{

"cells": [

{

"cell\_type": "markdown",

"id": "injured-empire",

"metadata": {},

"source": [

"# Importing the libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"id": "understanding-extreme",

"metadata": {},

"outputs": [],

"source": [

"import pandas as pd\n",

"import numpy as np\n",

"import seaborn as sns \n",

"import matplotlib.pyplot as plt\n",

"from sklearn import preprocessing\n",

"from sklearn import model\_selection\n",

"from sklearn import metrics \n",

"from sklearn import linear\_model\n",

"from sklearn import ensemble\n",

"from sklearn import tree\n",

"from sklearn import svm\n",

"import xgboost"

]

},

{

"cell\_type": "markdown",

"id": "focused-graphic",

"metadata": {},

"source": [

"# Importing the dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "jewish-kruger",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>...</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" <th>RainTomorrow</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2008-12-01</td>\n",

" <td>Albury</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>W</td>\n",

" <td>44.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>8.0</td>\n",

" <td>NaN</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2008-12-02</td>\n",

" <td>Albury</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>WNW</td>\n",

" <td>44.0</td>\n",

" <td>NNW</td>\n",

" <td>...</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2008-12-03</td>\n",

" <td>Albury</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>WSW</td>\n",

" <td>46.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>NaN</td>\n",

" <td>2.0</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2008-12-04</td>\n",

" <td>Albury</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NE</td>\n",

" <td>24.0</td>\n",

" <td>SE</td>\n",

" <td>...</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2008-12-05</td>\n",

" <td>Albury</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>W</td>\n",

" <td>41.0</td>\n",

" <td>ENE</td>\n",

" <td>...</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>7.0</td>\n",

" <td>8.0</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>"

],

"text/plain": [

" Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine \\\n",

"0 2008-12-01 Albury 13.4 22.9 0.6 NaN NaN \n",

"1 2008-12-02 Albury 7.4 25.1 0.0 NaN NaN \n",

"2 2008-12-03 Albury 12.9 25.7 0.0 NaN NaN \n",

"3 2008-12-04 Albury 9.2 28.0 0.0 NaN NaN \n",

"4 2008-12-05 Albury 17.5 32.3 1.0 NaN NaN \n",

"\n",

" WindGustDir WindGustSpeed WindDir9am ... Humidity9am Humidity3pm \\\n",

"0 W 44.0 W ... 71.0 22.0 \n",

"1 WNW 44.0 NNW ... 44.0 25.0 \n",

"2 WSW 46.0 W ... 38.0 30.0 \n",

"3 NE 24.0 SE ... 45.0 16.0 \n",

"4 W 41.0 ENE ... 82.0 33.0 \n",

"\n",

" Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \\\n",

"0 1007.7 1007.1 8.0 NaN 16.9 21.8 No \n",

"1 1010.6 1007.8 NaN NaN 17.2 24.3 No \n",

"2 1007.6 1008.7 NaN 2.0 21.0 23.2 No \n",

"3 1017.6 1012.8 NaN NaN 18.1 26.5 No \n",

"4 1010.8 1006.0 7.0 8.0 17.8 29.7 No \n",

"\n",

" RainTomorrow \n",

"0 No \n",

"1 No \n",

"2 No \n",

"3 No \n",

"4 No \n",

"\n",

"[5 rows x 23 columns]"

]

},

"execution\_count": 2,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"\n",

"import os, types\n",

"import pandas as pd\n",

"from botocore.client import Config\n",

"import ibm\_boto3\n",

"\n",

"def \_\_iter\_\_(self): return 0\n",

"\n",

"# @hidden\_cell\n",

"# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n",

"# You might want to remove those credentials before you share the notebook.\n",

"cos\_client = ibm\_boto3.client(service\_name='s3',\n",

" ibm\_api\_key\_id='vYxSqpgXYTE\_5CsKrkuRv\_pcMjpYBNBmaEy4lu9rgw03',\n",

" ibm\_auth\_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n",

" config=Config(signature\_version='oauth'),\n",

" endpoint\_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')\n",

"\n",

"bucket = 'exploratoryanalysisofrainfalldata-donotdelete-pr-nsxjcdoqodn5wl'\n",

"object\_key = 'weatherAUS.csv'\n",

"\n",

"body = cos\_client.get\_object(Bucket=bucket,Key=object\_key)['Body']\n",

"# add missing \_\_iter\_\_ method, so pandas accepts body as file-like object\n",

"if not hasattr(body, \"\_\_iter\_\_\"): body.\_\_iter\_\_ = types.MethodType( \_\_iter\_\_, body )\n",

"\n",

"data = pd.read\_csv(body)\n",

"data.head()\n"

]

},

{

"cell\_type": "markdown",

"id": "refined-milwaukee",

"metadata": {},

"source": [

"# Analyze the data"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "concrete-passion",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>...</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" <th>RainTomorrow</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2008-12-01</td>\n",

" <td>Albury</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>W</td>\n",

" <td>44.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>8.0</td>\n",

" <td>NaN</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2008-12-02</td>\n",

" <td>Albury</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>WNW</td>\n",

" <td>44.0</td>\n",

" <td>NNW</td>\n",

" <td>...</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2008-12-03</td>\n",

" <td>Albury</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>WSW</td>\n",

" <td>46.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>NaN</td>\n",

" <td>2.0</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2008-12-04</td>\n",

" <td>Albury</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>NE</td>\n",

" <td>24.0</td>\n",

" <td>SE</td>\n",

" <td>...</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2008-12-05</td>\n",

" <td>Albury</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>NaN</td>\n",

" <td>NaN</td>\n",

" <td>W</td>\n",

" <td>41.0</td>\n",

" <td>ENE</td>\n",

" <td>...</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>7.0</td>\n",

" <td>8.0</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>"

],

"text/plain": [

" Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine \\\n",

"0 2008-12-01 Albury 13.4 22.9 0.6 NaN NaN \n",

"1 2008-12-02 Albury 7.4 25.1 0.0 NaN NaN \n",

"2 2008-12-03 Albury 12.9 25.7 0.0 NaN NaN \n",

"3 2008-12-04 Albury 9.2 28.0 0.0 NaN NaN \n",

"4 2008-12-05 Albury 17.5 32.3 1.0 NaN NaN \n",

"\n",

" WindGustDir WindGustSpeed WindDir9am ... Humidity9am Humidity3pm \\\n",

"0 W 44.0 W ... 71.0 22.0 \n",

"1 WNW 44.0 NNW ... 44.0 25.0 \n",

"2 WSW 46.0 W ... 38.0 30.0 \n",

"3 NE 24.0 SE ... 45.0 16.0 \n",

"4 W 41.0 ENE ... 82.0 33.0 \n",

"\n",

" Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \\\n",

"0 1007.7 1007.1 8.0 NaN 16.9 21.8 No \n",

"1 1010.6 1007.8 NaN NaN 17.2 24.3 No \n",

"2 1007.6 1008.7 NaN 2.0 21.0 23.2 No \n",

"3 1017.6 1012.8 NaN NaN 18.1 26.5 No \n",

"4 1010.8 1006.0 7.0 8.0 17.8 29.7 No \n",

"\n",

" RainTomorrow \n",

"0 No \n",

"1 No \n",

"2 No \n",

"3 No \n",

"4 No \n",

"\n",

"[5 rows x 23 columns]"

]

},

"execution\_count": 3,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "nearby-messenger",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindSpeed9am</th>\n",

" <th>WindSpeed3pm</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>143975.000000</td>\n",

" <td>144199.000000</td>\n",

" <td>142199.000000</td>\n",

" <td>82670.000000</td>\n",

" <td>75625.000000</td>\n",

" <td>135197.000000</td>\n",

" <td>143693.000000</td>\n",

" <td>142398.000000</td>\n",

" <td>142806.000000</td>\n",

" <td>140953.000000</td>\n",

" <td>130395.00000</td>\n",

" <td>130432.000000</td>\n",

" <td>89572.000000</td>\n",

" <td>86102.000000</td>\n",

" <td>143693.000000</td>\n",

" <td>141851.00000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>12.194034</td>\n",

" <td>23.221348</td>\n",

" <td>2.360918</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>40.035230</td>\n",

" <td>14.043426</td>\n",

" <td>18.662657</td>\n",

" <td>68.880831</td>\n",

" <td>51.539116</td>\n",

" <td>1017.64994</td>\n",

" <td>1015.255889</td>\n",

" <td>4.447461</td>\n",

" <td>4.509930</td>\n",

" <td>16.990631</td>\n",

" <td>21.68339</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>6.398495</td>\n",

" <td>7.119049</td>\n",

" <td>8.478060</td>\n",

" <td>4.193704</td>\n",

" <td>3.785483</td>\n",

" <td>13.607062</td>\n",

" <td>8.915375</td>\n",

" <td>8.809800</td>\n",

" <td>19.029164</td>\n",

" <td>20.795902</td>\n",

" <td>7.10653</td>\n",

" <td>7.037414</td>\n",

" <td>2.887159</td>\n",

" <td>2.720357</td>\n",

" <td>6.488753</td>\n",

" <td>6.93665</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>-8.500000</td>\n",

" <td>-4.800000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>6.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>980.50000</td>\n",

" <td>977.100000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>-7.200000</td>\n",

" <td>-5.40000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>7.600000</td>\n",

" <td>17.900000</td>\n",

" <td>0.000000</td>\n",

" <td>2.600000</td>\n",

" <td>4.800000</td>\n",

" <td>31.000000</td>\n",

" <td>7.000000</td>\n",

" <td>13.000000</td>\n",

" <td>57.000000</td>\n",

" <td>37.000000</td>\n",

" <td>1012.90000</td>\n",

" <td>1010.400000</td>\n",

" <td>1.000000</td>\n",

" <td>2.000000</td>\n",

" <td>12.300000</td>\n",

" <td>16.60000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>12.000000</td>\n",

" <td>22.600000</td>\n",

" <td>0.000000</td>\n",

" <td>4.800000</td>\n",

" <td>8.400000</td>\n",

" <td>39.000000</td>\n",

" <td>13.000000</td>\n",

" <td>19.000000</td>\n",

" <td>70.000000</td>\n",

" <td>52.000000</td>\n",

" <td>1017.60000</td>\n",

" <td>1015.200000</td>\n",

" <td>5.000000</td>\n",

" <td>5.000000</td>\n",

" <td>16.700000</td>\n",

" <td>21.10000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>16.900000</td>\n",

" <td>28.200000</td>\n",

" <td>0.800000</td>\n",

" <td>7.400000</td>\n",

" <td>10.600000</td>\n",

" <td>48.000000</td>\n",

" <td>19.000000</td>\n",

" <td>24.000000</td>\n",

" <td>83.000000</td>\n",

" <td>66.000000</td>\n",

" <td>1022.40000</td>\n",

" <td>1020.000000</td>\n",

" <td>7.000000</td>\n",

" <td>7.000000</td>\n",

" <td>21.600000</td>\n",

" <td>26.40000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>33.900000</td>\n",

" <td>48.100000</td>\n",

" <td>371.000000</td>\n",

" <td>145.000000</td>\n",

" <td>14.500000</td>\n",

" <td>135.000000</td>\n",

" <td>130.000000</td>\n",

" <td>87.000000</td>\n",

" <td>100.000000</td>\n",

" <td>100.000000</td>\n",

" <td>1041.00000</td>\n",

" <td>1039.600000</td>\n",

" <td>9.000000</td>\n",

" <td>9.000000</td>\n",

" <td>40.200000</td>\n",

" <td>46.70000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" MinTemp MaxTemp Rainfall Evaporation \\\n",

"count 143975.000000 144199.000000 142199.000000 82670.000000 \n",

"mean 12.194034 23.221348 2.360918 5.468232 \n",

"std 6.398495 7.119049 8.478060 4.193704 \n",

"min -8.500000 -4.800000 0.000000 0.000000 \n",

"25% 7.600000 17.900000 0.000000 2.600000 \n",

"50% 12.000000 22.600000 0.000000 4.800000 \n",

"75% 16.900000 28.200000 0.800000 7.400000 \n",

"max 33.900000 48.100000 371.000000 145.000000 \n",

"\n",

" Sunshine WindGustSpeed WindSpeed9am WindSpeed3pm \\\n",

"count 75625.000000 135197.000000 143693.000000 142398.000000 \n",

"mean 7.611178 40.035230 14.043426 18.662657 \n",

"std 3.785483 13.607062 8.915375 8.809800 \n",

"min 0.000000 6.000000 0.000000 0.000000 \n",

"25% 4.800000 31.000000 7.000000 13.000000 \n",

"50% 8.400000 39.000000 13.000000 19.000000 \n",

"75% 10.600000 48.000000 19.000000 24.000000 \n",

"max 14.500000 135.000000 130.000000 87.000000 \n",

"\n",

" Humidity9am Humidity3pm Pressure9am Pressure3pm \\\n",

"count 142806.000000 140953.000000 130395.00000 130432.000000 \n",

"mean 68.880831 51.539116 1017.64994 1015.255889 \n",

"std 19.029164 20.795902 7.10653 7.037414 \n",

"min 0.000000 0.000000 980.50000 977.100000 \n",

"25% 57.000000 37.000000 1012.90000 1010.400000 \n",

"50% 70.000000 52.000000 1017.60000 1015.200000 \n",

"75% 83.000000 66.000000 1022.40000 1020.000000 \n",

"max 100.000000 100.000000 1041.00000 1039.600000 \n",

"\n",

" Cloud9am Cloud3pm Temp9am Temp3pm \n",

"count 89572.000000 86102.000000 143693.000000 141851.00000 \n",

"mean 4.447461 4.509930 16.990631 21.68339 \n",

"std 2.887159 2.720357 6.488753 6.93665 \n",

"min 0.000000 0.000000 -7.200000 -5.40000 \n",

"25% 1.000000 2.000000 12.300000 16.60000 \n",

"50% 5.000000 5.000000 16.700000 21.10000 \n",

"75% 7.000000 7.000000 21.600000 26.40000 \n",

"max 9.000000 9.000000 40.200000 46.70000 "

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.describe()"

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "owned-whole",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 145460 entries, 0 to 145459\n",

"Data columns (total 23 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 Date 145460 non-null object \n",

" 1 Location 145460 non-null object \n",

" 2 MinTemp 143975 non-null float64\n",

" 3 MaxTemp 144199 non-null float64\n",

" 4 Rainfall 142199 non-null float64\n",

" 5 Evaporation 82670 non-null float64\n",

" 6 Sunshine 75625 non-null float64\n",

" 7 WindGustDir 135134 non-null object \n",

" 8 WindGustSpeed 135197 non-null float64\n",

" 9 WindDir9am 134894 non-null object \n",

" 10 WindDir3pm 141232 non-null object \n",

" 11 WindSpeed9am 143693 non-null float64\n",

" 12 WindSpeed3pm 142398 non-null float64\n",

" 13 Humidity9am 142806 non-null float64\n",

" 14 Humidity3pm 140953 non-null float64\n",

" 15 Pressure9am 130395 non-null float64\n",

" 16 Pressure3pm 130432 non-null float64\n",

" 17 Cloud9am 89572 non-null float64\n",

" 18 Cloud3pm 86102 non-null float64\n",

" 19 Temp9am 143693 non-null float64\n",

" 20 Temp3pm 141851 non-null float64\n",

" 21 RainToday 142199 non-null object \n",

" 22 RainTomorrow 142193 non-null object \n",

"dtypes: float64(16), object(7)\n",

"memory usage: 25.5+ MB\n"

]

}

],

"source": [

"data.info()"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "terminal-olive",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(145460, 23)"

]

},

"execution\_count": 6,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "experienced-limitation",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.duplicated().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "about-fireplace",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Date 3436\n",

"Location 49\n",

"MinTemp 389\n",

"MaxTemp 505\n",

"Rainfall 681\n",

"Evaporation 358\n",

"Sunshine 145\n",

"WindGustDir 16\n",

"WindGustSpeed 67\n",

"WindDir9am 16\n",

"WindDir3pm 16\n",

"WindSpeed9am 43\n",

"WindSpeed3pm 44\n",

"Humidity9am 101\n",

"Humidity3pm 101\n",

"Pressure9am 546\n",

"Pressure3pm 549\n",

"Cloud9am 10\n",

"Cloud3pm 10\n",

"Temp9am 441\n",

"Temp3pm 502\n",

"RainToday 2\n",

"RainTomorrow 2\n",

"dtype: int64"

]

},

"execution\_count": 8,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"data.nunique()"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "valid-maple",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"No 110316\n",

"Yes 31877\n",

"Name: RainTomorrow, dtype: int64"

]

},

"execution\_count": 9,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"#target count\n",

"\n",

"data.RainTomorrow.value\_counts()"

]

},

{

"cell\_type": "markdown",

"id": "wound-spotlight",

"metadata": {},

"source": [

"# Visualization"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "immune-surfing",

"metadata": {},

"outputs": [],

"source": [

"def plotPerColumnDistribution(df, nGraphShown, nGraphPerRow):\n",

" nunique = df.nunique()\n",

" df = df[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displaying purposes, pick columns that have between 1 and 50 unique values\n",

" nRow, nCol = df.shape\n",

" columnNames = list(df)\n",

" nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow\n",

" plt.figure(num = None, figsize = (6 \* nGraphPerRow, 8 \* nGraphRow), dpi = 80, facecolor = 'w', edgecolor = 'k')\n",

" for i in range(min(nCol, nGraphShown)):\n",

" plt.subplot(nGraphRow, nGraphPerRow, i + 1)\n",

" columnDf = df.iloc[:, i]\n",

" if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):\n",

" valueCounts = columnDf.value\_counts()\n",

" valueCounts.plot.bar()\n",

" else:\n",

" columnDf.hist()\n",

" plt.ylabel('counts')\n",

" plt.xticks(rotation = 90)\n",

" plt.title(f'{columnNames[i]} (column {i})')\n",

" plt.tight\_layout(pad = 1.0, w\_pad = 1.0, h\_pad = 1.0)\n",

" plt.show()"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "collectible-gateway",

"metadata": {},

"outputs": [

{

"ename": "ValueError",

"evalue": "Number of rows must be a positive integer, not 2.8",

"output\_type": "error",

"traceback": [

"\u001b[0;31m---------------------------------------------------------------------------\u001b[0m",

"\u001b[0;31mValueError\u001b[0m Traceback (most recent call last)",

"\u001b[0;32m/tmp/wsuser/ipykernel\_164/451928612.py\u001b[0m in \u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[0;32m----> 1\u001b[0;31m \u001b[0mplotPerColumnDistribution\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mdata\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;36m10\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;36m5\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m",

"\u001b[0;32m/tmp/wsuser/ipykernel\_164/1562888048.py\u001b[0m in \u001b[0;36mplotPerColumnDistribution\u001b[0;34m(df, nGraphShown, nGraphPerRow)\u001b[0m\n\u001b[1;32m 7\u001b[0m \u001b[0mplt\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mfigure\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnum\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0;32mNone\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mfigsize\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0;34m(\u001b[0m\u001b[0;36m6\u001b[0m \u001b[0;34m\*\u001b[0m \u001b[0mnGraphPerRow\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0;36m8\u001b[0m \u001b[0;34m\*\u001b[0m \u001b[0mnGraphRow\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mdpi\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0;36m80\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mfacecolor\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0;34m'w'\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0medgecolor\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0;34m'k'\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 8\u001b[0m \u001b[0;32mfor\u001b[0m \u001b[0mi\u001b[0m \u001b[0;32min\u001b[0m \u001b[0mrange\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mmin\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnCol\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mnGraphShown\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m----> 9\u001b[0;31m \u001b[0mplt\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0msubplot\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnGraphRow\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mnGraphPerRow\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mi\u001b[0m \u001b[0;34m+\u001b[0m \u001b[0;36m1\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 10\u001b[0m \u001b[0mcolumnDf\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mdf\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0miloc\u001b[0m\u001b[0;34m[\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mi\u001b[0m\u001b[0;34m]\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 11\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0;34m(\u001b[0m\u001b[0;32mnot\u001b[0m \u001b[0mnp\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0missubdtype\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mtype\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mcolumnDf\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0miloc\u001b[0m\u001b[0;34m[\u001b[0m\u001b[0;36m0\u001b[0m\u001b[0;34m]\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mnp\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mnumber\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/matplotlib/pyplot.py\u001b[0m in \u001b[0;36msubplot\u001b[0;34m(\*args, \*\*kwargs)\u001b[0m\n\u001b[1;32m 1266\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 1267\u001b[0m \u001b[0;31m# First, search for an existing subplot with a matching spec.\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m-> 1268\u001b[0;31m \u001b[0mkey\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mSubplotSpec\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m\_from\_subplot\_args\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mfig\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0margs\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 1269\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 1270\u001b[0m \u001b[0;32mfor\u001b[0m \u001b[0max\u001b[0m \u001b[0;32min\u001b[0m \u001b[0mfig\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0maxes\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/matplotlib/gridspec.py\u001b[0m in \u001b[0;36m\_from\_subplot\_args\u001b[0;34m(figure, args)\u001b[0m\n\u001b[1;32m 593\u001b[0m f\"{len(args)} were given\")\n\u001b[1;32m 594\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 595\u001b[0;31m \u001b[0mgs\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mGridSpec\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m\_check\_gridspec\_exists\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mfigure\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mrows\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mcols\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 596\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0mgs\u001b[0m \u001b[0;32mis\u001b[0m \u001b[0;32mNone\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 597\u001b[0m \u001b[0mgs\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mGridSpec\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mrows\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mcols\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mfigure\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mfigure\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/matplotlib/gridspec.py\u001b[0m in \u001b[0;36m\_check\_gridspec\_exists\u001b[0;34m(figure, nrows, ncols)\u001b[0m\n\u001b[1;32m 221\u001b[0m \u001b[0;32mreturn\u001b[0m \u001b[0mgs\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 222\u001b[0m \u001b[0;31m# else gridspec not found:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 223\u001b[0;31m \u001b[0;32mreturn\u001b[0m \u001b[0mGridSpec\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnrows\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mncols\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mfigure\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mfigure\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m 224\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 225\u001b[0m \u001b[0;32mdef\u001b[0m \u001b[0m\_\_getitem\_\_\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mkey\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/matplotlib/gridspec.py\u001b[0m in \u001b[0;36m\_\_init\_\_\u001b[0;34m(self, nrows, ncols, figure, left, bottom, right, top, wspace, hspace, width\_ratios, height\_ratios)\u001b[0m\n\u001b[1;32m 381\u001b[0m \u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mfigure\u001b[0m \u001b[0;34m=\u001b[0m \u001b[0mfigure\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 382\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 383\u001b[0;31m super().\_\_init\_\_(nrows, ncols,\n\u001b[0m\u001b[1;32m 384\u001b[0m \u001b[0mwidth\_ratios\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mwidth\_ratios\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m 385\u001b[0m height\_ratios=height\_ratios)\n",

"\u001b[0;32m/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/matplotlib/gridspec.py\u001b[0m in \u001b[0;36m\_\_init\_\_\u001b[0;34m(self, nrows, ncols, height\_ratios, width\_ratios)\u001b[0m\n\u001b[1;32m 45\u001b[0m \"\"\"\n\u001b[1;32m 46\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0;32mnot\u001b[0m \u001b[0misinstance\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnrows\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mIntegral\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;32mor\u001b[0m \u001b[0mnrows\u001b[0m \u001b[0;34m<=\u001b[0m \u001b[0;36m0\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m---> 47\u001b[0;31m raise ValueError(\n\u001b[0m\u001b[1;32m 48\u001b[0m f\"Number of rows must be a positive integer, not {nrows!r}\")\n\u001b[1;32m 49\u001b[0m \u001b[0;32mif\u001b[0m \u001b[0;32mnot\u001b[0m \u001b[0misinstance\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mncols\u001b[0m\u001b[0;34m,\u001b[0m \u001b[0mIntegral\u001b[0m\u001b[0;34m)\u001b[0m \u001b[0;32mor\u001b[0m \u001b[0mncols\u001b[0m \u001b[0;34m<=\u001b[0m \u001b[0;36m0\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n",

"\u001b[0;31mValueError\u001b[0m: Number of rows must be a positive integer, not 2.8"

]

},

{

"data": {

"text/plain": [

"<Figure size 2400x1792 with 0 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plotPerColumnDistribution(data, 10, 5)"

]

},

{

"cell\_type": "code",

"execution\_count": 12,

"id": "consecutive-admission",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x360 with 1 Axes>"

]

},

"metadata": {

"needs\_background": "light"

},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(10,5))\n",

"sns.set\_style(\"darkgrid\")\n",

"plt.title('RainTomorrow value counts')\n",

"sns.countplot(x=data[\"RainTomorrow\"]);"

]

},

{

"cell\_type": "code",

"execution\_count": 13,

"id": "rapid-partnership",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1008x504 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(14,7))\n",

"sns.countplot(x=data[\"RainToday\"], hue=data[\"RainTomorrow\"], palette=sns.color\_palette(\"husl\")[4:]);"

]

},

{

"cell\_type": "code",

"execution\_count": 14,

"id": "martial-prediction",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1224x936 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(17,13))\n",

"sns.countplot(data=data, y='Location')\n",

"\n",

"plt.title('Location distribution')\n",

"plt.xlabel('')\n",

"plt.ylabel('')\n",

"plt.tight\_layout()"

]

},

{

"cell\_type": "code",

"execution\_count": 15,

"id": "perfect-pitch",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1224x936 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(17,13))\n",

"sns.countplot(data=data, y='Location', hue=\"RainToday\", palette=sns.color\_palette(\"Set2\"))\n",

"\n",

"plt.title('Today Rain count by LOC')\n",

"plt.xlabel('')\n",

"plt.ylabel('')\n",

"plt.tight\_layout()"

]

},

{

"cell\_type": "code",

"execution\_count": 16,

"id": "embedded-behavior",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x792 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(10,11))\n",

"plt.pie(data[\"WindDir9am\"].value\_counts(),\n",

" labels=list(data[\"WindDir9am\"].value\_counts().index),\n",

" autopct='%1.2f%%',\n",

" pctdistance=0.8,\n",

" );"

]

},

{

"cell\_type": "code",

"execution\_count": 17,

"id": "august-palestine",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 720x792 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"plt.figure(figsize=(10,11))\n",

"plt.pie(data[\"WindDir3pm\"].value\_counts(),\n",

" labels=list(data[\"WindDir3pm\"].value\_counts().index),\n",

" autopct='%1.2f%%',\n",

" pctdistance=0.8,\n",

" );"

]

},

{

"cell\_type": "code",

"execution\_count": 18,

"id": "available-squad",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 1224x936 with 16 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"data.hist(figsize=(17,13), color=\"m\");"

]

},

{

"cell\_type": "markdown",

"id": "excited-minnesota",

"metadata": {},

"source": [

"# Preprocessing"

]

},

{

"cell\_type": "code",

"execution\_count": 19,

"id": "twelve-object",

"metadata": {},

"outputs": [],

"source": [

"df\_preprocessed = data.copy()"

]

},

{

"cell\_type": "code",

"execution\_count": 20,

"id": "objective-teacher",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Date 0.000000\n",

"Location 0.000000\n",

"MinTemp 1.020899\n",

"MaxTemp 0.866905\n",

"Rainfall 2.241853\n",

"Evaporation 43.166506\n",

"Sunshine 48.009762\n",

"WindGustDir 7.098859\n",

"WindGustSpeed 7.055548\n",

"WindDir9am 7.263853\n",

"WindDir3pm 2.906641\n",

"WindSpeed9am 1.214767\n",

"WindSpeed3pm 2.105046\n",

"Humidity9am 1.824557\n",

"Humidity3pm 3.098446\n",

"Pressure9am 10.356799\n",

"Pressure3pm 10.331363\n",

"Cloud9am 38.421559\n",

"Cloud3pm 40.807095\n",

"Temp9am 1.214767\n",

"Temp3pm 2.481094\n",

"RainToday 2.241853\n",

"RainTomorrow 2.245978\n",

"dtype: float64"

]

},

"execution\_count": 20,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed.isnull().mean() \* 100"

]

},

{

"cell\_type": "code",

"execution\_count": 21,

"id": "incident-costume",

"metadata": {},

"outputs": [

{

"name": "stderr",

"output\_type": "stream",

"text": [

"/tmp/wsuser/ipykernel\_164/2563157355.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.\n",

" mean = df\_preprocessed.mean()\n"

]

},

{

"data": {

"text/plain": [

"Date 0\n",

"Location 0\n",

"MinTemp 0\n",

"MaxTemp 0\n",

"Rainfall 0\n",

"Evaporation 0\n",

"Sunshine 0\n",

"WindGustDir 10326\n",

"WindGustSpeed 0\n",

"WindDir9am 10566\n",

"WindDir3pm 4228\n",

"WindSpeed9am 0\n",

"WindSpeed3pm 0\n",

"Humidity9am 0\n",

"Humidity3pm 0\n",

"Pressure9am 0\n",

"Pressure3pm 0\n",

"Cloud9am 0\n",

"Cloud3pm 0\n",

"Temp9am 0\n",

"Temp3pm 0\n",

"RainToday 3261\n",

"RainTomorrow 3267\n",

"dtype: int64"

]

},

"execution\_count": 21,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"mean = df\_preprocessed.mean()\n",

"df\_preprocessed.fillna(mean, inplace=True)\n",

"\n",

"df\_preprocessed.isna().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 22,

"id": "worldwide-dialogue",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Date 0\n",

"Location 0\n",

"MinTemp 0\n",

"MaxTemp 0\n",

"Rainfall 0\n",

"Evaporation 0\n",

"Sunshine 0\n",

"WindGustDir 0\n",

"WindGustSpeed 0\n",

"WindDir9am 0\n",

"WindDir3pm 0\n",

"WindSpeed9am 0\n",

"WindSpeed3pm 0\n",

"Humidity9am 0\n",

"Humidity3pm 0\n",

"Pressure9am 0\n",

"Pressure3pm 0\n",

"Cloud9am 0\n",

"Cloud3pm 0\n",

"Temp9am 0\n",

"Temp3pm 0\n",

"RainToday 0\n",

"RainTomorrow 0\n",

"dtype: int64"

]

},

"execution\_count": 22,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed.replace(np.nan, 'NaN', inplace=True)\n",

"df\_preprocessed.isna().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 23,

"id": "sealed-python",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>...</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" <th>RainTomorrow</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2008-12-01</td>\n",

" <td>Albury</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>W</td>\n",

" <td>44.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>8.000000</td>\n",

" <td>4.50993</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2008-12-02</td>\n",

" <td>Albury</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>WNW</td>\n",

" <td>44.0</td>\n",

" <td>NNW</td>\n",

" <td>...</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2008-12-03</td>\n",

" <td>Albury</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>WSW</td>\n",

" <td>46.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>4.447461</td>\n",

" <td>2.00000</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2008-12-04</td>\n",

" <td>Albury</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>NE</td>\n",

" <td>24.0</td>\n",

" <td>SE</td>\n",

" <td>...</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2008-12-05</td>\n",

" <td>Albury</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>W</td>\n",

" <td>41.0</td>\n",

" <td>ENE</td>\n",

" <td>...</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>7.000000</td>\n",

" <td>8.00000</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>"

],

"text/plain": [

" Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine \\\n",

"0 2008-12-01 Albury 13.4 22.9 0.6 5.468232 7.611178 \n",

"1 2008-12-02 Albury 7.4 25.1 0.0 5.468232 7.611178 \n",

"2 2008-12-03 Albury 12.9 25.7 0.0 5.468232 7.611178 \n",

"3 2008-12-04 Albury 9.2 28.0 0.0 5.468232 7.611178 \n",

"4 2008-12-05 Albury 17.5 32.3 1.0 5.468232 7.611178 \n",

"\n",

" WindGustDir WindGustSpeed WindDir9am ... Humidity9am Humidity3pm \\\n",

"0 W 44.0 W ... 71.0 22.0 \n",

"1 WNW 44.0 NNW ... 44.0 25.0 \n",

"2 WSW 46.0 W ... 38.0 30.0 \n",

"3 NE 24.0 SE ... 45.0 16.0 \n",

"4 W 41.0 ENE ... 82.0 33.0 \n",

"\n",

" Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \\\n",

"0 1007.7 1007.1 8.000000 4.50993 16.9 21.8 No \n",

"1 1010.6 1007.8 4.447461 4.50993 17.2 24.3 No \n",

"2 1007.6 1008.7 4.447461 2.00000 21.0 23.2 No \n",

"3 1017.6 1012.8 4.447461 4.50993 18.1 26.5 No \n",

"4 1010.8 1006.0 7.000000 8.00000 17.8 29.7 No \n",

"\n",

" RainTomorrow \n",

"0 No \n",

"1 No \n",

"2 No \n",

"3 No \n",

"4 No \n",

"\n",

"[5 rows x 23 columns]"

]

},

"execution\_count": 23,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 24,

"id": "advised-external",

"metadata": {},

"outputs": [],

"source": [

"df\_preprocessed['Date'] = pd.to\_datetime(df\_preprocessed['Date'])"

]

},

{

"cell\_type": "code",

"execution\_count": 25,

"id": "environmental-theme",

"metadata": {},

"outputs": [],

"source": [

"#columns segregation\n",

"date\_columns = ['Date']\n",

"cat\_columns = df\_preprocessed.select\_dtypes(include=['object']).columns\n",

"num\_columns = df\_preprocessed.select\_dtypes(include=['float64', 'int64']).columns\n",

"target\_col = 'RainTomorrow'"

]

},

{

"cell\_type": "markdown",

"id": "pleased-departure",

"metadata": {},

"source": [

"Encoding categorical variables"

]

},

{

"cell\_type": "code",

"execution\_count": 26,

"id": "qualified-rachel",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>...</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" <th>RainTomorrow</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2008-12-01</td>\n",

" <td>Albury</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>W</td>\n",

" <td>44.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>8.000000</td>\n",

" <td>4.50993</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2008-12-02</td>\n",

" <td>Albury</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>WNW</td>\n",

" <td>44.0</td>\n",

" <td>NNW</td>\n",

" <td>...</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2008-12-03</td>\n",

" <td>Albury</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>WSW</td>\n",

" <td>46.0</td>\n",

" <td>W</td>\n",

" <td>...</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>4.447461</td>\n",

" <td>2.00000</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2008-12-04</td>\n",

" <td>Albury</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>NE</td>\n",

" <td>24.0</td>\n",

" <td>SE</td>\n",

" <td>...</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2008-12-05</td>\n",

" <td>Albury</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>W</td>\n",

" <td>41.0</td>\n",

" <td>ENE</td>\n",

" <td>...</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>7.000000</td>\n",

" <td>8.00000</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>"

],

"text/plain": [

" Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine \\\n",

"0 2008-12-01 Albury 13.4 22.9 0.6 5.468232 7.611178 \n",

"1 2008-12-02 Albury 7.4 25.1 0.0 5.468232 7.611178 \n",

"2 2008-12-03 Albury 12.9 25.7 0.0 5.468232 7.611178 \n",

"3 2008-12-04 Albury 9.2 28.0 0.0 5.468232 7.611178 \n",

"4 2008-12-05 Albury 17.5 32.3 1.0 5.468232 7.611178 \n",

"\n",

" WindGustDir WindGustSpeed WindDir9am ... Humidity9am Humidity3pm \\\n",

"0 W 44.0 W ... 71.0 22.0 \n",

"1 WNW 44.0 NNW ... 44.0 25.0 \n",

"2 WSW 46.0 W ... 38.0 30.0 \n",

"3 NE 24.0 SE ... 45.0 16.0 \n",

"4 W 41.0 ENE ... 82.0 33.0 \n",

"\n",

" Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \\\n",

"0 1007.7 1007.1 8.000000 4.50993 16.9 21.8 No \n",

"1 1010.6 1007.8 4.447461 4.50993 17.2 24.3 No \n",

"2 1007.6 1008.7 4.447461 2.00000 21.0 23.2 No \n",

"3 1017.6 1012.8 4.447461 4.50993 18.1 26.5 No \n",

"4 1010.8 1006.0 7.000000 8.00000 17.8 29.7 No \n",

"\n",

" RainTomorrow \n",

"0 No \n",

"1 No \n",

"2 No \n",

"3 No \n",

"4 No \n",

"\n",

"[5 rows x 23 columns]"

]

},

"execution\_count": 26,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 27,

"id": "opposite-significance",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"Index(['Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday',\n",

" 'RainTomorrow'],\n",

" dtype='object')"

]

},

"execution\_count": 27,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"cat\_columns"

]

},

{

"cell\_type": "code",

"execution\_count": 28,

"id": "korean-finding",

"metadata": {},

"outputs": [],

"source": [

"import numpy as np\n",

"from sklearn.impute import SimpleImputer\n",

"imp\_mode=SimpleImputer(missing\_values=np.nan,strategy='most\_frequent')"

]

},

{

"cell\_type": "code",

"execution\_count": 29,

"id": "adapted-easter",

"metadata": {},

"outputs": [],

"source": [

"data\_cat = data[['Location','RainToday','WindGustDir','WindDir9am','WindDir3pm','RainTomorrow']]"

]

},

{

"cell\_type": "code",

"execution\_count": 30,

"id": "functioning-classroom",

"metadata": {},

"outputs": [],

"source": [

"#Fitting and Transforming the missing data\n",

"data\_cat = imp\_mode.fit\_transform(data\_cat)"

]

},

{

"cell\_type": "code",

"execution\_count": 31,

"id": "worst-zimbabwe",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Unique Values in Location Column: 49\n",

"-----------------------------------------------------\n",

"Unique Values in WindGustDir Column: 17\n",

"-----------------------------------------------------\n",

"Unique Values in WindDir9am Column: 17\n",

"-----------------------------------------------------\n",

"Unique Values in WindDir3pm Column: 17\n",

"-----------------------------------------------------\n",

"Unique Values in RainToday Column: 3\n",

"-----------------------------------------------------\n",

"Unique Values in RainTomorrow Column: 3\n",

"-----------------------------------------------------\n"

]

}

],

"source": [

"for i in cat\_columns:\n",

" print('Unique Values in ' + i + ' Column:', end = \" \")\n",

" print(len(df\_preprocessed[i].unique()))\n",

" print('-----------------------------------------------------')"

]

},

{

"cell\_type": "code",

"execution\_count": 32,

"id": "spectacular-fisher",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>...</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" <th>RainTomorrow</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2008-12-01</td>\n",

" <td>2</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>14</td>\n",

" <td>44.0</td>\n",

" <td>14</td>\n",

" <td>...</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>8.000000</td>\n",

" <td>4.50993</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2008-12-02</td>\n",

" <td>2</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>15</td>\n",

" <td>44.0</td>\n",

" <td>6</td>\n",

" <td>...</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2008-12-03</td>\n",

" <td>2</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>16</td>\n",

" <td>46.0</td>\n",

" <td>14</td>\n",

" <td>...</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>4.447461</td>\n",

" <td>2.00000</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2008-12-04</td>\n",

" <td>2</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>4</td>\n",

" <td>24.0</td>\n",

" <td>10</td>\n",

" <td>...</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2008-12-05</td>\n",

" <td>2</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>14</td>\n",

" <td>41.0</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>7.000000</td>\n",

" <td>8.00000</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>No</td>\n",

" <td>No</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>"

],

"text/plain": [

" Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine \\\n",

"0 2008-12-01 2 13.4 22.9 0.6 5.468232 7.611178 \n",

"1 2008-12-02 2 7.4 25.1 0.0 5.468232 7.611178 \n",

"2 2008-12-03 2 12.9 25.7 0.0 5.468232 7.611178 \n",

"3 2008-12-04 2 9.2 28.0 0.0 5.468232 7.611178 \n",

"4 2008-12-05 2 17.5 32.3 1.0 5.468232 7.611178 \n",

"\n",

" WindGustDir WindGustSpeed WindDir9am ... Humidity9am Humidity3pm \\\n",

"0 14 44.0 14 ... 71.0 22.0 \n",

"1 15 44.0 6 ... 44.0 25.0 \n",

"2 16 46.0 14 ... 38.0 30.0 \n",

"3 4 24.0 10 ... 45.0 16.0 \n",

"4 14 41.0 1 ... 82.0 33.0 \n",

"\n",

" Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \\\n",

"0 1007.7 1007.1 8.000000 4.50993 16.9 21.8 No \n",

"1 1010.6 1007.8 4.447461 4.50993 17.2 24.3 No \n",

"2 1007.6 1008.7 4.447461 2.00000 21.0 23.2 No \n",

"3 1017.6 1012.8 4.447461 4.50993 18.1 26.5 No \n",

"4 1010.8 1006.0 7.000000 8.00000 17.8 29.7 No \n",

"\n",

" RainTomorrow \n",

"0 No \n",

"1 No \n",

"2 No \n",

"3 No \n",

"4 No \n",

"\n",

"[5 rows x 23 columns]"

]

},

"execution\_count": 32,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"from sklearn.preprocessing import LabelEncoder\n",

"le = LabelEncoder()\n",

"for i in ['Location','WindGustDir','WindDir9am','WindDir3pm']:\n",

" df\_preprocessed[i] = le.fit\_transform(df\_preprocessed[i])\n",

"df\_preprocessed.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 33,

"id": "british-judgment",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RainToday\n",

"No 110319\n",

"Yes 31880\n",

"NaN 3261\n",

"dtype: int64"

]

},

"execution\_count": 33,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed[['RainToday']].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 34,

"id": "rural-tender",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RainTomorrow\n",

"No 110316\n",

"Yes 31877\n",

"NaN 3267\n",

"dtype: int64"

]

},

"execution\_count": 34,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed[['RainTomorrow']].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 35,

"id": "devoted-knife",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"RainToday 0\n",

"RainTomorrow 0\n",

"dtype: int64"

]

},

"execution\_count": 35,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df\_preprocessed[['RainToday','RainTomorrow']].isna().sum()"

]

},

{

"cell\_type": "code",

"execution\_count": 36,

"id": "prime-colonial",

"metadata": {},

"outputs": [],

"source": [

"dataframe = df\_preprocessed[(df\_preprocessed != \"NaN\").all(axis=1)]\n",

"del df\_preprocessed"

]

},

{

"cell\_type": "code",

"execution\_count": 37,

"id": "extraordinary-animation",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"No 109332\n",

"Yes 31455\n",

"Name: RainToday, dtype: int64"

]

},

"execution\_count": 37,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataframe['RainToday'].value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 38,

"id": "mathematical-comparative",

"metadata": {},

"outputs": [],

"source": [

"label\_dict = {'No': 0,'Yes':1}\n",

"dataframe['RainTomorrow'] = dataframe['RainTomorrow'].map(label\_dict)\n",

"dataframe['RainToday'] = dataframe['RainToday'].map(label\_dict)"

]

},

{

"cell\_type": "code",

"execution\_count": 39,

"id": "timely-promise",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Date</th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>Evaporation</th>\n",

" <th>Sunshine</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>...</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Cloud9am</th>\n",

" <th>Cloud3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" <th>RainTomorrow</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2008-12-01</td>\n",

" <td>2</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>14</td>\n",

" <td>44.0</td>\n",

" <td>14</td>\n",

" <td>...</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>8.000000</td>\n",

" <td>4.50993</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2008-12-02</td>\n",

" <td>2</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>15</td>\n",

" <td>44.0</td>\n",

" <td>6</td>\n",

" <td>...</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2008-12-03</td>\n",

" <td>2</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>16</td>\n",

" <td>46.0</td>\n",

" <td>14</td>\n",

" <td>...</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>4.447461</td>\n",

" <td>2.00000</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2008-12-04</td>\n",

" <td>2</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>4</td>\n",

" <td>24.0</td>\n",

" <td>10</td>\n",

" <td>...</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>4.447461</td>\n",

" <td>4.50993</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2008-12-05</td>\n",

" <td>2</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>5.468232</td>\n",

" <td>7.611178</td>\n",

" <td>14</td>\n",

" <td>41.0</td>\n",

" <td>1</td>\n",

" <td>...</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>7.000000</td>\n",

" <td>8.00000</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>0</td>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"<p>5 rows × 23 columns</p>\n",

"</div>"

],

"text/plain": [

" Date Location MinTemp MaxTemp Rainfall Evaporation Sunshine \\\n",

"0 2008-12-01 2 13.4 22.9 0.6 5.468232 7.611178 \n",

"1 2008-12-02 2 7.4 25.1 0.0 5.468232 7.611178 \n",

"2 2008-12-03 2 12.9 25.7 0.0 5.468232 7.611178 \n",

"3 2008-12-04 2 9.2 28.0 0.0 5.468232 7.611178 \n",

"4 2008-12-05 2 17.5 32.3 1.0 5.468232 7.611178 \n",

"\n",

" WindGustDir WindGustSpeed WindDir9am ... Humidity9am Humidity3pm \\\n",

"0 14 44.0 14 ... 71.0 22.0 \n",

"1 15 44.0 6 ... 44.0 25.0 \n",

"2 16 46.0 14 ... 38.0 30.0 \n",

"3 4 24.0 10 ... 45.0 16.0 \n",

"4 14 41.0 1 ... 82.0 33.0 \n",

"\n",

" Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \\\n",

"0 1007.7 1007.1 8.000000 4.50993 16.9 21.8 0 \n",

"1 1010.6 1007.8 4.447461 4.50993 17.2 24.3 0 \n",

"2 1007.6 1008.7 4.447461 2.00000 21.0 23.2 0 \n",

"3 1017.6 1012.8 4.447461 4.50993 18.1 26.5 0 \n",

"4 1010.8 1006.0 7.000000 8.00000 17.8 29.7 0 \n",

"\n",

" RainTomorrow \n",

"0 0 \n",

"1 0 \n",

"2 0 \n",

"3 0 \n",

"4 0 \n",

"\n",

"[5 rows x 23 columns]"

]

},

"execution\_count": 39,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"dataframe.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 40,

"id": "secret-gender",

"metadata": {},

"outputs": [],

"source": [

"# removing Date,Evoporation,Sunshine,Cloud9am,Cloud3pm\n",

"dataframe.drop(['Date','Evaporation','Sunshine','Cloud9am','Cloud3pm'], axis=1, inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 41,

"id": "under-boards",

"metadata": {},

"outputs": [],

"source": [

"dataframe.reset\_index(drop=True, inplace=True)"

]

},

{

"cell\_type": "code",

"execution\_count": 42,

"id": "filled-failure",

"metadata": {},

"outputs": [],

"source": [

"## train test split\n",

"\n",

"from sklearn.model\_selection import train\_test\_split\n",

"\n",

"X = dataframe.drop(columns=[\"RainTomorrow\"])\n",

"y = dataframe[\"RainTomorrow\"]\n",

"\n",

"X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)"

]

},

{

"cell\_type": "code",

"execution\_count": 43,

"id": "conscious-advice",

"metadata": {},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Location</th>\n",

" <th>MinTemp</th>\n",

" <th>MaxTemp</th>\n",

" <th>Rainfall</th>\n",

" <th>WindGustDir</th>\n",

" <th>WindGustSpeed</th>\n",

" <th>WindDir9am</th>\n",

" <th>WindDir3pm</th>\n",

" <th>WindSpeed9am</th>\n",

" <th>WindSpeed3pm</th>\n",

" <th>Humidity9am</th>\n",

" <th>Humidity3pm</th>\n",

" <th>Pressure9am</th>\n",

" <th>Pressure3pm</th>\n",

" <th>Temp9am</th>\n",

" <th>Temp3pm</th>\n",

" <th>RainToday</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>0</th>\n",

" <td>2</td>\n",

" <td>13.4</td>\n",

" <td>22.9</td>\n",

" <td>0.6</td>\n",

" <td>14</td>\n",

" <td>44.0</td>\n",

" <td>14</td>\n",

" <td>15</td>\n",

" <td>20.0</td>\n",

" <td>24.0</td>\n",

" <td>71.0</td>\n",

" <td>22.0</td>\n",

" <td>1007.7</td>\n",

" <td>1007.1</td>\n",

" <td>16.9</td>\n",

" <td>21.8</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>1</th>\n",

" <td>2</td>\n",

" <td>7.4</td>\n",

" <td>25.1</td>\n",

" <td>0.0</td>\n",

" <td>15</td>\n",

" <td>44.0</td>\n",

" <td>6</td>\n",

" <td>16</td>\n",

" <td>4.0</td>\n",

" <td>22.0</td>\n",

" <td>44.0</td>\n",

" <td>25.0</td>\n",

" <td>1010.6</td>\n",

" <td>1007.8</td>\n",

" <td>17.2</td>\n",

" <td>24.3</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>2</th>\n",

" <td>2</td>\n",

" <td>12.9</td>\n",

" <td>25.7</td>\n",

" <td>0.0</td>\n",

" <td>16</td>\n",

" <td>46.0</td>\n",

" <td>14</td>\n",

" <td>16</td>\n",

" <td>19.0</td>\n",

" <td>26.0</td>\n",

" <td>38.0</td>\n",

" <td>30.0</td>\n",

" <td>1007.6</td>\n",

" <td>1008.7</td>\n",

" <td>21.0</td>\n",

" <td>23.2</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>3</th>\n",

" <td>2</td>\n",

" <td>9.2</td>\n",

" <td>28.0</td>\n",

" <td>0.0</td>\n",

" <td>4</td>\n",

" <td>24.0</td>\n",

" <td>10</td>\n",

" <td>0</td>\n",

" <td>11.0</td>\n",

" <td>9.0</td>\n",

" <td>45.0</td>\n",

" <td>16.0</td>\n",

" <td>1017.6</td>\n",

" <td>1012.8</td>\n",

" <td>18.1</td>\n",

" <td>26.5</td>\n",

" <td>0</td>\n",

" </tr>\n",

" <tr>\n",

" <th>4</th>\n",

" <td>2</td>\n",

" <td>17.5</td>\n",

" <td>32.3</td>\n",

" <td>1.0</td>\n",

" <td>14</td>\n",

" <td>41.0</td>\n",

" <td>1</td>\n",

" <td>7</td>\n",

" <td>7.0</td>\n",

" <td>20.0</td>\n",

" <td>82.0</td>\n",

" <td>33.0</td>\n",

" <td>1010.8</td>\n",

" <td>1006.0</td>\n",

" <td>17.8</td>\n",

" <td>29.7</td>\n",

" <td>0</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Location MinTemp MaxTemp Rainfall WindGustDir WindGustSpeed \\\n",

"0 2 13.4 22.9 0.6 14 44.0 \n",

"1 2 7.4 25.1 0.0 15 44.0 \n",

"2 2 12.9 25.7 0.0 16 46.0 \n",

"3 2 9.2 28.0 0.0 4 24.0 \n",

"4 2 17.5 32.3 1.0 14 41.0 \n",

"\n",

" WindDir9am WindDir3pm WindSpeed9am WindSpeed3pm Humidity9am \\\n",

"0 14 15 20.0 24.0 71.0 \n",

"1 6 16 4.0 22.0 44.0 \n",

"2 14 16 19.0 26.0 38.0 \n",

"3 10 0 11.0 9.0 45.0 \n",

"4 1 7 7.0 20.0 82.0 \n",

"\n",

" Humidity3pm Pressure9am Pressure3pm Temp9am Temp3pm RainToday \n",

"0 22.0 1007.7 1007.1 16.9 21.8 0 \n",

"1 25.0 1010.6 1007.8 17.2 24.3 0 \n",

"2 30.0 1007.6 1008.7 21.0 23.2 0 \n",

"3 16.0 1017.6 1012.8 18.1 26.5 0 \n",

"4 33.0 1010.8 1006.0 17.8 29.7 0 "

]

},

"execution\_count": 43,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"X.head()"

]

},

{

"cell\_type": "code",

"execution\_count": 44,

"id": "loving-poster",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"0 0\n",

"1 0\n",

"2 0\n",

"3 0\n",

"4 0\n",

"Name: RainTomorrow, dtype: int64"

]

},

"execution\_count": 44,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"y.head()"

]

},

{

"cell\_type": "markdown",

"id": "tutorial-birth",

"metadata": {},

"source": [

"# Feature Scaling"

]

},

{

"cell\_type": "code",

"execution\_count": 45,

"id": "advanced-lexington",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.preprocessing import MinMaxScaler\n",

"\n",

"scaler = MinMaxScaler()\n",

"X\_train\_scaled = scaler.fit\_transform(X\_train)\n",

"X\_test\_scaled = scaler.transform(X\_test)"

]

},

{

"cell\_type": "code",

"execution\_count": 46,

"id": "criminal-catch",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"(98550, 17)"

]

},

"execution\_count": 46,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"X\_train\_scaled.shape"

]

},

{

"cell\_type": "code",

"execution\_count": 47,

"id": "marked-journey",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([3.54166667e-01, 5.21226415e-01, 5.19157088e-01, 5.39083558e-04,\n",

" 1.25000000e-01, 3.25581395e-01, 4.37500000e-01, 4.37500000e-01,\n",

" 1.53846154e-02, 1.32530120e-01, 6.96969697e-01, 6.80000000e-01,\n",

" 3.20338983e-01, 3.19218241e-01, 5.47413793e-01, 5.19305019e-01,\n",

" 0.00000000e+00])"

]

},

"execution\_count": 47,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"X\_train\_scaled[0,:]"

]

},

{

"cell\_type": "markdown",

"id": "practical-margin",

"metadata": {},

"source": [

"# Feature Selection"

]

},

{

"cell\_type": "code",

"execution\_count": 48,

"id": "frequent-joining",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.feature\_selection import SelectKBest\n",

"\n",

"fs = SelectKBest(k=10)\n",

"X\_train\_scaled = fs.fit\_transform(X\_train\_scaled, y\_train)\n",

"X\_test\_scaled = fs.transform(X\_test\_scaled)"

]

},

{

"cell\_type": "markdown",

"id": "entertaining-renaissance",

"metadata": {},

"source": [

"# Training and testing the model"

]

},

{

"cell\_type": "code",

"execution\_count": 49,

"id": "spiritual-pontiac",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.ensemble import RandomForestClassifier"

]

},

{

"cell\_type": "code",

"execution\_count": 50,

"id": "special-elizabeth",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.svm import SVC"

]

},

{

"cell\_type": "code",

"execution\_count": 51,

"id": "quantitative-grace",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.tree import DecisionTreeClassifier"

]

},

{

"cell\_type": "code",

"execution\_count": 52,

"id": "sitting-malpractice",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.ensemble import GradientBoostingClassifier"

]

},

{

"cell\_type": "code",

"execution\_count": 53,

"id": "spare-naples",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.linear\_model import LogisticRegression"

]

},

{

"cell\_type": "code",

"execution\_count": 54,

"id": "applied-witness",

"metadata": {},

"outputs": [],

"source": [

"#Models initialization of the models\n",

"XGBoost = xgboost.XGBRFClassifier()\n",

"Rand\_forest = RandomForestClassifier()\n",

"svm = SVC()\n",

"Dtree = DecisionTreeClassifier()\n",

"GBM = GradientBoostingClassifier()\n",

"log = LogisticRegression()"

]

},

{

"cell\_type": "markdown",

"id": "04e41945",

"metadata": {},

"source": [

"Comparing all the model's accuracy, Random Forest scores the highest. So, Random Forest model is selected."

]

},

{

"cell\_type": "markdown",

"id": "affected-investigator",

"metadata": {},

"source": [

"# Fitting the model"

]

},

{

"cell\_type": "code",

"execution\_count": 56,

"id": "caroline-decade",

"metadata": {},

"outputs": [],

"source": [

"rand\_mod = Rand\_forest.fit(X\_train\_scaled,y\_train)"

]

},

{

"cell\_type": "markdown",

"id": "alleged-algorithm",

"metadata": {},

"source": [

"# Predicting the Train values"

]

},

{

"cell\_type": "code",

"execution\_count": 57,

"id": "human-feelings",

"metadata": {},

"outputs": [],

"source": [

"p1 = Rand\_forest.predict(X\_train\_scaled)"

]

},

{

"cell\_type": "markdown",

"id": "personal-norwegian",

"metadata": {},

"source": [

"# Model Evaluation"

]

},

{

"cell\_type": "markdown",

"id": "italian-minneapolis",

"metadata": {},

"source": [

"# 1.Accuracy"

]

},

{

"cell\_type": "code",

"execution\_count": 60,

"id": "careful-antique",

"metadata": {},

"outputs": [],

"source": [

"t1 = Rand\_forest.predict(X\_test\_scaled)"

]

},

{

"cell\_type": "code",

"execution\_count": 61,

"id": "impressed-harassment",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Rand\_forest: 0.8475507256670691\n"

]

}

],

"source": [

"print(\"Rand\_forest:\",metrics.accuracy\_score(y\_test,t1))"

]

},

{

"cell\_type": "markdown",

"id": "stupid-venezuela",

"metadata": {},

"source": [

"# 2.Confusion Matrix"

]

},

{

"cell\_type": "markdown",

"id": "quarterly-criticism",

"metadata": {},

"source": [

"# Random forest Confusion matrix"

]

},

{

"cell\_type": "code",

"execution\_count": 62,

"id": "still-concert",

"metadata": {},

"outputs": [],

"source": [

"conf\_matrix = metrics.confusion\_matrix(y\_test,t1)"

]

},

{

"cell\_type": "code",

"execution\_count": 63,

"id": "multiple-matthew",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 540x540 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"fig,ax = plt.subplots(figsize=(7.5,7.5))\n",

"ax.matshow(conf\_matrix,alpha=0.3)\n",

"for i in range(conf\_matrix.shape[0]):\n",

" for j in range(conf\_matrix.shape[1]):\n",

" ax.text(x=j, y=i, s=conf\_matrix[i,j], va ='center', ha='center',size='xx-large')\n",

"plt.xlabel('Predictions',fontsize=18)\n",

"plt.ylabel('Actuals',fontsize=18)\n",

"plt.title('Confusion Matrix',fontsize=18)\n",

"plt.show()"

]

},

{

"cell\_type": "code",

"execution\_count": 64,

"id": "built-scheme",

"metadata": {},

"outputs": [],

"source": [

"from sklearn.metrics import accuracy\_score\n",

"from sklearn.metrics import precision\_score\n",

"from sklearn.metrics import recall\_score\n",

"from sklearn.metrics import f1\_score"

]

},

{

"cell\_type": "code",

"execution\_count": 65,

"id": "toxic-license",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"[[31184 1636]\n",

" [ 4803 4614]]\n",

"Accuracy: 0.8475507256670691\n",

"Precision: 0.73824\n",

"Recall: 0.48996495699267284\n",

"F1-score: 0.5890087444947979\n"

]

}

],

"source": [

"print(conf\_matrix)\n",

"print(\"Accuracy:\",accuracy\_score(y\_test,t1))\n",

"print(\"Precision:\",precision\_score(y\_test,t1))\n",

"print(\"Recall:\",recall\_score(y\_test,t1))\n",

"print(\"F1-score:\",f1\_score(y\_test,t1))"

]

},

{

"cell\_type": "markdown",

"id": "regional-vietnamese",

"metadata": {},

"source": [

"# 3.ROC AUC SCORE"

]

},

{

"cell\_type": "code",

"execution\_count": 66,

"id": "changing-newport",

"metadata": {},

"outputs": [

{

"data": {

"image/png": "\n",

"text/plain": [

"<Figure size 960x800 with 1 Axes>"

]

},

"metadata": {},

"output\_type": "display\_data"

}

],

"source": [

"auc = metrics.roc\_auc\_score(y\_test,t1)\n",

"\n",

"fpr,tpr,thresholds = metrics.roc\_curve(y\_test,t1)\n",

"\n",

"plt.figure(figsize = (12,10),dpi=80)\n",

"plt.axis('scaled')\n",

"plt.xlim([0,1])\n",

"plt.ylim([0,1])\n",

"plt.title(\"AUC & ROC Curve\")\n",

"plt.plot(fpr,tpr,'v')\n",

"plt.fill\_between(fpr,tpr,facecolor = 'blue',alpha=0.8)\n",

"plt.text(1,0.05, 'AUC =%0.4f' % auc, ha='right', fontsize =10, weight ='bold' , color = 'black')\n",

"plt.xlabel(\"False Positive Rate\")\n",

"plt.ylabel(\"True Positive Rate\")\n",

"plt.show()"

]

},

{

"cell\_type": "code",

"execution\_count": 67,

"id": "1fb9ef64",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Requirement already satisfied: ibm\_watson\_machine\_learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.255)\n",

"Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (21.3)\n",

"Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (4.8.2)\n",

"Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (1.26.7)\n",

"Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (2022.9.24)\n",

"Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (1.3.4)\n",

"Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (0.8.9)\n",

"Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (0.3.3)\n",

"Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (2.26.0)\n",

"Requirement already satisfied: ibm-cos-sdk==2.11.\* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm\_watson\_machine\_learning) (2.11.0)\n",

"Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm\_watson\_machine\_learning) (2.11.0)\n",

"Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm\_watson\_machine\_learning) (2.11.0)\n",

"Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm\_watson\_machine\_learning) (0.10.0)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.\*->ibm\_watson\_machine\_learning) (2.8.2)\n",

"Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm\_watson\_machine\_learning) (2021.3)\n",

"Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.5.0,>=0.24.2->ibm\_watson\_machine\_learning) (1.20.3)\n",

"Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.\*->ibm\_watson\_machine\_learning) (1.15.0)\n",

"Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm\_watson\_machine\_learning) (2.0.4)\n",

"Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->ibm\_watson\_machine\_learning) (3.3)\n",

"Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-metadata->ibm\_watson\_machine\_learning) (3.6.0)\n",

"Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from packaging->ibm\_watson\_machine\_learning) (3.0.4)\n"

]

}

],

"source": [

"!pip install ibm\_watson\_machine\_learning"

]

},

{

"cell\_type": "code",

"execution\_count": 68,

"id": "ad3d6a74",

"metadata": {},

"outputs": [],

"source": [

"from ibm\_watson\_machine\_learning import APIClient\n",

"wml\_credentials = {\n",

" \"url\": \"https://us-south.ml.cloud.ibm.com\",\n",

" \"apikey\" : \"PQBr9MBF7mFuSh2VVLfOE-liIA04VH-h5VEk8EfjFIuw\"\n",

"}\n",

"\n",

"client = APIClient(wml\_credentials)"

]

},

{

"cell\_type": "code",

"execution\_count": 69,

"id": "9e9fe85c",

"metadata": {},

"outputs": [],

"source": [

"def guid\_from\_space\_name(client,space\_name):\n",

" space = client.spaces.get\_details()\n",

" #print(space)\n",

" return(next(item for item in space['resources'] if item['entity'][\"name\"] == space\_name)['metadata']['id'])"

]

},

{

"cell\_type": "code",

"execution\_count": 71,

"id": "b14cf3e9",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"Space UID = 48a743e2-399c-4c37-93fb-e7412163309b\n"

]

}

],

"source": [

"space\_uid = guid\_from\_space\_name(client,'models')\n",

"print(\"Space UID = \" + space\_uid)"

]

},

{

"cell\_type": "code",

"execution\_count": 73,

"id": "ac5ae3ae",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'SUCCESS'"

]

},

"execution\_count": 73,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"client.set.default\_space(space\_uid)"

]

},

{

"cell\_type": "code",

"execution\_count": 96,

"id": "ce3ba750",

"metadata": {},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"----------------------------- ------------------------------------ ----\n",

"NAME ASSET\_ID TYPE\n",

"default\_py3.6 0062b8c9-8b7d-44a0-a9b9-46c416adcbd9 base\n",

"kernel-spark3.2-scala2.12 020d69ce-7ac1-5e68-ac1a-31189867356a base\n",

"pytorch-onnx\_1.3-py3.7-edt 069ea134-3346-5748-b513-49120e15d288 base\n",

"scikit-learn\_0.20-py3.6 09c5a1d0-9c1e-4473-a344-eb7b665ff687 base\n",

"spark-mllib\_3.0-scala\_2.12 09f4cff0-90a7-5899-b9ed-1ef348aebdee base\n",

"pytorch-onnx\_rt22.1-py3.9 0b848dd4-e681-5599-be41-b5f6fccc6471 base\n",

"ai-function\_0.1-py3.6 0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda base\n",

"shiny-r3.6 0e6e79df-875e-4f24-8ae9-62dcc2148306 base\n",

"tensorflow\_2.4-py3.7-horovod 1092590a-307d-563d-9b62-4eb7d64b3f22 base\n",

"pytorch\_1.1-py3.6 10ac12d6-6b30-4ccd-8392-3e922c096a92 base\n",

"tensorflow\_1.15-py3.6-ddl 111e41b3-de2d-5422-a4d6-bf776828c4b7 base\n",

"runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base\n",

"scikit-learn\_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base\n",

"default\_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base\n",

"pytorch-onnx\_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base\n",

"kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base\n",

"pytorch-onnx\_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base\n",

"tensorflow\_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base\n",

"spark-mllib\_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base\n",

"tensorflow\_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base\n",

"runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base\n",

"do\_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base\n",

"autoai-ts\_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base\n",

"tensorflow\_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base\n",

"kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base\n",

"pytorch\_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base\n",

"spark-mllib\_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base\n",

"pytorch-onnx\_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base\n",

"spark-mllib\_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base\n",

"spark-mllib\_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base\n",

"xgboost\_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base\n",

"pytorch-onnx\_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base\n",

"default\_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base\n",

"autoai-ts\_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base\n",

"autoai-obm\_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base\n",

"pmml-3.0\_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base\n",

"spark-mllib\_2.4-r\_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base\n",

"xgboost\_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base\n",

"pytorch-onnx\_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base\n",

"autoai-ts\_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base\n",

"spark-mllib\_2.4-scala\_2.11 55a70f99-7320-4be5-9fb9-9edb5a443af5 base\n",

"spark-mllib\_3.0 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9 base\n",

"autoai-obm\_2.0 5c2e37fa-80b8-5e77-840f-d912469614ee base\n",

"spss-modeler\_18.1 5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b base\n",

"cuda-py3.8 5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e base\n",

"autoai-kb\_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base\n",

"pytorch-onnx\_1.7-py3.8 634d3cdc-b562-5bf9-a2d4-ea90a478456b base\n",

"spark-mllib\_2.3-r\_3.6 6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c base\n",

"tensorflow\_2.4-py3.7 65e171d7-72d1-55d9-8ebb-f813d620c9bb base\n",

"spss-modeler\_18.2 687eddc9-028a-4117-b9dd-e57b36f1efa5 base\n",

"----------------------------- ------------------------------------ ----\n",

"Note: Only first 50 records were displayed. To display more use 'limit' parameter.\n"

]

}

],

"source": [

"client.software\_specifications.list()"

]

},

{

"cell\_type": "code",

"execution\_count": 97,

"id": "876684d7",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'12b83a17-24d8-5082-900f-0ab31fbfd3cb'"

]

},

"execution\_count": 97,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"software\_spec\_uid = client.software\_specifications.get\_uid\_by\_name(\"runtime-22.1-py3.9\")\n",

"software\_spec\_uid"

]

},

{

"cell\_type": "code",

"execution\_count": 99,

"id": "db811976",

"metadata": {},

"outputs": [],

"source": [

"model\_details = client.repository.store\_model(model = rand\_mod, meta\_props = {\n",

" client.repository.ModelMetaNames.NAME:\"rainfall\",\n",

" client.repository.ModelMetaNames.TYPE:\"scikit-learn\_1.0\",\n",

" client.repository.ModelMetaNames.SOFTWARE\_SPEC\_UID : software\_spec\_uid\n",

"})\n",

"model\_id = client.repository.get\_model\_id(model\_details)"

]

},

{

"cell\_type": "code",

"execution\_count": 100,

"id": "1d9c35bb",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"'6d4c1451-5f18-4af5-a23f-c74cdf2a36c7'"

]

},

"execution\_count": 100,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"model\_id"

]

},

{

"cell\_type": "code",

"execution\_count": 101,

"id": "007f0730",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([5.19157088e-01, 5.39083558e-04, 3.25581395e-01, 1.53846154e-02,\n",

" 6.96969697e-01, 6.80000000e-01, 3.20338983e-01, 3.19218241e-01,\n",

" 5.19305019e-01, 0.00000000e+00])"

]

},

"execution\_count": 101,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"X\_train\_scaled[0]"

]

},

{

"cell\_type": "code",

"execution\_count": 102,

"id": "8abd4344",

"metadata": {},

"outputs": [

{

"data": {

"text/plain": [

"array([1])"

]

},

"execution\_count": 102,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"rand\_mod.predict([[5.19157088e-01, 5.39083558e-04, 3.25581395e-01, 1.53846154e-02,\n",

" 6.96969697e-01, 6.80000000e-01, 3.20338983e-01, 3.19218241e-01,\n",

" 5.19305019e-01, 0.00000000e+00]])"

]

},

{

"cell\_type": "code",

"execution\_count": null,

"id": "d87e4870",

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"kernelspec": {

"display\_name": "Python 3",

"language": "python",

"name": "python3"

},

"language\_info": {

"codemirror\_mode": {

"name": "ipython",

"version": 3

},

"file\_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert\_exporter": "python",

"pygments\_lexer": "ipython3",

"version": "3.8.8"

}

},

"nbformat": 4,

"nbformat\_minor": 5

}