
**Faculty of Computing and Information Technology**
**Department of Computer Science and Artificial Intelligence**

# Applied machine learning

## Weather Prediction project

**Supervisor**
Dr. Roshayu Mohammed

**By**

Ghadeer Alharbi

# Table of content

# Table of figures



# Table of equations

## Abstract

Weather plays a very important role in many important production areas, such as agriculture. Weather change is now expensive, so the old weather prediction is becoming less and less relevant but are still perceived as disruptive. This is why it is very important to embellish and modify the weather prediction model. Weather is one of the biggest natural barriers to all aspects of our lives in the world; we must take into account weather conditions, including humidity, rain, temperature, and other defenses against bad weather. Therefore, this paper focuses on the development of a machine-learning model for a weather prediction system using python that can provide more accurate information allows people to make informed decisions. The machine learning algorithms used in this paper are K Neighbors Classifier, Decision Tree Classifier, Support Vector Machine, Logistic Regression, and the Gaussian NB algorithm. The used database is from Kaggle. The results of the prediction show that the Gaussian NB algorithm outperforms other comparable algorithms.

## 1. Introduction

Machine learning (ML) has the capabilities associated with human intelligence while being able to learn and refine its analysis through computer algorithms. These algorithms use large input and output data sets to identify patterns effectively so that the machine can make recommendations or decisions on its own. After a sufficient number of iterations and changes to the algorithm, the machine is able to accept input data and predict the output. The results are compared to a set of known outputs to assess the accuracy of the algorithm, which is iteratively adjusted to refine its ability to predict other outputs [1, 2]. Weather conditions are changing rapidly and continuously all over the world. Accurate weather forecasts are essential in today's world. We rely heavily on weather forecasts for everything we do, from agriculture to manufacturing, from travel to daily commutes. Because the whole worlds is subject to constant environmental change and its consequences, accurate weather forecasts are critical for smooth daily operations and rapid response [3].

For decades, artificial neural networks have been used in meteorology and climatology. They have been used successfully in studies of climate attribution [4] and El Niño prediction [5, 6]. Hsieh and Tang provide an excellent overview of the first applications. Furthermore, as they are used to map weather conditions at a large scale in regional domains, and closely related support vector machines have been shown to be effective in scaling global weather models [7], where they describe the large-scale regional atmospheric climate. Other applications include cloud classification, tornado detection and forecasting, radar quality control, and decision-making [8]. Deep convolutional networks, a type of artificial neural network, have recently demonstrated excellent performance in a variety of non-linear problems, such as image recognition [9], and have also been applied to extreme weather data in climate modelling. Many important efforts have been made, with positive results, to forecast the weather using statistical models, such as machine learning techniques. The input data for weather prediction are high-density time series collected from various weather stations across the country. Chen and Hwang [10, l1] proposed a fuzzy time series model to predict the temperature based on historical data presented as linguistic values. Another study [12] showed that a set of artificial neural networks (ANNs) can successfully learn weather patterns based on meteorological parameters. In another study [13], a neural network based on chaotic wavelets was proposed for short-term wind prediction from LIDAR data. An integrated fuzzy wavelet ANN and logic model for long-term rainfall prediction [l4]. A study on drought prediction using NN, NN wavelet, and support vector regression (SVR) [15]. An artificial neural network-based rainfall forecasting model.

The paper is structured as follows: In Section 3 presents related work, Section 4 presents the research methodology, Section 5 presents the discussion of the findings, and the conclusion is reported in Section 6.

## 2. Related work

Machine learning can provide an alternative approach to forecast uncertainty in weather forecasting given the large-scale atmospheric state upon initialization. The proposed method in [16] based on deep learning with artificial convolutional neural networks that are trained on weather forecasts. While this method has less skill than aggregate weather forecasting models, it is computationally very efficient and outperforms alternative methods that do not involve making numerical forecasts. Machine learning algorithms estimate the impact of weather variables such as temperature and humidity on the transmission of COVID-19 in [17]. They extract the relationship between the number of confirmed cases and the weather variables in certain regions. Weather variables are more relevant in predicting the mortality rate when compared to population, age, and urbanization.

The uncertainty in weather forecast models is due to the interaction of multiple physical processes. Machine learning techniques are used to learn the relationships between the choice of physical processes and the resulting forecast errors. I examine the estimation of systematic model errors in output quantities of interest at future times and the use of this information to improve the model forecasts. To address these questions in [18], employ two machine-learning approaches: random forests and artificial neural networks. Numerical experiments are carried out with the Weather Research and Forecasting (WRF) model. The output of interest is the precipitation of the model, a variable that is extremely important and difficult to predict. The experiments demonstrate the powerful potential of machine learning approaches to help examine model errors. In addition, the capacity of two power lines in Northern Ireland was estimated using four machine-learning algorithms in a dynamic line classification experiment [19]. These methods are multivariate adaptive regression lines, generalized linear models, quantitative random forests, and random forests. When the results are compared with reference models, the performance of point and forecast forecasts is significantly improved. The calculation of an estimate of the margin of safety that can be used to avoid dangerous situations demonstrates the feasibility of probabilistic estimates in this area. These findings have clear implications for the protection and monitoring of transmission and distribution infrastructures, particularly in the presence of distributed power generation and/or renewable energy sources.

There are a number of studies that propose a hyper model for the task of weather prediction. For instance, the method described in [20]. NN is used as a statistical or machine learning tool to develop exact and quick simulations for laborious model physics components (model physics parameterizations).

The short- and long-wave radiation parameterizations, or total model radiation NN emulations, of the most time-consuming model physics components, are combined with the remaining deterministic elements (like model dynamics) of the original complex environmental model, a general circulation model or global climate model (GCM), to form a hybrid GCM (HGCM). Although HGCM is substantially faster, the parallel GCM and HGCM simulations yield highly similar findings. The ability to improve models is made possible by the acceleration of model calculations. Examples of built HGCMs demonstrate the novel method's viability and effectiveness in modeling complex multi-dimensional multidisciplinary systems. In [21], researchers applied neural networks to historical time series to correlate historical periods of sand deposition in semi-arid grasslands with

external climatic conditions, land-use pressures, and fire occurrence. The author has developed an innovative method of determining relationships. Reactivation and sedimentation events in the Nebraska Sandhills. The author shows how to accurately estimate the duration of vegetation growth and sediment redisposition. Individual factor sensitivity tests show that local forgings (overgrazing and forest fires) have statistically significant effects when the climate is held constant. But the most significant impact is climatic drought. Our method has great potential for predicting future landscape sensitivity to climate and land-use scenarios in a variety of potentially vulnerable drylands.

*Table 1 Summary of related works.*

| Reference | Finding or development | Technique used with the standard acronym |
|---|---|---|
| [22] | Predict hydrological variables (evapotranspiration) from meteorological variables (precipitation, temperature) around the world. | Fuzzy logic, Least Squares Support Vector Regression (LS-SVR), Adaptive Neuro-Fuzzy Inference System (ANFIS), Artificial Neural Networks (ANN), |
| [23] | Evaluate the influence of weather drivers on sand deposition in semi-arid regions. | Artificial Neural Networks (ANN). |
| [24] | Enhance the speed of global weather model parameterization. | Artificial Neural Networks (ANN). |
| [25] | Simulate weather models to broaden the range of parameter values that a climate model can handle. | A range of inversion methods, including Ensemble Kalman inversion and the Markov Chain Monte Carlo (MCMC) algorithm. |
| [26] | Evaluate the global impact of future weather change on hydrology, including river flow. | Principal Components Analysis (PCA) and Relevance Vector Machine (RVM) |

## 3. Research Methodology

Data preprocessing after loading the dataset, the information of the dataset has been checked and summed and summed the null values. The unique values of different attributes dropped the irrelevant attributes according to the evaluation dataset. The null values were recalculated and then filled. Following that, the dataset's description is examined, and the relevant attributes are checked. Then, checked the correlation on the development dataset and rationalized the values of the related attribute for regression analysis, and visualization fallowed by plotting the different graphs show in Figure 1,2. Figure 3 shows the flowchart of a developing weather prediction model.

```
fig = px.histogram(df, x='weather',y="precipitation",color='weather',histfunc="avg",text_auto=True, title='Average o
fig.show()
```

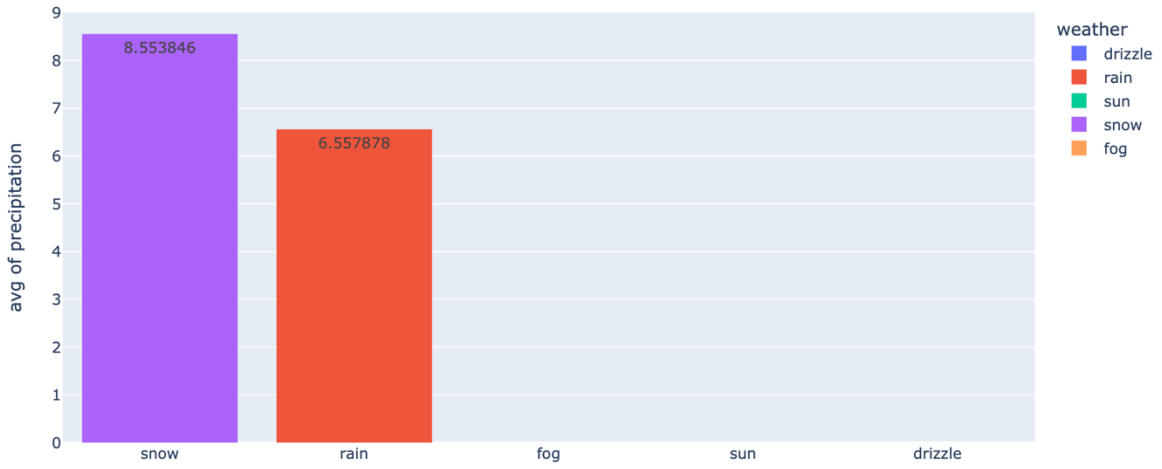



*Figure 1 The most weather state percent*

```
fig = px.pie(df, names="weather",color='weather',title='the most weather percent  ')
fig.show()
```

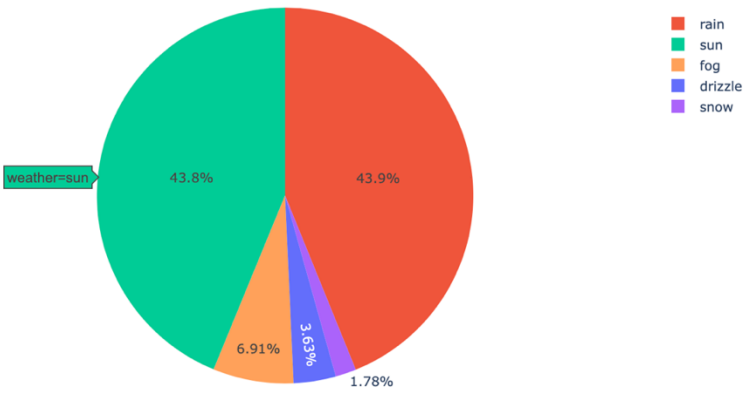



*Figure 2 Average of precipitation in each weather state*

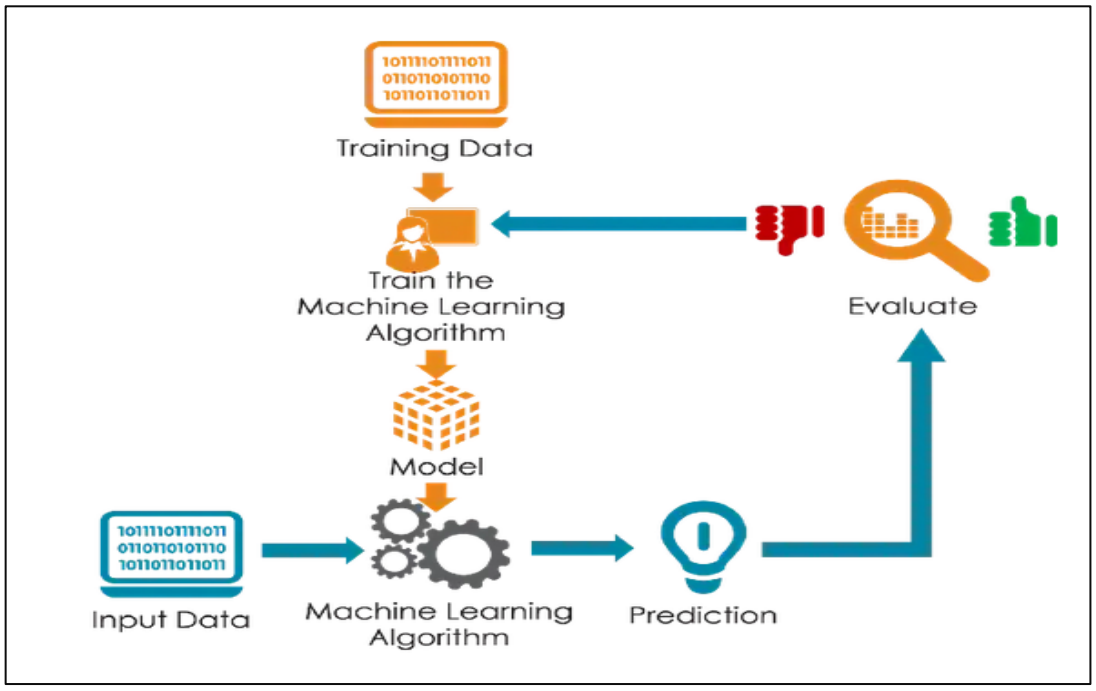



*Figure 3 Flowchart of weather prediction [27].*

## 3.1 Dataset and Pre-processing

### 3.1.1 The Dataset

The data was acquired from Kaggle online, which includes six columns. The names of the six columns in the database are date, precipitation, temp_max, temp_min, wind, and weather as show in Figure 4.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           1461 non-null   object
 1   precipitation  1461 non-null   float64
 2   temp_max       1461 non-null   float64
 3   temp_min       1461 non-null   float64
 4   wind           1461 non-null   float64
 5   weather        1461 non-null   object
dtypes: float64(4), object(2)
memory usage: 68.6+ KB
```

*Figure 4 Statistics of Dataset used.*

### 3.1.2 Dataset Preparation

Firstly, replacing null values in the seattle-weather.csv file with the mean of the column to which they belong as a step from the cleaning process.

### 3.1.3 Analysis Dataset

The subsubsection explains the dataset that is used in the paper as follows:

- ❖ Provide fast analytics for the different factors affecting weather.
- ❖ After collecting the data, a very basic cleaning of the data has been performed.
- ❖ Use different prediction and classification algorithms.
- ❖ Calculating and comparing evaluation measure.

## 3.2 Extract Features

Data extracted as dependent variables and this includes:
(`'date', 'precipitation', 'temp_max', 'temp_min', 'wind'`), where the independent target variable is (`'weather'`). Figure 5 show the code of extracting such features and also thee obtained results.

```
x = df[['precipitation', 'temp_max','temp_min','wind']].copy()
y=df[['weather']].copy()
x.head()
```

| | precipitation | temp_max | temp_min | wind |
|---|---|---|---|---|
| **0** | 0.0 | 12.8 | 5.0 | 4.7 |
| **1** | 10.9 | 10.6 | 2.8 | 4.5 |
| **2** | 0.8 | 11.7 | 7.2 | 2.3 |
| **3** | 20.3 | 12.2 | 5.6 | 4.7 |
| **4** | 1.3 | 8.9 | 2.8 | 6.1 |

*Figure 5 Extract Features code and result*

### 3.3 Building the Proposed Models

In this study, five machine learning algorithms were used to develop the proposed model. They are addressed in depth below:

#### 3.3.1 Support vector machines

Support vector machine (SVM) is a type of linear classifier that works on the margin maximization principle [28]. Such classifier minimizes structural risks, increasing the complexity of the classifier in order to achieve excellent generalization performance. The SVM performs classification by constructing the hyperplane that best divides the data into two categories in a higher dimensional space. Figure 6 describes the formulation of SVM.

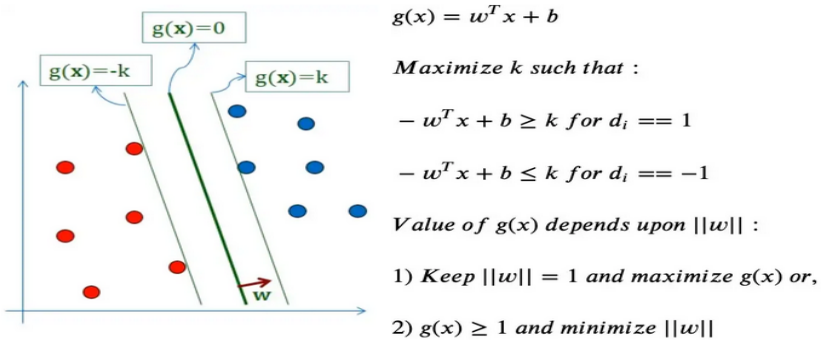



*Figure 6 the formula of SVM [29].*

#### 3.3.2 K-Neighbors Classifier

K-Neighbor is a simple Machine Learning algorithm that uses also the Supervised Learning technique [30]. K- Neighbor presumes similarity between the new case/data and existing cases and places the new case in the category that is most similar to the existing categories. Furthermore, this algorithm stores all available data and categorizes a new data point based on similarity. This means that when new data arrives, it can be quickly classified into a well-suited category using the K- Neighbor algorithm as show in below equation.

$$h(x) = \mathrm{mod}\, e(\{y'' : (x'', y'') \in S_x\})$$

**Equation 1 the K- Neighbor classifier.**

#### 3.3.3 Logistic Regression

Logistic regression [31] is frequently used in classification and predictive analytics. Based on a set of independent variables, logistic regression calculates the likelihood of an event occurring, such as voting or not

voting. The dependent variable is bounded between 0 and 1 because the outcome is a probability. A logistic transformation is used on the odds in logistic regression, which is the probability of success divided by the probability of failure. This logistic function is also known as the log odds or the natural logarithm of odds, and it is represented in Equation 2:

$$\log(p_j) = 1/(1+\exp(-p_j))$$

$$In(p_j/(1-p_j)) = Beta\_0 + Beta\_1 * x\_1 + ..... + B\_k * K\_k$$

**Equation 2 the logistic regression.**

### 3.3.4 Decision Tree Classifier

Decision tree classifier is a machine-learning algorithm that makes decisions using human like rules. It can solve both classification and regression problems but is most often used to solve classification problems. In a tree-structured classifier, internal nodes represent data set features, branches represent decision rules, and leaf nodes represent results. Two below equations explain the formula of this algorithm [32].

$$G(node) = \sum_{j=1}^{n} p_j (1 - p_j)$$

**Equation 3 the Gini Impurity of a node.**

$$Entropy(node) = -\sum_{i=1}^{n} p_j \log(p_j)$$

**Equation 4 the Entropy of a node.**

### 3.3.5 Gaussian Naïve Bayes

Gaussian Naive Bayes (GNB) is a machine learning technique that is based on a probabilistic approach and a Gaussian distribution. GNB assumes that each parameter can predict the output variable independently. The final prediction is the combination of all parameter predictions, which returns a probability of the dependent variable being classified in each group. The group with the highest probability receives the final classification. The formula of the GNB algorithm as follows [15]:

$$f(y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$$

**Equation 5 the formula of the GNB.**

### 3.3.6 Artificial Neural Networks

The term "artificial neural network " refers to a biologically inspired subfield of artificial intelligence modeled after the brain. An artificial neural network is usually a computational network based on biological neural networks that build the structure of the human brain. Just like a human brain has neurons interconnected with each other, artificial neural networks also have neurons that are connected with each other at various layers of the networks. These neurons are known as nodes [31], Figure 7 shows thee default architecture of ANN.

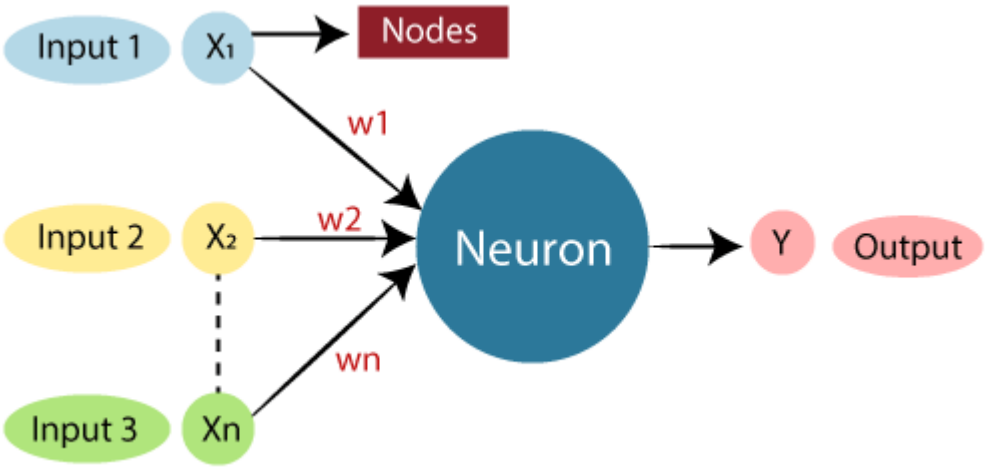



*Figure 7 Artificial Neural Network [31]*

### 3.4 Model Training and Evaluation

Different partitioning was used based on the model's analysis results, with the most efficient partition using 70% of the data for training and 30% for testing how well the model performed. Because this was a classification problem, using four matrices that represented the output as a probability to assess how well the fifth algorithms performed.

The root mean square error (RMSE), mean square error (MSE), mean absolute error (MAE), and mean absolute error/mean absolute error (MAE/MAE) were the accuracy measures and used to assess how well the algorithm performed (MAPE). This is the most commonly used accuracy matrix, and it indicates how close the observed data values are to the model's projected values, as measured by the model's accuracy matrix. This is known as the RMSE. The root mean square error is depicted in Equation 7. (RMSE). Because the dataset contains a large number of samples, the RMSE is more accurate. Figure 8 below show the code of the developed algorithm of the model.

$$RMSE = \sqrt{\sum_{j=1}^{n} \frac{(\hat{y}_j - y_j)^2}{n}}$$

**Equation 6 the RMSE.**

```
models = []
models.append(("SVM",SVC()))
models.append(("NB",GaussianNB()))
models.append(("KNN",KNeighborsClassifier()))
models.append(("dt",DecisionTreeClassifier()))
models.append(("LR",LogisticRegression()))
models.append(("ANN",MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1)))

results = []
names = []
m=[]
for name,model in models:
    kfold = KFold(n_splits=10)
    cv_result = cross_val_score(model,X_train,Y_train, cv = kfold,scoring = "accuracy")
    names.append(name)
    results.append(cv_result)
for i in range(len(names)):
    m.append(results[i].mean())
    print('accurcy of ',names[i],':' ,results[i].mean())
```

```
accurcy of  SVM : 0.7903113822848727
accurcy of  NB : 0.8670418682937091
accurcy of  KNN : 0.7617609930570166
accurcy of  dt : 0.7781190826846203
accurcy of  LR : 0.859899011150852
accurcy of  ANN : 0.8598884914790658
```

*Figure 8 The code of the developed algorithm of the model*

## 4. Results and Discussion

This section shows the results of the experimental analysis. The dataset has 1461 date, 111 precipitation, 67 temp_max, 55 temp_min, 79 wind, and 5 weather. Figure 9 illustrates the most common meteorological phenomena. Therefore, the target variable is the weather, which contains five values ,namely , drizzle, rain, sun, snow, and fog.

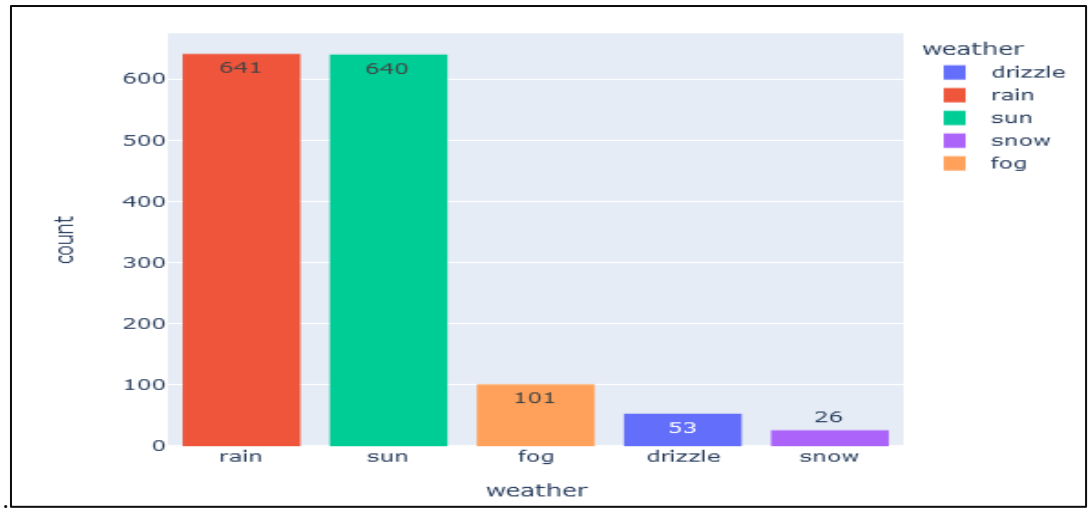



Figure 9 Meteorological phenomena.

The results of the evaluation metrics proved that the Gaussian Naive Bayes algorithm is superior compared to the ANN, Decision Tree Classifier, Logistic regression, K-Neighbor, and

SVM with lower MAE of 39.17%. Figure 10 shows the code and the highest accuracy value of the NB model which is 86.70%. Figure 11 illustrates wind averages for different weather conditions.

```
X_train,X_test,Y_train,Y_test = train_test_split(x,y, random_state = 22, test_size = 0.3)
nb = GaussianNB()
nb.fit(X_train,Y_train)
predictions = nb.predict(X_test)
# accuracy of svm(evaluate accuracy)
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error,r2_score,mean_squared_error
print("accuracy_score :",accuracy_score(Y_test,predictions))
print("MAE :", mean_absolute_error(Y_test,predictions))
```

```
accuracy_score : 0.8633257403189066
MAE : 0.3917995444191344
```

Figure 10 Code of accuracy rate and MAE

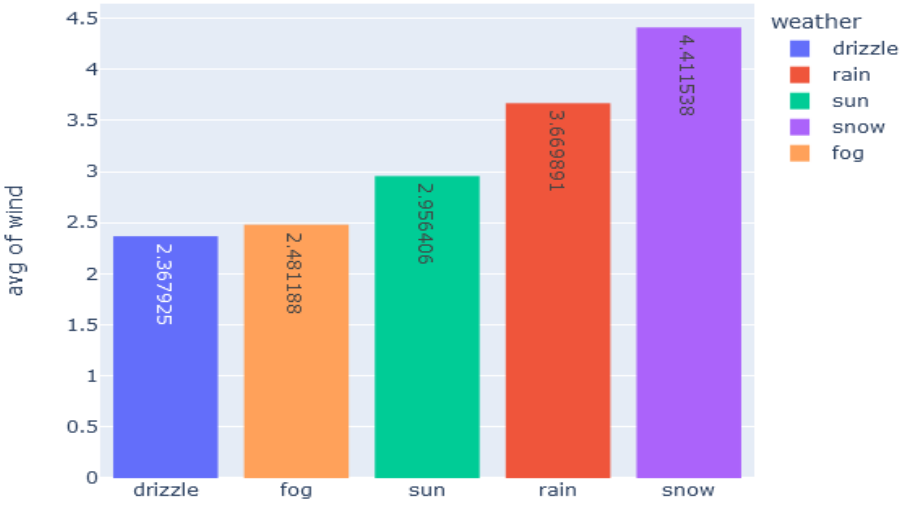



Figure 11 Wind averages.

Furthermore, the Artificial neural network comes in the second position with an accuracy value of 85.98%, and the Logistic regression comes in the third position with an accuracy value of 85.88%, followed by SVM with an accuracy value of 79%, followed by Decision Tree Classifier with accuracy 78%. But the K-Neighbor classifier comes in the last position with an accuracy value of 76.07%. Figure 12 demonstrate the correlation matrix of numerical variables and figure 13 shows the accuracy of different classification models, figure 14 below show the code of weather prediction

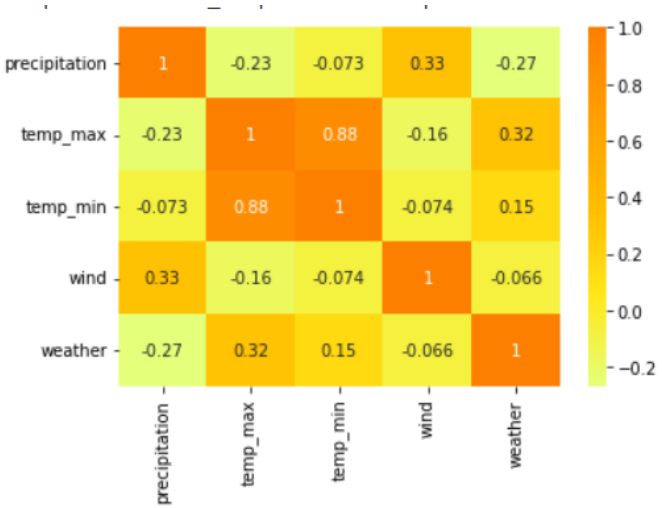



Figure 12 Correlation matrix of numerical variables.

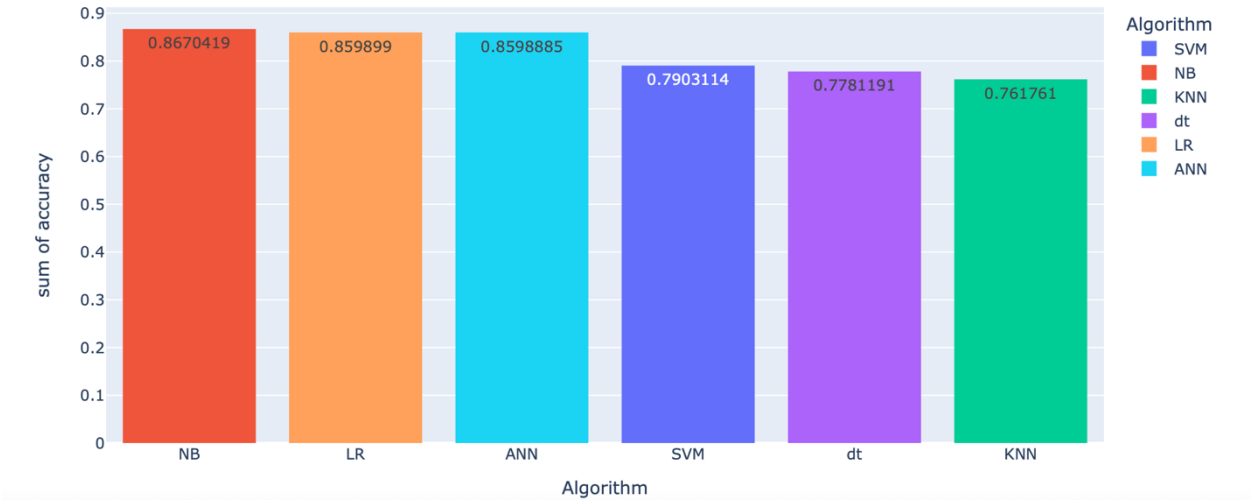



Figure 13 Accuracies of different classification models.

```
# entry shape ['precipitation', 'temp_max','temp_min','wind']
pred=nb.predict([[0,10,9,2]])

Interest_prediction=labelencoder.inverse_transform(pred)

print("weather Prediction is:",Interest_prediction)
```

```
weather Prediction is: ['sun']
```

*Figure 14 Weather prediction code*

## 5. Conclusion and Future Works

This paper provides a set of experiments using a different machine learning models to predicate weather accurately. As mentioned previously, weather plays a very important role in many important production areas, such as agriculture. The main objective of this paper is to construct a well predictive model through the use of machine learning techniques. In particular, this study utilized five different machine learning algorithms: K-Neighbors Classifier, Decision Tree Classifier, Support Vector Machine, Logistic Regression, and the Gaussian NB algorithm GNB. We noticed that the GNB model had the highest accuracy with the lowest RMSE. Additionally, the GNB model surpasses the LR model on a performance basis. On the other hand, the GNB model is considered to be the optimal model for weather predication in our dataset.

As a future direction, another feature set such as date will be used in the model training and testing and this would presumably enhance the model's performance. Also, a novel and real dataset that will be built specifically for this task, can be beneficial to examine the performance of the mode's performance on real dataset.

## References


[1] Helm, J. M., Swiergosz, A. M., Haeberle, H. S., Karnuta, J. M., Schaffer, J. L., Krebs, V. E., ... & Ramkumar, P. N. (2020). Machine learning and artificial intelligence: definitions, applications, and future directions. Current reviews in musculoskeletal medicine, 13(1), 69-76.

[2] Bini, S. A. (2018). Artificial intelligence, machine learning, deep learning, and cognitive computing: what do these terms mean and how will they impact health care. The Journal of arthroplasty, 33(8), 2358-2361.

[3] Lynch, P. (2008). The origins of computer weather prediction and climate modeling. Journal of computational physics, 227(7), 3431-3444.

[4] Pasini, A., Racca, P., Amendola, S., Cartocci, G., & Cassardo, C. (2017). Attribution of recent temperature behavior reassessed by a neural-network method. Scientific reports, 7(1), 1-10.

[5] X. Zhou Tangang, F. T., Tang, B., Monahan, A. H., & Hsieh, W. W. (1998). Forecasting ENSO events: A neural network–extended EOF approach. Journal of Climate, 11(1), 29-41.

[6] Holmstrom, M., Liu, D., & Vo, C. (2016). Machine learning applied to weather forecasting. Meteorol. Appl, 10, 1-5.

[7] Ghosh, S., & Mujumdar, P. P. (2008). Statistical downscaling of GCM simulations to streamflow using relevance vector machine. Advances in water resources, 31(1), 132-146.

[8] Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., ... & Collins, W. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. arXiv preprint arXiv:1605.01156.

[9] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

[10] De Mendonça, L. M., Blanco, C. J. C., & de Oliveira Carvalho, F. (2022). Recurrent neural networks for rainfall-runoff modeling of small Amazon catchments. Modeling Earth Systems and Environment, 1-15.

[11] Domańska, D., & Wojtylak, M. (2012). Application of fuzzy time series models for forecasting pollution concentrations. Expert Systems with Applications, 39(9), 7673-7679.

[12] Maqsood, I., Khan, M. R., & Abraham, A. (2004). An ensemble of neural networks for weather forecasting. Neural Computing & Applications, 13(2), 112-122.

[13] Kwong, K. M., Max, H. Y., Raymond, S. T., & James, N. K. (2009, August). Financial

trend forecasting with fuzzy chaotic oscillatory-based neural networks (CONN). In 2009 IEEE International Conference on Fuzzy Systems (pp. 1947-1952). IEEE.

[14] Belayneh, A., & Adamowski, J. (2012). Standard precipitation index drought forecasting using neural networks, wavelet neural networks, and support vector regression. Applied computational intelligence and soft computing, 2012.

[15] Maqsood, I., Khan, M. R., & Abraham, A. (2004). An ensemble of neural networks for weather forecasting. Neural Computing & Applications, 13(2), 112-122.

[16] Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. Quarterly Journal of the Royal Meteorological Society, 144(717), 2830-2841.

[17] Malki, Z., Atlam, E. S., Hassanien, A. E., Dagnew, G., Elhosseini, M. A., & Gad, I. (2020). Association between weather data and COVID-19 pandemic predicting mortality rate: Machine learning approaches. Chaos, Solitons & Fractals, 138, 110137.

[18] Moosavi, A., Rao, V., & Sandu, A. (2021). Machine learning based algorithms for uncertainty quantification in numerical weather prediction models. Journal of Computational Science, 50, 101295.

[19] Aznarte, J. L., & Siebert, N. (2016). Dynamic line rating using numerical weather predictions and machine learning: A case study. IEEE Transactions on Power Delivery, 32(1), 335-343.

[20] Krasnopolsky, V. M., & Fox-Rabinovitz, M. S. (2006). Complex hybrid models combining deterministic and machine learning components for numerical climate modeling and weather prediction. Neural Networks, 19(2), 122-134.

[21] Goyal, M. K., Bharti, B., Quilty, J., Adamowski, J., & Pandey, A. (2014). Modeling of daily pan evaporation in sub-tropical climates using ANN, LS-SVR, Fuzzy Logic, and ANFIS. Expert systems with applications, 41(11), 5267-5276.

[22] Buckland, C. E., Bailey, R. M., & Thomas, D. S. G. (2019). Using artificial neural networks to predict future dryland responses to human and climate disturbances. Scientific reports, 9(1), 1-13.

[23] Krasnopolsky, V. M., Fox-Rabinovitz, M. S., & Chalikov, D. V. (2005). New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. Monthly Weather Review, 133(5), 1370-1383.

[24] Schneider, T., Lan, S., Stuart, A., & Teixeira, J. (2017). Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. Geophysical Research Letters, 44(24), 12-396.

[25] Ghosh, S., & Mujumdar, P. P. (2008). Statistical downscaling of GCM simulations to streamflow using relevance vector machine. Advances in water resources, 31(1), 132-146.

[26] Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR). [Internet], 9, 381-386.

[27] Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification (pp. 207-235). Springer, Boston, MA.

[28] https://towardsdatascience.com/support-vector-machine-formulation-and-derivation-b146ce89f28.

[29] Choi, S., Kim, Y. J., Briceno, S., & Mavris, D. (2016, September). Prediction of weather-induced airline delays based on machine learning algorithms. In 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC) (pp. 1-6). IEEE.

[30] Deng, F. (2020, November). Research on the applicability of weather forecast model—based on logistic regression and decision tree. In Journal of Physics: Conference Series (Vol. 1678, No. 1, p. 012110). IOP Publishing.

[31] Kwon, Y., Kwasinski, A., & Kwasinski, A. (2019). Solar irradiance forecast using naïve Bayes classifier based on publicly available weather forecasting variables. energies, 12(8), 1529.

# Appendix

**Import the library used in this project code:**

```
import pandas as pd
import matplotlib as plt
import seaborn as sns
import plotly.express as px
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import KFold
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

**Read weather dataset code:**

```
#importons le dataset
df=pd.read_csv('seattle-weather.csv')
```

```
df.head()
```

| | date | precipitation | temp_max | temp_min | wind | weather |
|---|---|---|---|---|---|---|
| 0 | 2012-01-01 | 0.0 | 12.8 | 5.0 | 4.7 | drizzle |
| 1 | 2012-01-02 | 10.9 | 10.6 | 2.8 | 4.5 | rain |
| 2 | 2012-01-03 | 0.8 | 11.7 | 7.2 | 2.3 | rain |
| 3 | 2012-01-04 | 20.3 | 12.2 | 5.6 | 4.7 | rain |
| 4 | 2012-01-05 | 1.3 | 8.9 | 2.8 | 6.1 | rain |

## Understand and Prepare the Data code:

```
print ("Rows     : " ,df.shape[0])
print ("Columns  : " ,df.shape[1])
print ("\nFeatures : \n" ,df.columns.tolist())
print ("\nMissing values :  ", df.isnull().sum().values.sum())
print ("\nUnique values :  \n",df.nunique())
```

```
Rows     :  1461
Columns  :  6

Features :
 ['date', 'precipitation', 'temp_max', 'temp_min', 'wind', 'weather']

Missing values :   0

Unique values :
 date             1461
precipitation     111
temp_max           67
temp_min           55
wind               79
weather             5
dtype: int64
```

```
df.corr()
```

| | precipitation | temp_max | temp_min | wind |
|---|---|---|---|---|
| **precipitation** | 1.000000 | -0.228555 | -0.072684 | 0.328045 |
| **temp_max** | -0.228555 | 1.000000 | 0.875687 | -0.164857 |
| **temp_min** | -0.072684 | 0.875687 | 1.000000 | -0.074185 |
| **wind** | 0.328045 | -0.164857 | -0.074185 | 1.000000 |

```
df["weather"].unique()
```

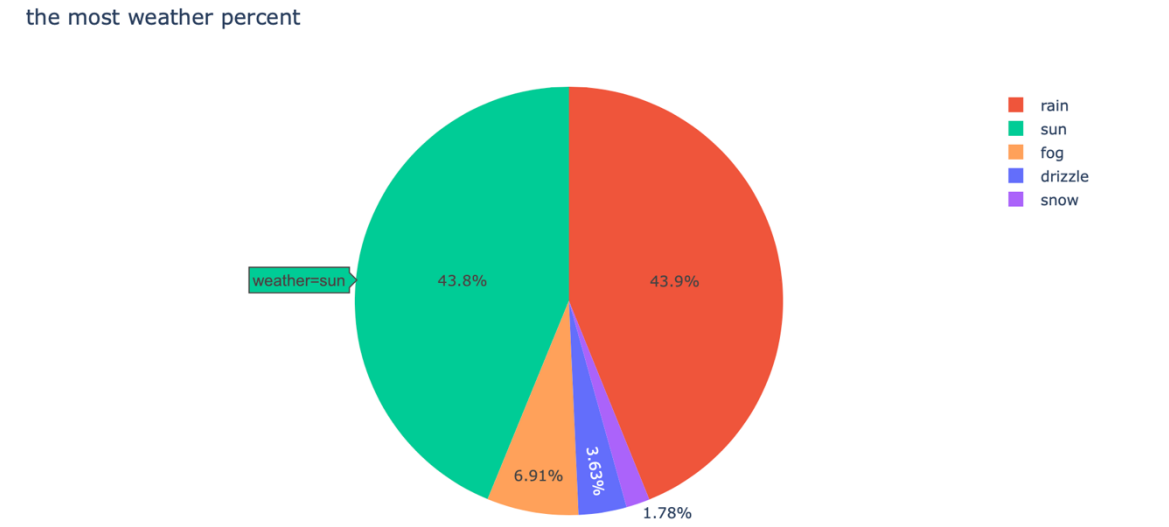
```
array(['drizzle', 'rain', 'sun', 'snow', 'fog'], dtype=object)
```

```
df.info()
```

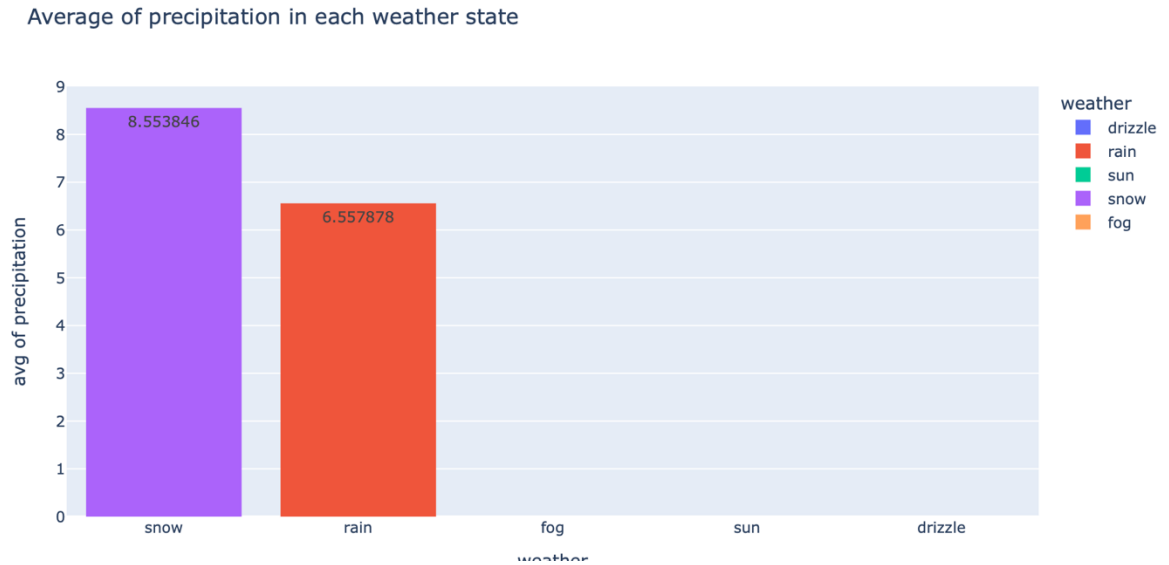
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1461 entries, 0 to 1460
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           1461 non-null   object
 1   precipitation  1461 non-null   float64
 2   temp_max       1461 non-null   float64
 3   temp_min       1461 non-null   float64
 4   wind           1461 non-null   float64
 5   weather        1461 non-null   object
dtypes: float64(4), object(2)
memory usage: 68.6+ KB
```

## Visualisation code:

```
fig = px.pie(df, names="weather",color='weather',title='the most weather percent  ')
fig.show()
```




```
fig = px.histogram(df, x='weather',y="precipitation",color='weather',histfunc="avg",text_auto=True, title='Average o
fig.show()
```



**Feature engineering and selection code:**

```
# feature engineering and selection
x = df[['precipitation', 'temp_max','temp_min','wind']].copy()
y=df[['weather']].copy()
x.head()
```

| | precipitation | temp_max | temp_min | wind |
|---|---|---|---|---|
| **0** | 0.0 | 12.8 | 5.0 | 4.7 |
| **1** | 10.9 | 10.6 | 2.8 | 4.5 |
| **2** | 0.8 | 11.7 | 7.2 | 2.3 |
| **3** | 20.3 | 12.2 | 5.6 | 4.7 |
| **4** | 1.3 | 8.9 | 2.8 | 6.1 |

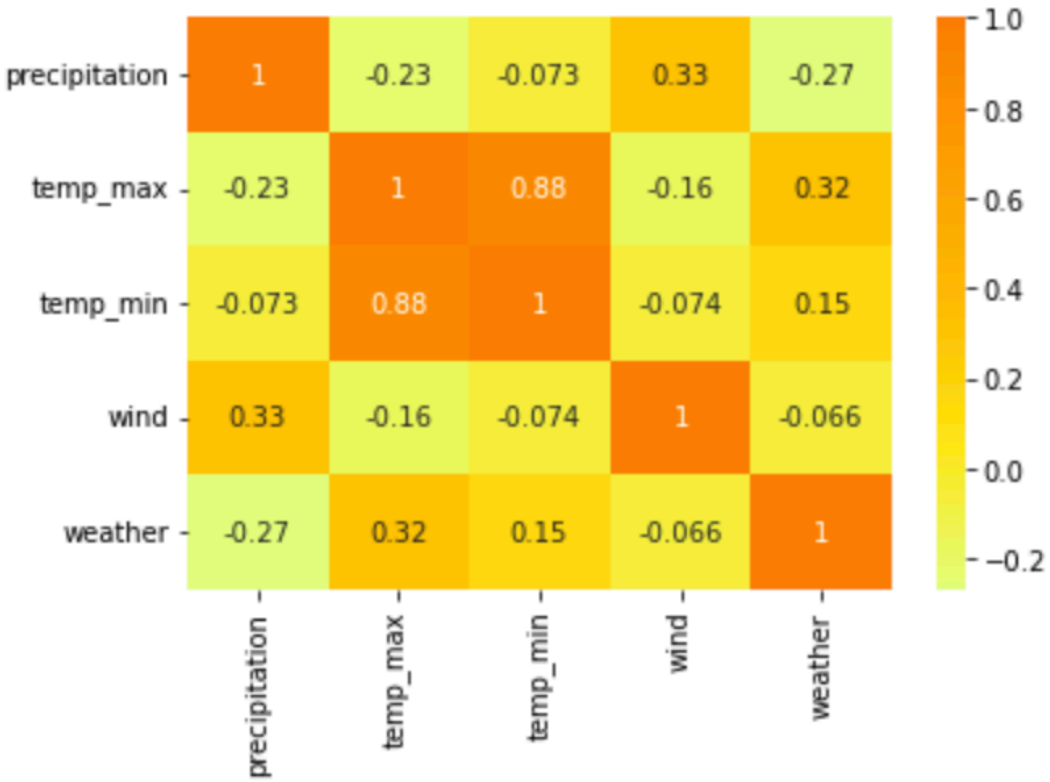
**Convert weather value to numerical value code:**

```
from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
# conver subject column to numirical value
df['weather']= labelencoder.fit_transform(df[['weather']].copy())
y=df[['weather']].copy()
```

**Correlation Matrix code :**

```
# the relationship between variables
sns.heatmap(df.corr(),cmap = 'Wistia', annot= True)
```

```
<AxesSubplot:>
```




**Split dateset with cross validation code:**

```
from sklearn.metrics import mean_absolute_error,r2_score,mean_squared_error
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(x,y, random_state = 33, test_size = 0.33)

from sklearn.neural_network import MLPClassifier
```

## Build our algorithms code :

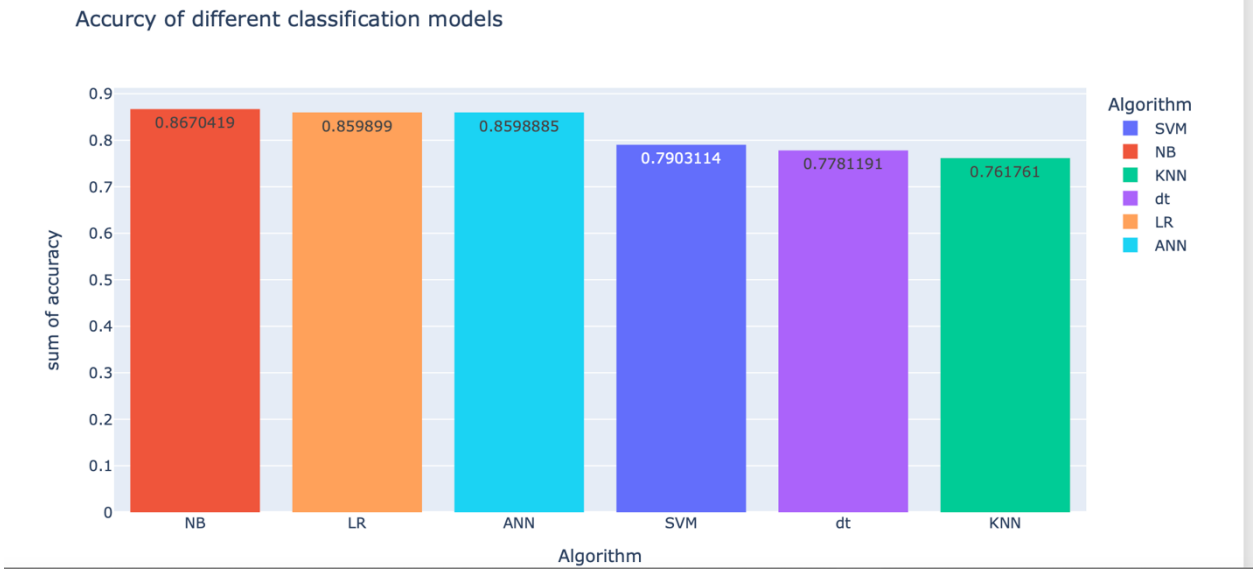
```
models = []
models.append(("SVM",SVC()))
models.append(("NB",GaussianNB()))
models.append(("KNN",KNeighborsClassifier()))
models.append(("dt",DecisionTreeClassifier()))
models.append(("LR",LogisticRegression()))
models.append(("ANN",MLPClassifier(solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(5, 2), random_state=1)))

results = []
names = []
m=[]
for name,model in models:
    kfold = KFold(n_splits=10)
    cv_result = cross_val_score(model,X_train,Y_train, cv = kfold,scoring = "accuracy")
    names.append(name)
    results.append(cv_result)
for i in range(len(names)):
    m.append(results[i].mean())
    print('accurcy of ',names[i],':' ,results[i].mean())
```

```
accurcy of  SVM : 0.7903113822848727
accurcy of  NB : 0.8670418682937091
accurcy of  KNN : 0.7617609930570166
accurcy of  dt : 0.7781190826846203
accurcy of  LR : 0.859899011150852
accurcy of  ANN : 0.8598884914790658
```

## Accurcy of different classification models histogram code :

```
accuracy = pd.DataFrame({'Algorithm': names,'accuracy': m })
fig = px.histogram(accuracy, x='Algorithm',y="accuracy",color='Algorithm',text_auto=True, title='Accurcy of differen
fig.show()
```



**Best Accuracy of Machine learning Algorithm is Naive Bayes for my dataset code:**

```
X_train,X_test,Y_train,Y_test = train_test_split(x,y, random_state = 22, test_size = 0.3)
nb = GaussianNB()
nb.fit(X_train,Y_train)
predictions = nb.predict(X_test)
# accuracy of svm(evaluate accuracy)
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_error,r2_score,mean_squared_error
print("accuracy_score :",accuracy_score(Y_test,predictions))
print("MAE :", mean_absolute_error(Y_test,predictions))
```

```
accuracy_score : 0.8633257403189066
MAE : 0.3917995444191344
```

**Classification report and confusion matrix code :**

```
from sklearn.metrics import classification_report, confusion_matrix


print('\n')
print("Precision, Recall, F1")
print('\n')
CR=classification_report(Y_test,predictions)
print(CR)
print('\n')
```

```
Precision, Recall, F1


              precision    recall  f1-score   support

           0       0.33      0.13      0.19        15
           1       0.00      0.00      0.00        25
           2       0.98      0.92      0.95       186
           3       0.62      0.56      0.59         9
           4       0.80      0.98      0.88       204

    accuracy                           0.86       439
   macro avg       0.55      0.52      0.52       439
weighted avg       0.81      0.86      0.83       439
```

```
print('\n')
print("Confusion Matrix")
print('\n')
CM=confusion_matrix(Y_test,predictions)
print(CM)
```

```
Confusion Matrix


[[  2   0   0   0  13]
 [  0   0   0   0  25]
 [  0   0 172   3  11]
 [  0   0   4   5   0]
 [  4   0   0   0 200]]
```

**Predict new data entry code:**

```
# entry shape ['precipitation', 'temp_max','temp_min','wind']
pred=nb.predict([[0,10,9,2]])

Interest_prediction=labelencoder.inverse_transform(pred)

print("weather Prediction is:",Interest_prediction)
```

```
weather Prediction is: ['sun']
```