==1):
    print("Fog")
elif(ot==2):
    print("Rain")
elif(ot==3):
    print("snow")
else:
    print("Sun")
```

So, when the user gives the input, for example [1.140175, 8.9, 2.8, 2.469818], the output turns to be:

"The weather is: Rain"

IV. Conclusion

The project of predicting the weather using Support Vector Machines (SVMs) was a success. The model was trained on a dataset of weather conditions and was able to accurately predict the type of weather given the input data. The input data included precipitation, maximum temperature, minimum temperature, and wind speed, and the model was able to accurately predict the type of weather with an accuracy of 77.51%.

This accuracy was achieved due to the use of the radial basis function (RBF) kernel, which allowed the model to transform non-linear data into a higher-dimensional space for better classification. The model was also able to identify the most important features in the data and how they related to the labels, which allowed it to make more accurate predictions. In conclusion, the project of predicting the weather using SVMs was a success, and its accuracy can be improved further by tuning the parameters or using other techniques.

In conclusion, this project has demonstrated the potential of SVMs for predicting weather conditions, and it can be further improved and used in real-world applications.

REFERENCES

[1] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[2] https://www.techtarget.com/whatis/definition/support-vector-machine-SVM

[3] https://techvidvan.com/tutorials/svm-kernel-functions/

The weather data was downloaded from the below link:

[4] https://www.kaggle.com/datasets/ananthr1/weather-prediction