

1.Import necessary library

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

from sklearn import preprocessing

from sklearn import model_selection

from sklearn import metrics

from sklearn import linear_model

from sklearn import ensemble

from sklearn import tree

from sklearn import svm

import xgboost

from google.colab import drive

drive.mount('/content/drive')

Mounted at /content/drive

df=pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Rainfall weather.csv")
```

2.analyse the data

```
df.head()
```

Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine					
	WindGustDir	WindGustSpeed	WindDir9am	...	Humidity9am	Humidity3pm					
	Pressure9am	Pressure3pm	Cloud9am	Cloud3pm	Temp9am	Temp3pm					
	RainToday	RainTomorrow									
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	...
	71.0	22.0	1007.7	1007.1	8.0	NaN	16.9	21.8	No	No	

1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	...
	44.0	25.0	1010.6	1007.8	NaN	NaN	17.2	24.3	No	No	
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	...
	38.0	30.0	1007.6	1008.7	NaN	2.0	21.0	23.2	No	No	
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	...
	45.0	16.0	1017.6	1012.8	NaN	NaN	18.1	26.5	No	No	
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	...
	82.0	33.0	1010.8	1006.0	7.0	8.0	17.8	29.7	No	No	

5 rows × 23 columns

3.Handling missing values

```
df.isnull().sum()*100/len(df)
```

Date 0.000000

Location 0.000000

MinTemp 1.020899

MaxTemp 0.866905

Rainfall 2.241853

Evaporation 43.166506

Sunshine 48.009762

WindGustDir 7.098859

WindGustSpeed 7.055548

WindDir9am 7.263853

WindDir3pm 2.906641

WindSpeed9am 1.214767

WindSpeed3pm 2.105046

Humidity9am 1.824557

Humidity3pm	3.098446
Pressure9am	10.356799
Pressure3pm	10.331363
Cloud9am	38.421559
Cloud3pm	40.807095
Temp9am	1.214767
Temp3pm	2.481094
RainToday	2.241853
RainTomorrow	2.245978
dtype: float64	
df.isnull().sum()	
Date	0
Location	0
MinTemp	1485
MaxTemp	1261
Rainfall	3261
Evaporation	62790
Sunshine	69835
WindGustDir	10326
WindGustSpeed	10263
WindDir9am	10566
WindDir3pm	4228
WindSpeed9am	1767
WindSpeed3pm	3062
Humidity9am	2654

```
df['WindSpeed3pm'].fillna(df['WindSpeed3pm'].mean(), inplace=True)
```

```

df['Humidity9am'].fillna(df['Humidity9am'].mean(), inplace=True)
df['Humidity3pm'].fillna(df['Humidity3pm'].mean(), inplace=True)
df['Pressure9am'].fillna(df['Pressure9am'].mean(), inplace=True)
df['Pressure3pm'].fillna(df['Pressure3pm'].mean(), inplace=True)
df['Temp9am'].fillna(df['Temp9am'].mean(), inplace=True)
df['Temp3pm'].fillna(df['Temp3pm'].mean(), inplace=True)

#Loading the names of categorical columns
cat_names=df_cat.columns

# initializing the simple imputer for missing categorical values
import numpy as np
from sklearn.impute import SimpleImputer

imp_mode= SimpleImputer(missing_values=np.nan, strategy='most_frequent')

# fitting and transforming the missing data
df_cat=imp_mode.fit_transform(df_cat)

# converting array to dataframe
df_cat= pd.DataFrame(df_cat, columns=cat_names)

# concatenating the categorical and numeric
df = pd.concat([df, df_cat], axis=1)

```

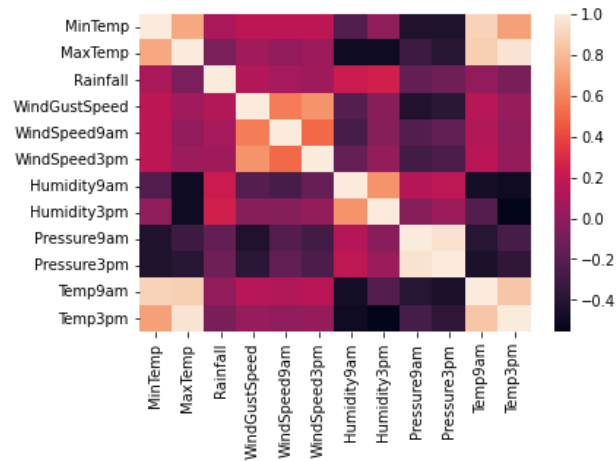
4.Data visulization

```
df.corr()
```

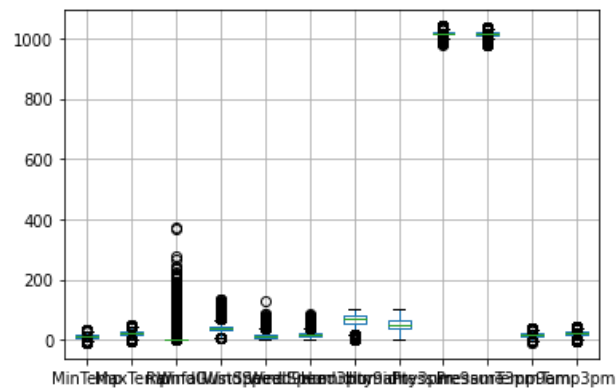
MinTemp	MaxTemp	Rainfall	WindGustSpeed	WindSpeed9am	WindSpeed3pm	
	Humidity9am	Humidity3pm	Pressure9am	Pressure3pm	Temp9am	Temp3pm
MinTemp	1.000000	0.733400	0.102706	0.172553	0.173404	
	0.173058	-0.230970	0.005995	-0.423584	-0.433147	0.897692
	0.699211					

MaxTemp	0.733400	1.000000	-0.074040	0.065895	0.014294	
	0.049717	-0.497927	-0.498760	-0.308309	-0.396622	0.879170
	0.968713					
Rainfall	0.102706	-0.074040	1.000000	0.126446	0.085925	0.056527
	0.221380	0.248905	-0.159055	-0.119541	0.011069	-0.077684
WindGustSpeed	0.172553	0.065895	0.126446	1.000000		0.577319
	0.657243	-0.207964	-0.025355	-0.425760	-0.383938	0.145904
	0.031884					
WindSpeed9am	0.173404	0.014294	0.085925	0.577319	1.000000	
	0.512427	-0.268271	-0.030887	-0.215339	-0.165388	0.127592
	0.004476					
WindSpeed3pm	0.173058	0.049717	0.056527	0.657243	0.512427	
	1.000000	-0.143458	0.016275	-0.277604	-0.239659	0.161060
	0.027587					
Humidity9am	-0.230970	-0.497927	0.221380	-0.207964	-0.268271	-
	0.143458	1.000000	0.659072	0.131503	0.176009	-0.469641
	0.490709					-
Humidity3pm	0.005995	-0.498760	0.248905	-0.025355	-0.030887	
	0.016275	0.659072	1.000000	-0.025848	0.048695	-0.216964
	-0.555608					
Pressure9am	-0.423584	-0.308309	-0.159055	-0.425760	-0.215339	-
	0.277604	0.131503	-0.025848	1.000000	0.959662	-0.397131
	0.265532					-
Pressure3pm	-0.433147	-0.396622	-0.119541	-0.383938	-0.165388	-
	0.239659	0.176009	0.048695	0.959662	1.000000	-0.441459
	0.360707					-
Temp9am	0.897692	0.879170	0.011069	0.145904	0.127592	
	0.161060	-0.469641	-0.216964	-0.397131	-0.441459	1.000000
	0.846141					
Temp3pm	0.699211	0.968713	-0.077684	0.031884	0.004476	
	0.027587	-0.490709	-0.555608	-0.265532	-0.360707	0.846141
	1.000000					

```
cor = df.corr()sns.heatmap(data=cor,xtickla
```



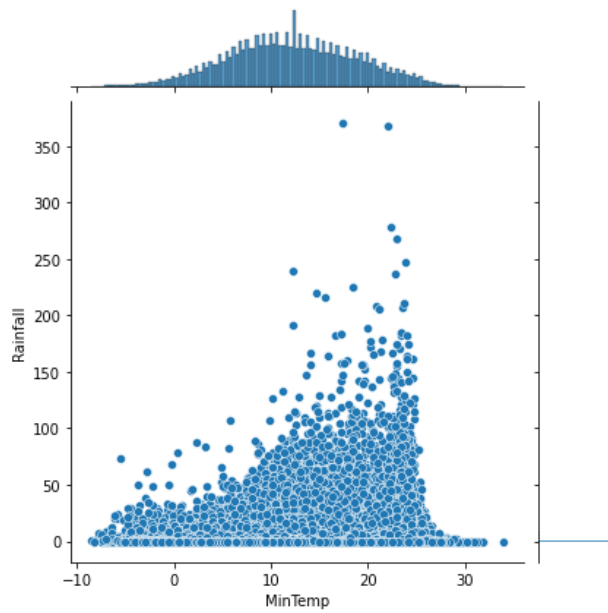
`df.boxplot()`



`sns.jointplot(df["MinTemp"],df["Rainfall"])`

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

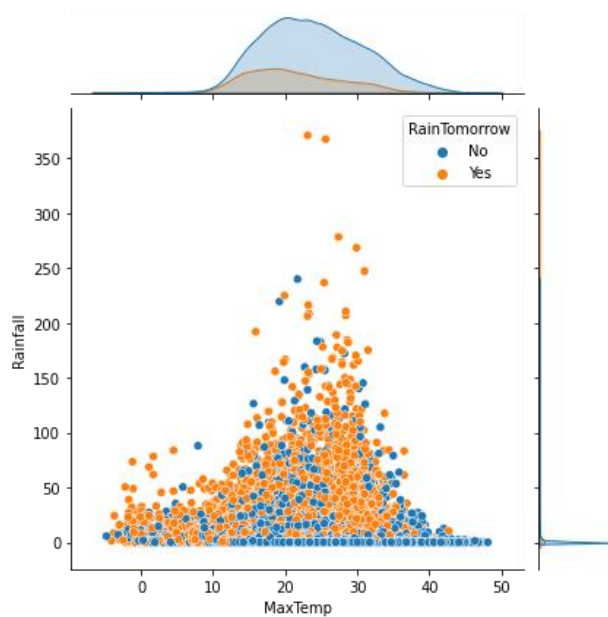
FutureWarning

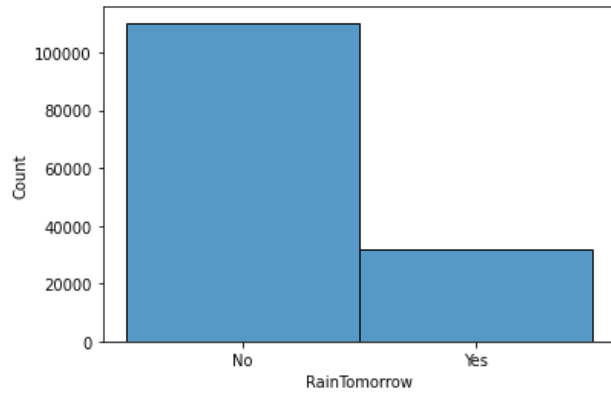


```
sns.jointplot(df["MaxTemp"],df['Rainfall'],hue=df['RainTomorrow'])
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning

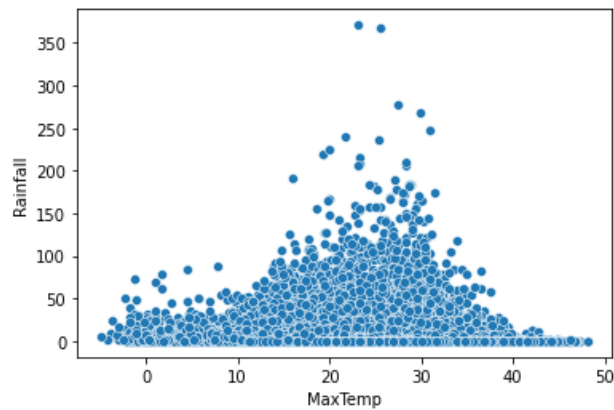




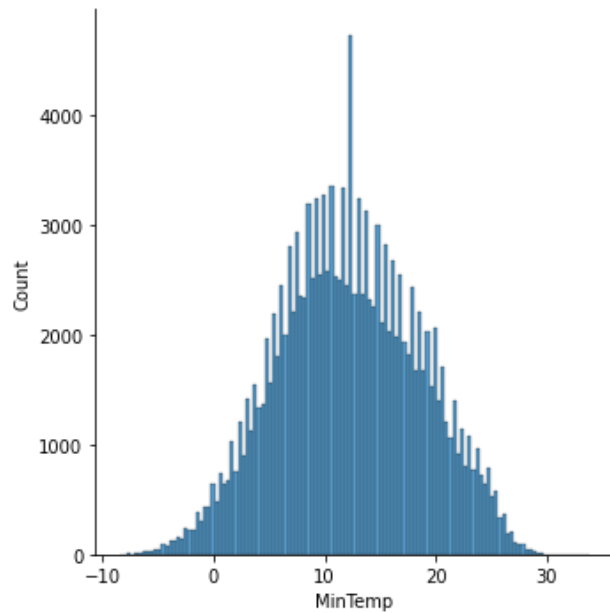
```
sns.scatterplot(df['MaxTemp'],df['Rainfall'])
```

/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

FutureWarning



```
sns.displot(df['MinTemp'])
```



5.Splitting the dataset into Dependent and independent variable

```
y=df['RainTomorrow']
x=df.drop('RainTomorrow',axis=1)
```

6.Feature Scaling

```
from sklearn.preprocessing import StandardScaler

y=df['RainTomorrow']
x=df.drop('RainTomorrow',axis=1)

names=x.columns

names

Index(['Date', 'Location', 'MinTemp', 'MaxTemp', 'Rainfall', 'WindGustSpeed',
      'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm',
      'Pressure9am', 'Pressure3pm', 'Temp9am', 'Temp3pm', 'RainToday',
      'WindGustDir', 'WindDir9am', 'WindDir3pm'],
      dtype='object')
```

```
sc=StandardScaler()
```

7.splitting the data into train and test

```
sklearn import model_selection
```

```
x_train,x_test,y_train,y_test
```