

# RAINFALL PREDICTION THROUGH MACHINE LEARNING METHODS

**B.Raghupathi<sup>1</sup>|G.Venkatesh<sup>2</sup>|Dr.U.Naresh<sup>3</sup>|TARUNOJU KOMALIKA<sup>4</sup>**

1 ,2 & 3 Associate Professor, CSE department, Kasireddy Narayanreddy College of Engineering  
And Research, Hyderabad, TS.

4 UG SCHOLAR, CSE department, Kasireddy Narayanreddy College of Engineering And Research,  
Hyderabad, TS.

**ABSTRACT:** One of the difficult and unpredictable tasks that has a big influence on human culture is predicting rainfall. Proactively minimizing financial and human losses can be achieved by timely and precise forecasting. Based on weather data for that specific day in key Australian cities, this paper provides a series of experiments that employ popular machine learning techniques to create models that forecast whether or not it will rain tomorrow. Three components are the focus of this comparison study: pre-processing approaches, modeling methodologies, and modeling inputs. The findings compare different evaluation measures of these machine learning methods and show how reliable they are at forecasting rainfall based on weather data analysis.

**KEYWORDS:** Rain fall, Machine learning, Data, Human society.

**I.INTRODUCTION:** India's welfare is agriculture. The achievement of agriculture is dependent on rainfall. It also helps with water resources. Rainfall information in the past helps farmers better manage their crops, leading to economic growth in the country. Prediction of precipitation is beneficial to prevent flooding that saves people's lives and property. Fluctuation in the timing of precipitation and its amount makes forecasting of rainfall a problem for meteorological scientists. Forecasting is one of the utmost challenges for researchers from a variety of fields, such as weather data mining, environmental machine learning, functional hydrology, and numerical forecasting, to create a predictive model for accurate rainfall. In these problems, a common question is how to infer the past predictions and make use of future predictions. A variety of sub-processes are typically composed of the substantial process in rainfall. It is at times not promising to predict the precipitation correctly by on its global system. Climate forecasting stands out for all countries around the globe in all the benefits and services provided by the meteorological department. The job is very complicated because it needs specific numbers and all signals are intimated without any assurance. Accurate precipitation

forecasting has been an important issue in hydrological science as early notice of stern weather can help avoid natural disaster injuries and damage if prompt and accurate forecasts are made. The theory of the modular model and the integrati2on of different models has recently gained more interest in rainfall forecasting to address this challenge. A huge range of rainfall prediction methodologies is available in India. In India, there are two primary methods of forecasting rainfall. Regression, Artificial Neural Network (ANN), Decision Tree algorithm, Fuzzy logic and team process of data handling are the majority frequently used computational methods used for weather forecasting The basic goal is to follow information rules and relationships while gaining intangible and potentially expensive knowledge. Artificial NN is a promising part of this wide field

Rainfall prediction remains a serious concern and has attracted the attention of governments, industries, risk management entities, as well as the scientific community. Rainfall is a climatic factor that affects many human activities like agricultural production, construction, power generation, forestry and tourism, among others [1]. To this extent, rainfall prediction is essential since this variable is the one with the highest correlation with adverse natural events such as landslides, flooding, mass movements and avalanches. These incidents have affected society for years [2]. Therefore, having an appropriate approach for rainfall prediction makes it possible to take preventive and mitigation measures for these natural phenomena .

## **II.EXISTING SYSTEM:**

Rainfall prediction is important as heavy rainfall can lead to many disasters. The prediction helps people to take preventive measures and moreover the prediction should be accurate. There are two types of prediction short term rainfall prediction and long term rainfall. Prediction mostly short term prediction can gives us the accurate result. The main challenge is to build a model for long term rainfall prediction. Heavy precipitation prediction could be a major drawback for earth science department because it is closely associated with the economy and lifetime of human.

### III. PROPOSED SYSTEM:

It's a cause for natural disasters like flood and drought that square measure encountered by individuals across the world each year. Accuracy of rainfall statement has nice importance for countries like India whose economy is basically dependent on agriculture. The dynamic nature of atmosphere, applied mathematics techniques fail to provide sensible accuracy for precipitation statement. The prediction of precipitation using machine learning techniques may use regression. Intention of this project is to offer non-experts easy access to the techniques, approaches utilized in the sector of precipitation prediction and provide a comparative study among the various machine learning techniques.

The overall architecture include four major components: Data Exploration and Analysis, Data Pre-processing, Model Implementation, and Model Evaluation, as shown in Fig.



Over All Architecture

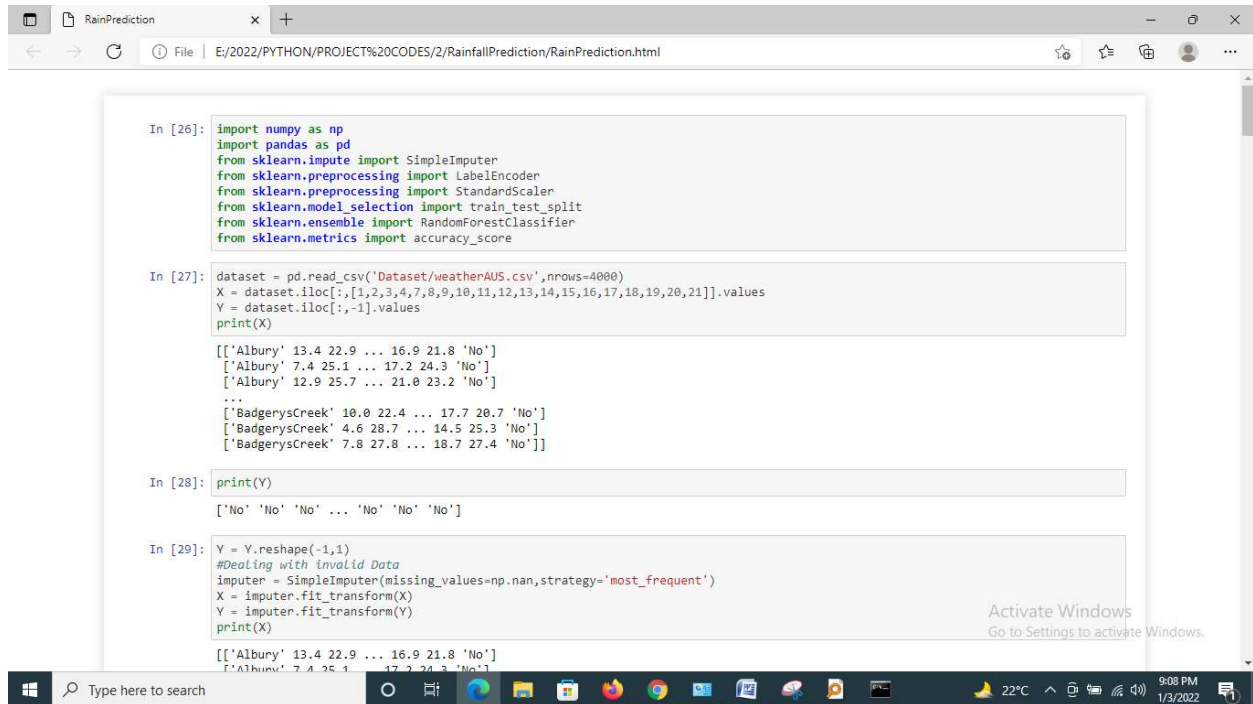
#### Data Exploration and Analysis

Exploratory Data Analysis is valuable to machine learning problems since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts [4]. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and researchers

We performed EDA using two methods - Univariate Visualization which provides summary statistics for each field in the raw data set (figure 2) and Pair-wise Correlation Matrix which is performed to understand interactions between different fields in the data set.

## IV.RESULTS:

## Packages



```

In [26]: import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

In [27]: dataset = pd.read_csv('Dataset/weatherAUS.csv', nrows=4000)
X = dataset.iloc[:, [1,2,3,4,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]].values
Y = dataset.iloc[:, -1].values
print(X)

[['Albury' 13.4 22.9 ... 16.9 21.8 'No']
['Albury' 7.4 25.1 ... 17.2 24.3 'No']
['Albury' 12.9 25.7 ... 21.0 23.2 'No']
...
['BadgerysCreek' 10.0 22.4 ... 17.7 20.7 'No']
['BadgerysCreek' 4.6 28.7 ... 14.5 25.3 'No']
['BadgerysCreek' 7.8 27.8 ... 18.7 27.4 'No']]

In [28]: print(Y)

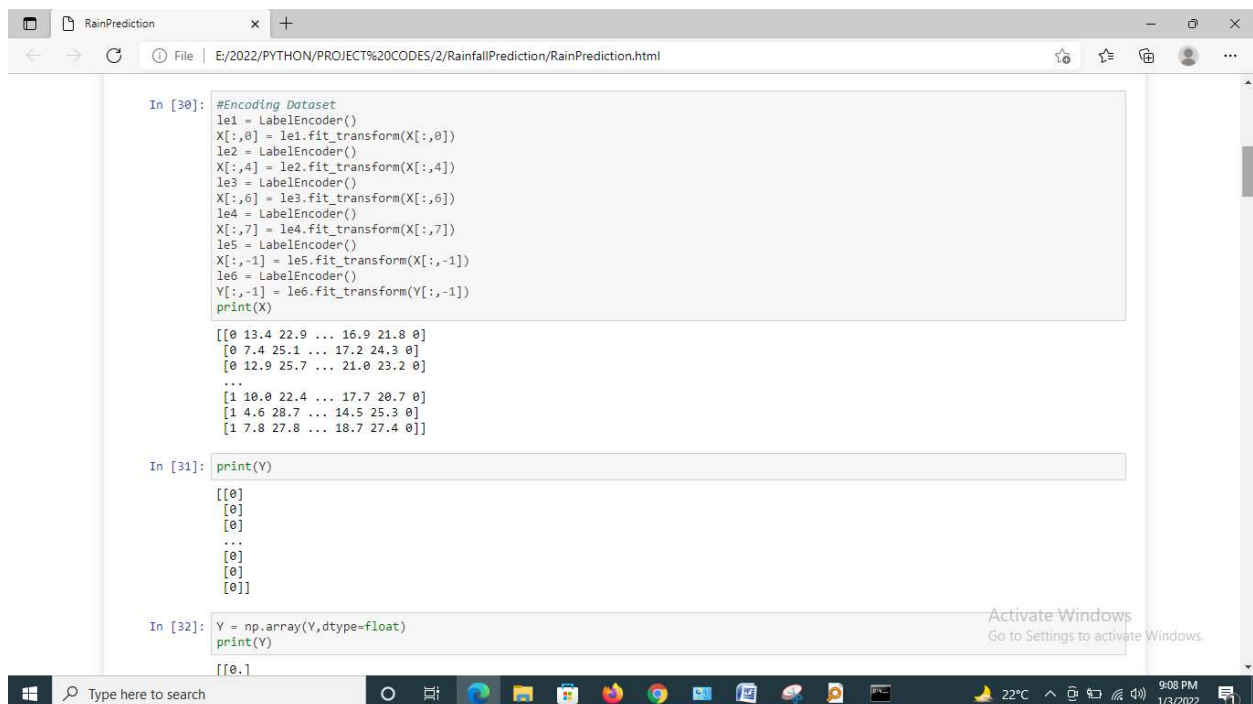
['No' 'No' 'No' ... 'No' 'No' 'No']

In [29]: Y = Y.reshape(-1,1)
#Dealing with invalid Data
imputer = SimpleImputer(missing_values=np.nan, strategy='most_frequent')
X = imputer.fit_transform(X)
Y = imputer.fit_transform(Y)
print(X)

[['Albury' 13.4 22.9 ... 16.9 21.8 'No']
['Albury' 7.4 25.1 ... 17.2 24.3 'No']
...
]]

```

## Analysis



```

In [30]: #Encoding Dataset
le1 = LabelEncoder()
X[:,0] = le1.fit_transform(X[:,0])
le2 = LabelEncoder()
X[:,4] = le2.fit_transform(X[:,4])
le3 = LabelEncoder()
X[:,6] = le3.fit_transform(X[:,6])
le4 = LabelEncoder()
X[:,7] = le4.fit_transform(X[:,7])
le5 = LabelEncoder()
X[:, -1] = le5.fit_transform(X[:, -1])
le6 = LabelEncoder()
Y[:, -1] = le6.fit_transform(Y[:, -1])
print(X)

[[0 13.4 22.9 ... 16.9 21.8 0]
[0 7.4 25.1 ... 17.2 24.3 0]
[0 12.9 25.7 ... 21.0 23.2 0]
...
[1 10.0 22.4 ... 17.7 20.7 0]
[1 4.6 28.7 ... 14.5 25.3 0]
[1 7.8 27.8 ... 18.7 27.4 0]]

In [31]: print(Y)

[[0]
[0]
[0]
...
[0]
[0]
[0]]

In [32]: Y = np.array(Y, dtype=float)
print(Y)

[[0.]

```

## Training

```

In [32]: Y = np.array(Y, dtype=float)
print(Y)

[[0.]
 [0.]
 [0.]
 ...
 [0.]
 [0.]
 [0.]]

In [33]: #Feature Scaling
sc = StandardScaler()
X = sc.fit_transform(X)
print(X)

[[-5.61951487e-01  5.87241679e-01  1.54909661e-03 ...  3.42299439e-01
  3.88963551e-02 -4.99609344e-01]
 [-5.61951487e-01 -4.05818061e-01  2.96615118e-01 ...  3.91129021e-01
  3.84764709e-01 -4.99609344e-01]
 [-5.61951487e-01  5.04487292e-01  3.77087669e-01 ...  1.00963706e+00
  2.32582633e-01 -4.99609344e-01]
 ...
 [ 1.77951304e+00  2.45118493e-02 -6.55113628e-02 ...  4.72511657e-01
 -1.13285721e-01 -4.99609344e-01]
 [ 1.77951304e+00 -8.69235526e-01  7.79450426e-01 ... -4.83372170e-02
  5.23112051e-01 -4.99609344e-01]
 [ 1.77951304e+00 -3.39607452e-01  6.58741599e-01 ...  6.35276931e-01
  8.13641468e-01 -4.99609344e-01]]

In [34]: #Splitting Dataset into Training set and Test set
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=0)
print(X_train)

[[-0.56195149  0.42173291  0.44414813 ...  0.30974638  0.45393838
  -0.49960934]
 [-0.56195149 -1.13404956 -1.1653029 ... -1.18769413 -1.04021291]

```

## Algorithms

```

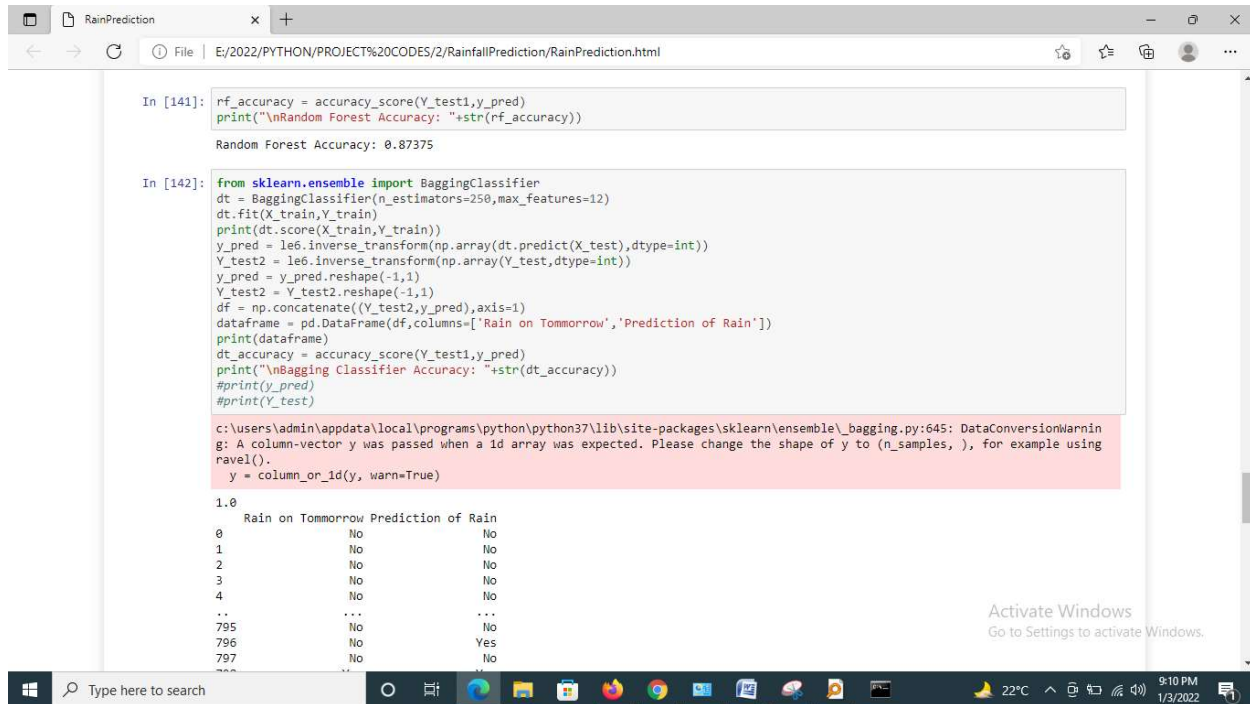
In [138]: #Training Model
classifier = RandomForestClassifier(n_estimators=100, random_state=0)
classifier.fit(X_train, Y_train)
print(classifier.score(X_train, Y_train))
y_pred = le6.inverse_transform(np.array(classifier.predict(X_test), dtype=int))
Y_test1 = le6.inverse_transform(np.array(Y_test, dtype=int))
print(y_pred)

c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\ipykernel_launcher.py:3: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using.ravel().
This is separate from the ipykernel package so we can avoid doing imports until

0.9996875
['No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No'
'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes'
'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'
'Yes' 'Yes' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No'
'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'
'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'No'

```

## Random Forest



```
In [141]: rf_accuracy = accuracy_score(Y_test1,y_pred)
print("\nRandom Forest Accuracy: "+str(rf_accuracy))

Random Forest Accuracy: 0.87375

In [142]: from sklearn.ensemble import BaggingClassifier
dt = BaggingClassifier(n_estimators=250,max_features=12)
dt.fit(X_train,Y_train)
print(dt.score(X_train,Y_train))
y_pred = le6.inverse_transform(np.array(dt.predict(X_test),dtype=int))
Y_test2 = le6.inverse_transform(np.array(Y_test,dtype=int))
y_pred = y_pred.reshape(-1,1)
Y_test2 = Y_test2.reshape(-1,1)
df = np.concatenate((Y_test2,y_pred),axis=1)
dataframe = pd.DataFrame(df,columns=['Rain on Tommorrow','Prediction of Rain'])
print(dataframe)
dt_accuracy = accuracy_score(Y_test1,y_pred)
print("\nBagging Classifier Accuracy: "+str(dt_accuracy))
#print(y_pred)
#print(Y_test)
```

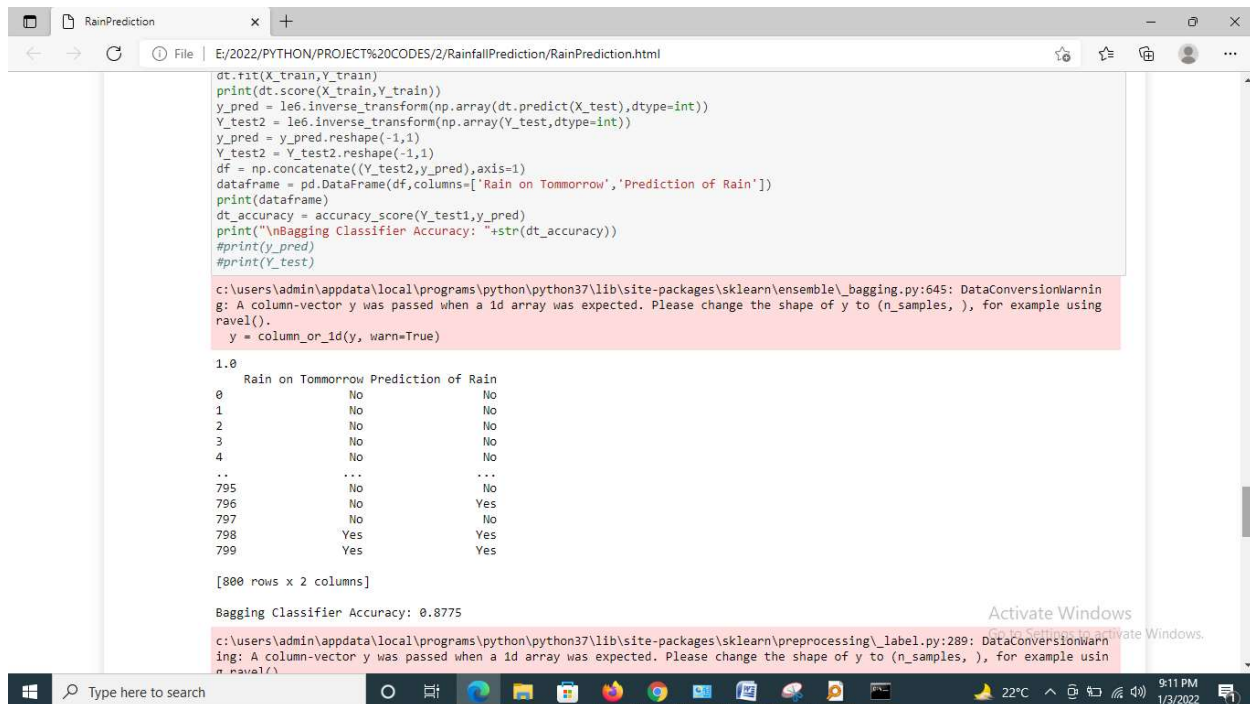
c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\sklearn\ensemble\\_bagging.py:645: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

	Rain on Tommorrow	Prediction of Rain
0	No	No
1	No	No
2	No	No
3	No	No
4	No	No
...	...	...
795	No	No
796	No	Yes
797	No	No
...	...	...

Activate Windows  
Go to Settings to activate Windows.

## Bagging Classifier



```
dt.fit(X_train,Y_train)
print(dt.score(X_train,Y_train))
y_pred = le6.inverse_transform(np.array(dt.predict(X_test),dtype=int))
Y_test2 = le6.inverse_transform(np.array(Y_test,dtype=int))
y_pred = y_pred.reshape(-1,1)
Y_test2 = Y_test2.reshape(-1,1)
df = np.concatenate((Y_test2,y_pred),axis=1)
dataframe = pd.DataFrame(df,columns=['Rain on Tommorrow','Prediction of Rain'])
print(dataframe)
dt_accuracy = accuracy_score(Y_test1,y_pred)
print("\nBagging Classifier Accuracy: "+str(dt_accuracy))
#print(y_pred)
#print(Y_test)
```

c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\sklearn\ensemble\\_bagging.py:645: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

	Rain on Tommorrow	Prediction of Rain
0	No	No
1	No	No
2	No	No
3	No	No
4	No	No
...	...	...
795	No	No
796	No	Yes
797	No	No
798	Yes	Yes
799	Yes	Yes

[800 rows x 2 columns]

Bagging Classifier Accuracy: 0.8775

c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\sklearn\preprocessing\\_label.py:289: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

Activate Windows  
Go to Settings to activate Windows.

## Gradient Boosting

```
In [143]: from sklearn.ensemble import GradientBoostingClassifier
dt = GradientBoostingClassifier(n_estimators=170,max_depth=1)
dt.fit(X_train,Y_train)
print(dt.score(X_train,Y_train))
y_pred = le6.inverse_transform(np.array(dt.predict(X_test),dtype=int))
Y_test3 = le6.inverse_transform(np.array(Y_test,dtype=int))
y_pred = y_pred.reshape(-1,1)
Y_test3 = Y_test3.reshape(-1,1)
df = np.concatenate((Y_test3,y_pred),axis=1)
dataframe = pd.DataFrame(df,columns=['Rain on Tomorrow','Prediction of Rain'])
print(dataframe)
dt_accuracy = accuracy_score(Y_test1,y_pred)
print("\nGradient Boosting Accuracy: "+str(dt_accuracy))
#print(y_pred)
#print(Y_test)
```

c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\sklearn\ensemble\\_gb.py:1454: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

```
0.878125
   Rain on Tomorrow Prediction of Rain
0                No                No
1                No                No
2                No                No
3                No                No
4                No                No
..              ...                ...
795              No                No
796              No                Yes
797              No                No
798              Yes                Yes
799              Yes                Yes
```

[800 rows x 2 columns]

Gradient Boosting Accuracy: 0.88

## Xgboost

```
Y_test4 = le6.inverse_transform(np.array(Y_test,dtype=int))
#print(y_pred)
#print(Y_test)
y_pred = y_pred.reshape(-1,1)
Y_test4 = Y_test4.reshape(-1,1)
df = np.concatenate((Y_test4,y_pred),axis=1)
dataframe = pd.DataFrame(df,columns=['Rain on Tomorrow','Prediction of Rain'])
print(dataframe)
dt_accuracy = accuracy_score(Y_test1,y_pred)
print("\nXGBoost Accuracy: "+str(dt_accuracy))
```

c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\sklearn\preprocessing\\_label.py:235: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

c:\users\admin\appdata\local\programs\python\python37\lib\site-packages\sklearn\preprocessing\\_label.py:268: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n\_samples, ), for example using ravel().

```
y = column_or_1d(y, warn=True)
```

```
1.0
   Rain on Tomorrow Prediction of Rain
0                No                No
1                No                No
2                No                No
3                No                No
4                No                No
..              ...                ...
795              No                No
796              No                No
797              No                No
798              Yes                Yes
799              Yes                No
```

[800 rows x 2 columns]

XGBoost Accuracy: 0.875

## V.CONCLUSION AND FUTURE WORK

In this work, we investigated and used a number of preprocessing techniques and discovered how they affected our classifiers' overall performance. In order to see how the input data can impact the model predictions, we also conducted a comparative analysis of all the classifiers using various input data. We can draw the conclusion that Australian weather is unpredictable and that rainfall and the corresponding time and place do not correlate. We discovered several correlations and trends in the data that aided in identifying key characteristics. See the part in the appendix. We may use Deep Learning models like the Multilayer Perceptron and Convolutional Neural Network, among others, because we have a vast amount of data. A comparative analysis of deep learning models and machine learning classifiers would be fantastic.

## REFERENCES

1. World Health Organization: Climate Change and Human Health: Risks and Responses. World Health Organization, January 2003
2. Alcantara-Ayala, I.: Geomorphology, natural hazards, vulnerability and prevention of natural disasters in developing countries. *Geomorphology* 47(24), 107124 (2002)
3. Nicholls, N.: Atmospheric and climatic hazards: Improved monitoring and prediction for disaster mitigation. *Natural Hazards* 23(23), 137155 (2001)
4. [Online] InDataLabs, Exploratory Data Analysis: the Best way to Start a Data Science Project. Available: <https://medium.com/@InDataLabs/why-start-a-data-science-project-with-exploratory-data-analysis-f90c0efcbe49>
5. [Online] Pandas Documentation. Available: [https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get\\_dummies.html](https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.get_dummies.html)
6. [Online] Sckit-Learn Documentation Available: [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_extraction.FeatureHasher.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.FeatureHasher.html)
7. [Online] Sckit-Learn Documentation Available: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
8. [Online] Sckit Learn Documentation Available: [https://scikit-learn.org/stable/modules/generated/sklearn.feature\\_selection.SelectKBest.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html)
9. [Online] Raheel Shaikh, Feature Selection Techniques in Machine Learning with Python Available: <https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e>
10. [Online] Imbalanced-learn Documentation Available: <https://imbalanced-learn.readthedocs.io/en/stable/introduction.html>



11. V. Veeralakshmi and D. Ramyachitra, Ripple Down Rule learner (RIDOR) Classifier for IRIS Dataset. Issues, vol 1, p. 79-85.

12. [Online] Aditya Mishra, Metrics to Evaluate your Machine Learning Algorithm Available: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>