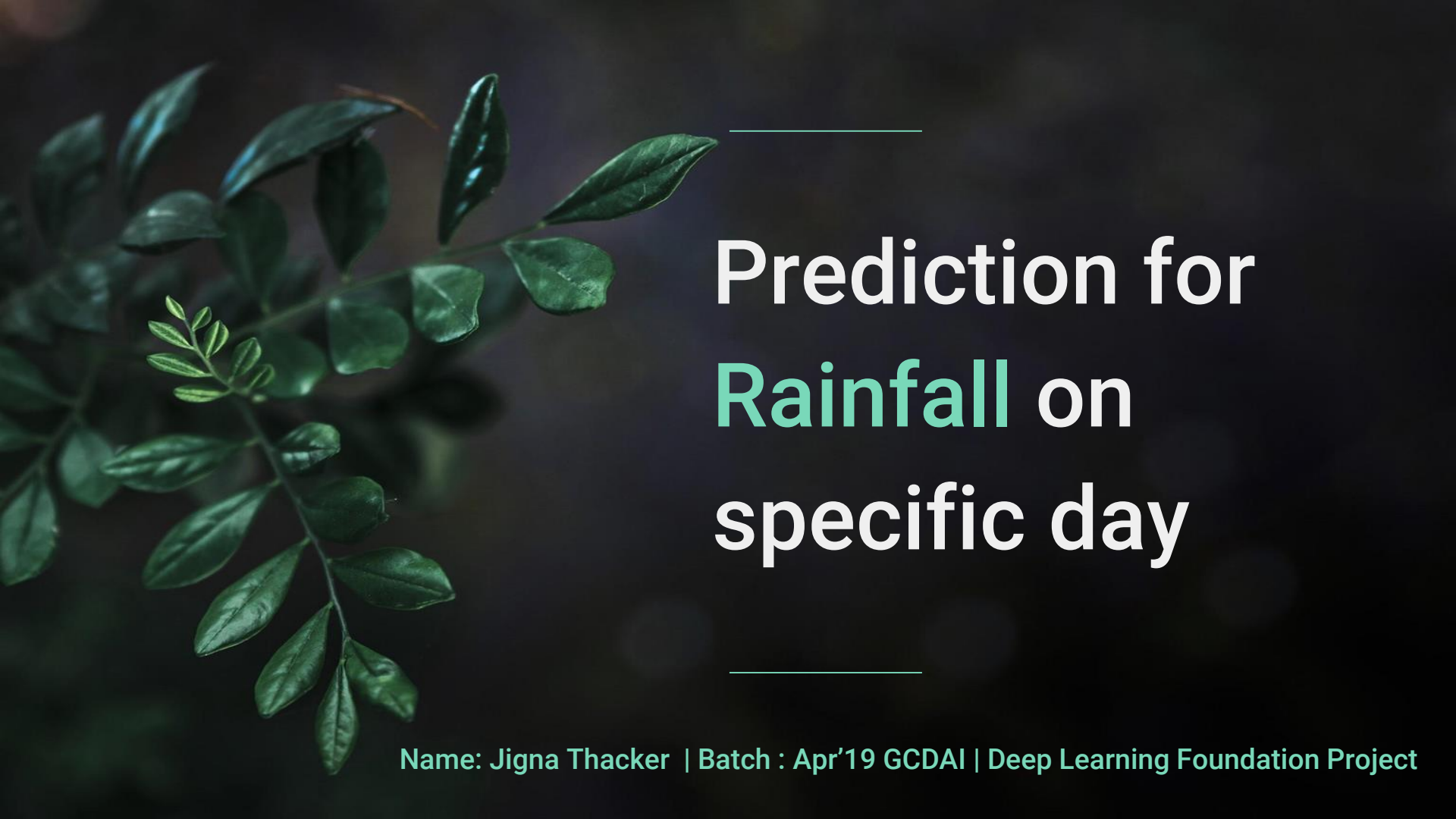


Prediction for Rainfall on specific day

Name: Jigna Thacker | Batch : Apr'19 GCDAI | Deep Learning Foundation Project

Agenda

1

Introduction

Project Objective

Background about the Seattle

2

Database Handling

About the data

Exploratory Data Analysis (EDA)

ANN Model Building

3

Conclusion

Summary & Conclusion

Introduction

Project objective
Background about the Seattle

Project Objective

" Build a model which is able to predict whether it will rain on specific day using **binary classification** approach "

Background !

Seattle is a seaport city on the West Coast of the **United States**. City is famous for coffee, grunge and technology companies, one of the things that **Seattle** is most famous for is how often it **Rains**…

Seattle Weather

“The sound of rains
needs no translation.”

—Someone famous

Database Handling

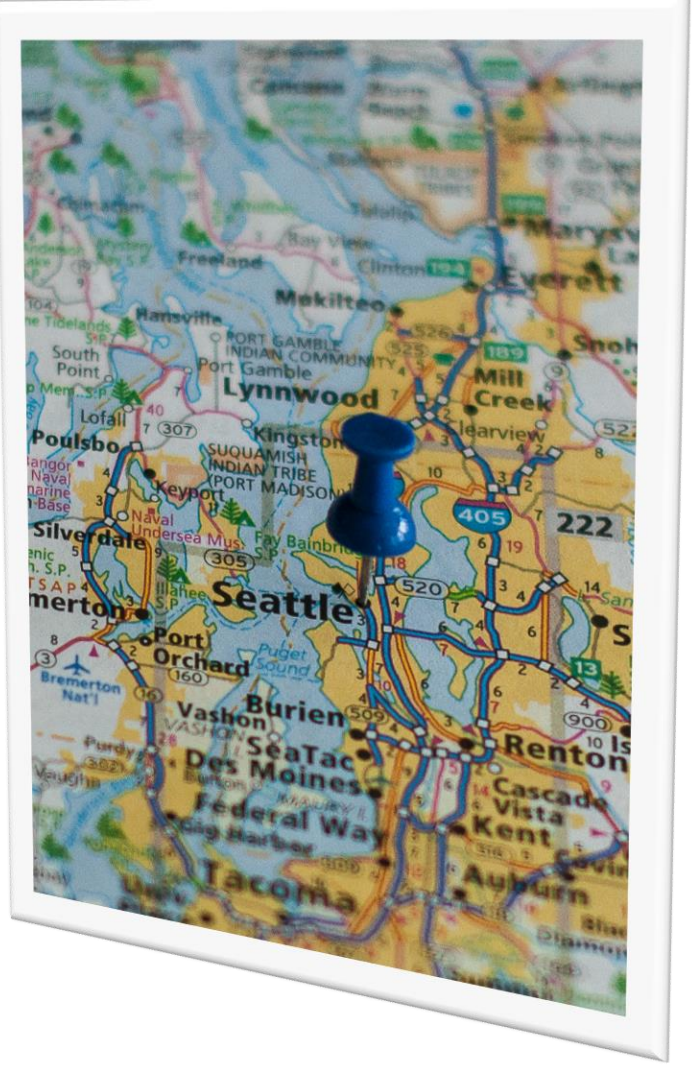
About the data
Exploratory Data Analysis (EDA)
ANN Model Building

About the Database

Data Description

Variable Name	Description
Date	The date of observation
PRCP	The amount of precipitation, in inches
TMAX	The maximum temperature for that day, in degrees Fahrenheit
TMIN	The minimum temperature for that day, in degrees Fahrenheit
RAIN [target]	**TRUE if rain was observed on that day, FALSE if it was not.**

The dataset contains complete records of daily rainfall patterns from January 1st, 1948 to December 12, 2017

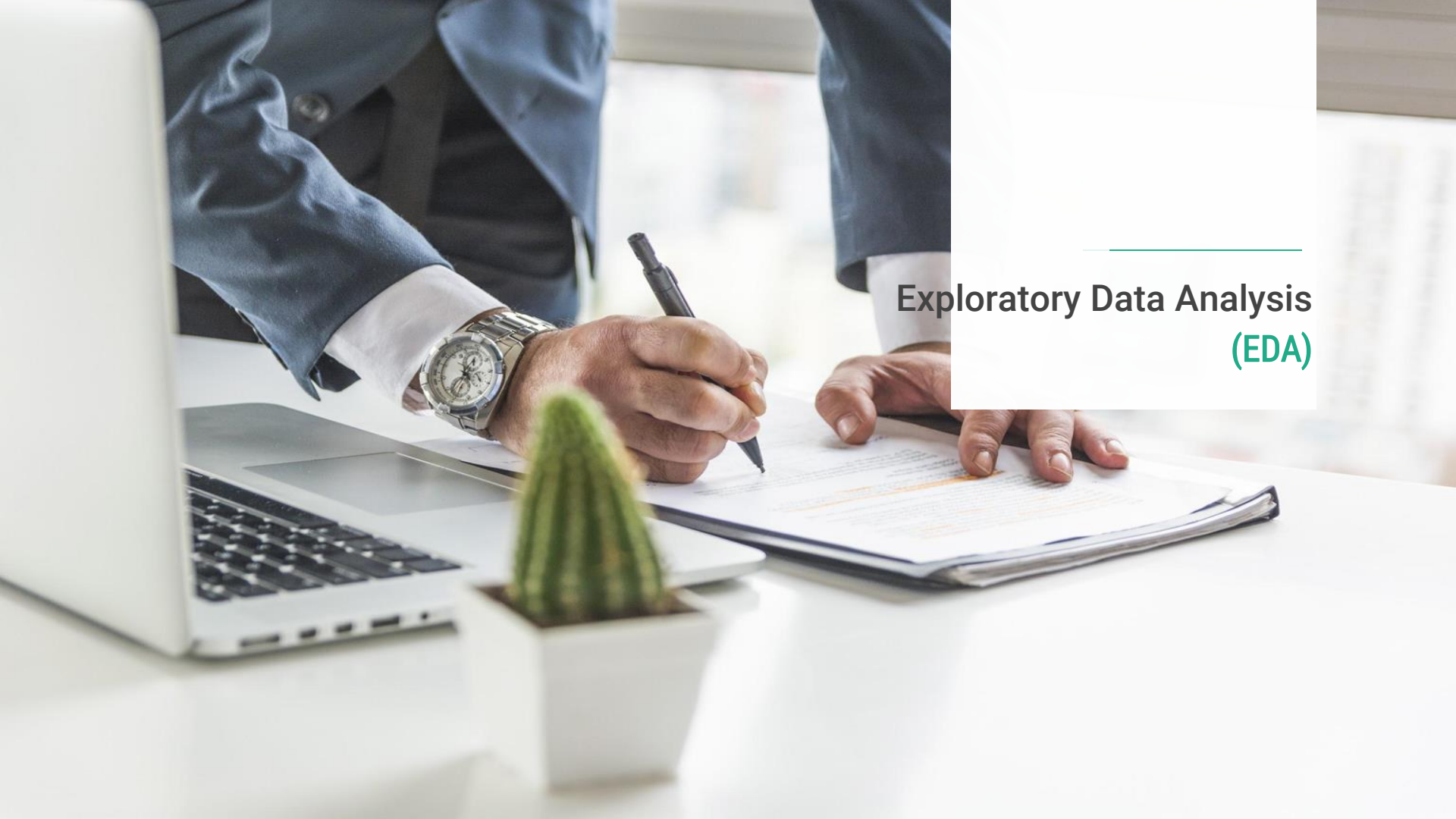


25,551

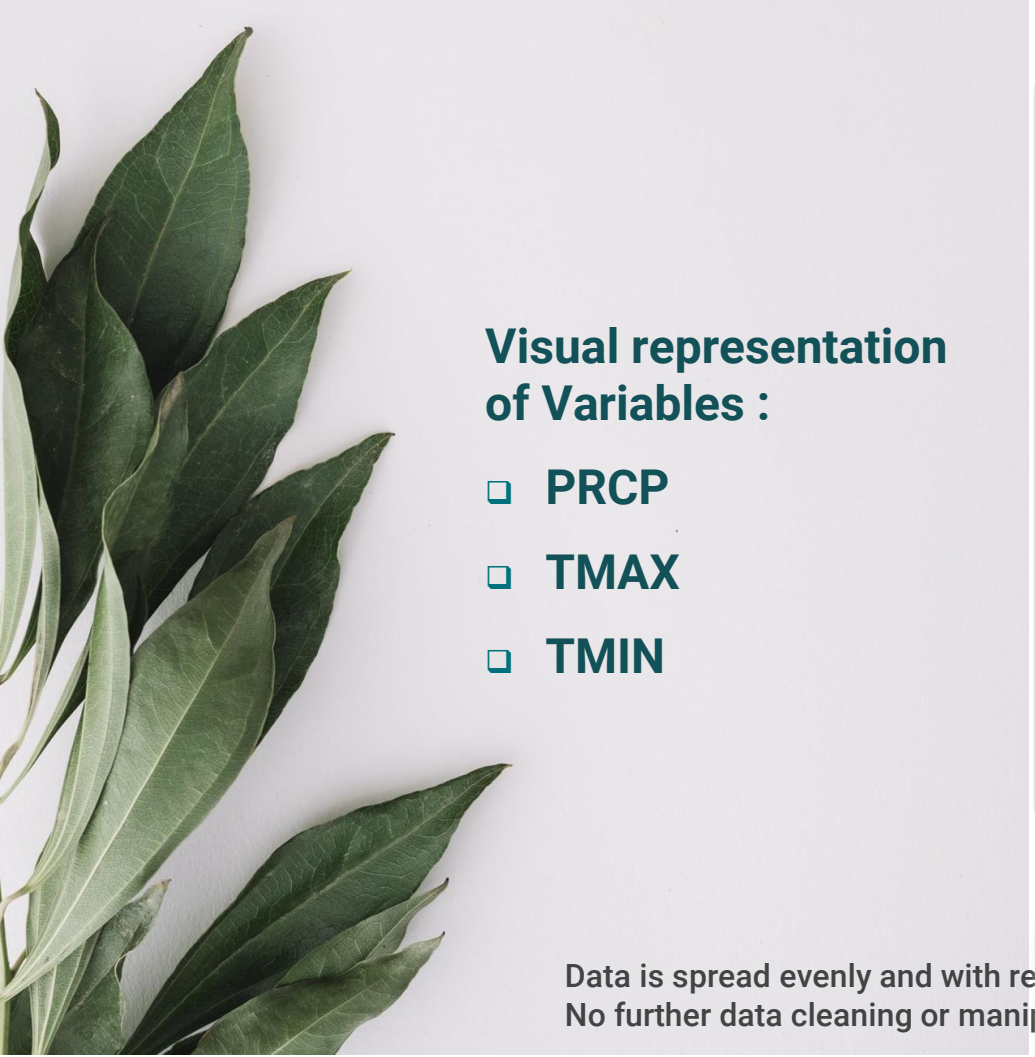
Approximately data sample

25,548

Final sample : Post eliminating null values

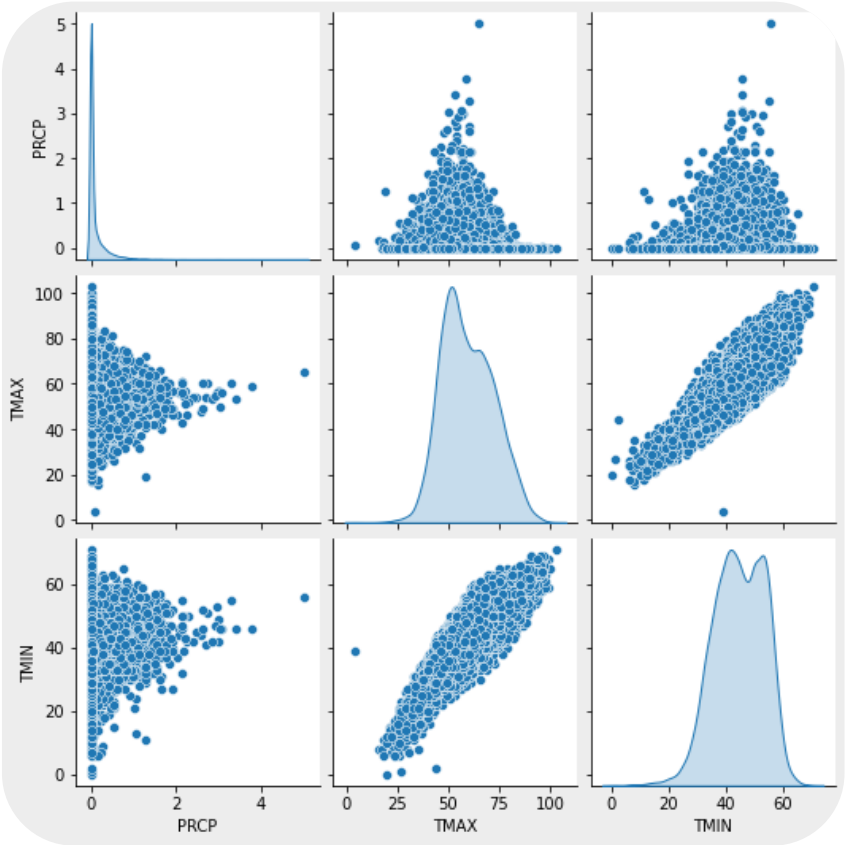


Exploratory Data Analysis (EDA)



Visual representation of Variables :

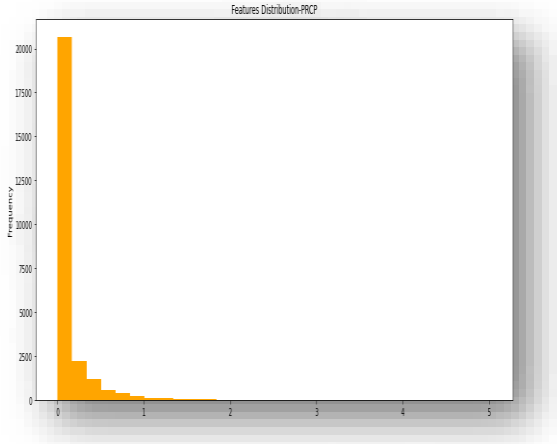
- PRCP
- TMAX
- TMIN



Data is spread evenly and with respect to each variables PRCP, TMAX and TMIN.
No further data cleaning or manipulation is required.

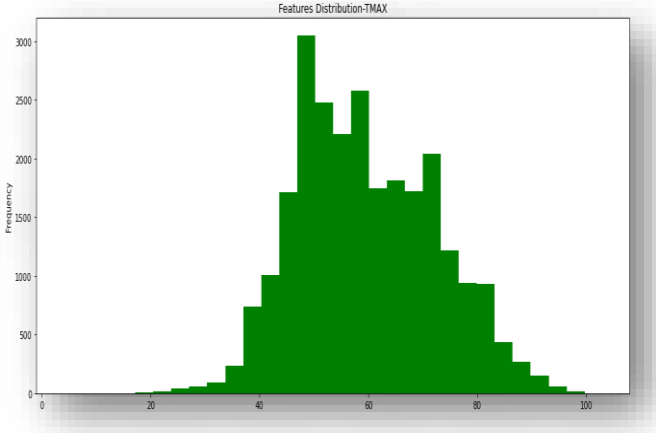
Let’s see feature distribution, for all three variables independently

PRCP



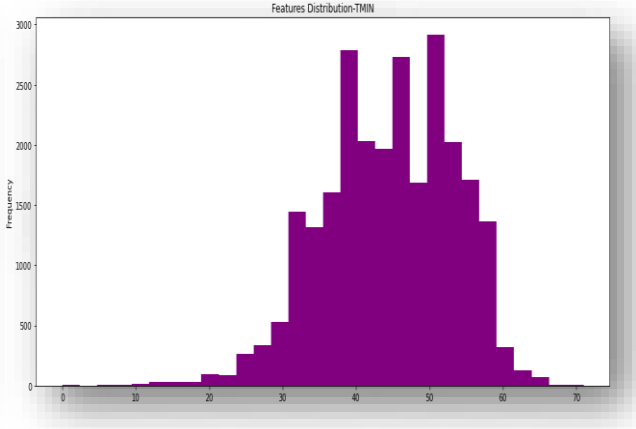
Distribution is between 0-5 with majority data is <1

TMAX



Distribution is normally distributed with few data points spread over right side of the chart

TMIN

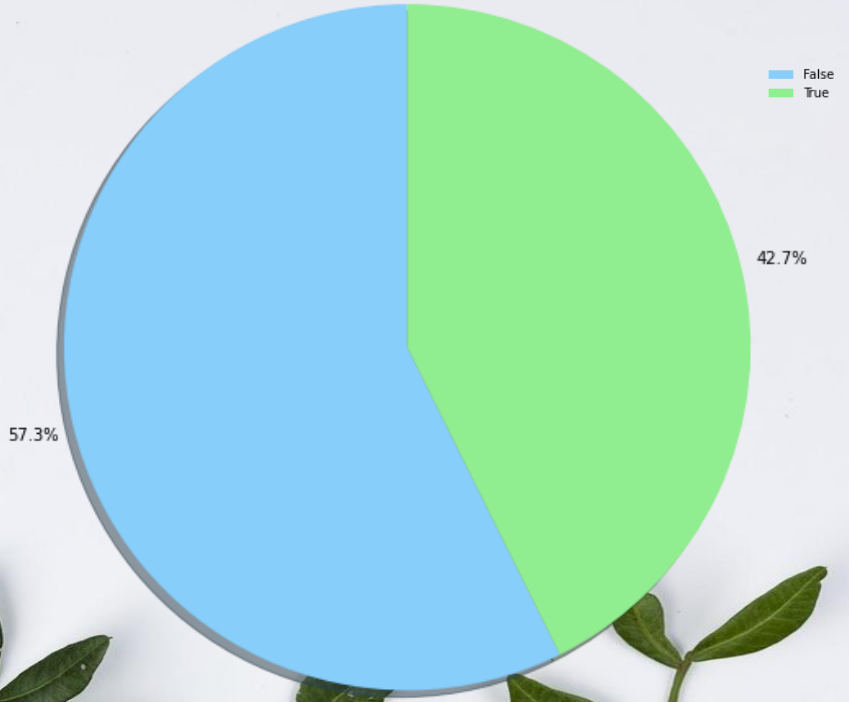


Distribution is normally distributed with few data points spread over letf side of the chart

RAIN Observation

We can see from the chart 42.7% rains observed against 57.3 not observed.

This looks balance data and hence no data smoothening technique is required.



ANN
Model Building

What is “ANN” ?

Artificial neural networks (ANNs) are biologically inspired computer programs designed to simulate the way in which the human brain processes information.

ANNs gather their knowledge by detecting the patterns and relationships in data and learn (or are trained) through experience, not from programming.

The various applications of ANNs can be summarized into classification or pattern recognition, prediction and modeling.

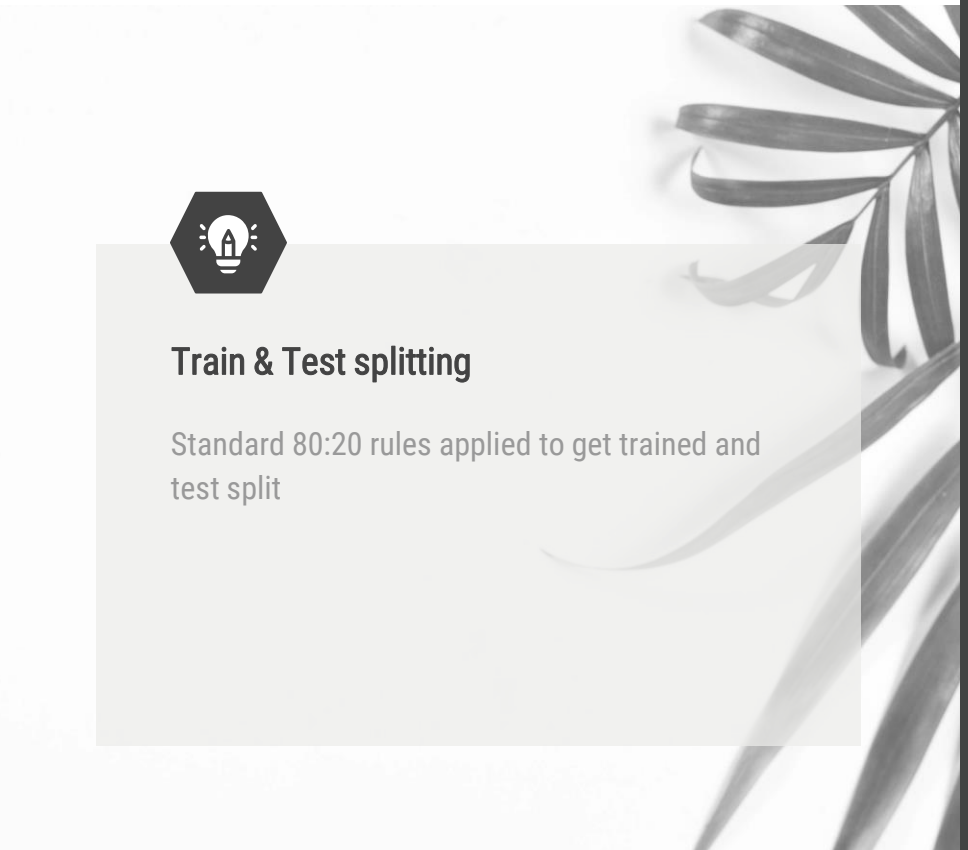
Preparing “ANN” using Sequential Model Building!

```
model  = Sequential()
model.add(Dense(units= 16,kernel_initializer= 'uniform’,
          activation = 'relu', input_dim=3))
model.add(Dense(units= 16,kernel_initializer= 'uniform’,
          activation = 'relu'))
model.add(Dense(units= 1,kernel_initializer= 'uniform’,
          activation = 'sigmoid'))
model.compile(optimizer='adam',loss='binary_crossentropy’,
              metrics=['accuracy'])
```

Model Summary!

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 16)                64
_________________________________________________________________
dense_1 (Dense)              (None, 16)                272
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 17
=================================================================
Total params: 353
Trainable params: 353
Non-trainable params: 0
_________________________________________________________________
```

Model Fitting



Train & Test splitting

Standard 80:20 rules applied to get trained and test split

ANN Model features

After various iterations; below are the final **selection** for model fitting:

- Units : 16,16,1
- Kernel_initializer: uniform
- Activation : relu
- Final activation : sigmoid
- Input_dim : 3
- Optimizer : adam
- Loss function : binary_crossentropy
- Metrics : accuracy
- Epochs : 20
- Batch size : 100

… Confusion matrix

5110 test sample

Actual vs Predicted		Actual	
		Rain=False	Rain=True
Predicted	Rain=False	2952 **TP**	1
	Rain=True	215	1942 **TN**

TP (Rain=False) : 93.21%
TN (Rain=True) : 99.94%

Over all model evaluation:
Test accuracy : 95.77%

Conclusion

Summary & Conclusion
Note on worth thinking

Summary & Conclusion!

Data is relatively very clean and overall sample is small for ANN Model building. This could be one of the reason model is giving very good accuracy.

Model is executed very well. While executing model it is giving 95.77% accuracy

Point to note
sample size - 25548 is generally very low sample to perform ANN

While applying same model with normalisation and standard scaler - accuracy is reaching 99.99%, technically overfitted model

Machine Learning Model Building using Pycaret:-
Most of the algorithm is giving very high accuracy > 90%

Something worth thinking ☺

When things are pretty manageable with normal ML algorithm why to go with ANN?

Thank You

	For further information please contact:
	Jigna Thacker
	jignazt@yahoo.com
	https://github.com/jmps967/INSAID-DLF_Jigna-Thacker