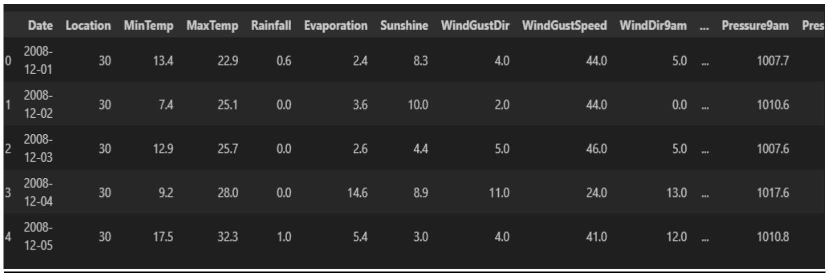
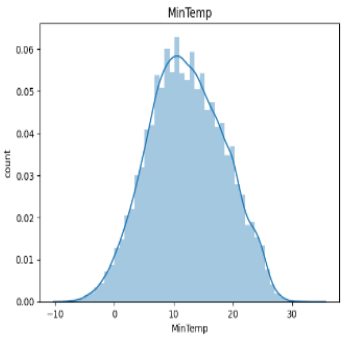
Data Collection and Preprocessing Phase

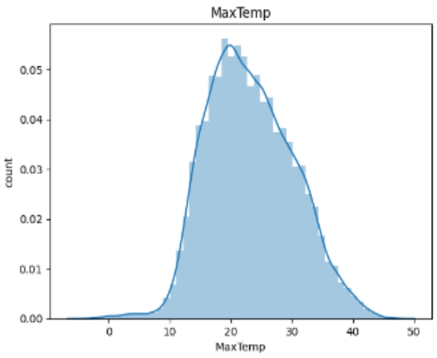
Date	02 October 2024
Team ID	LTVIP2024TMID24922
Project Title	Rainfall Prediction Using Machine Learning
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

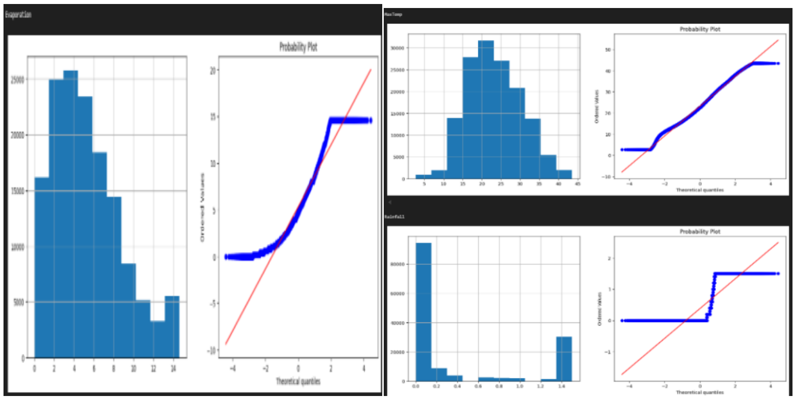
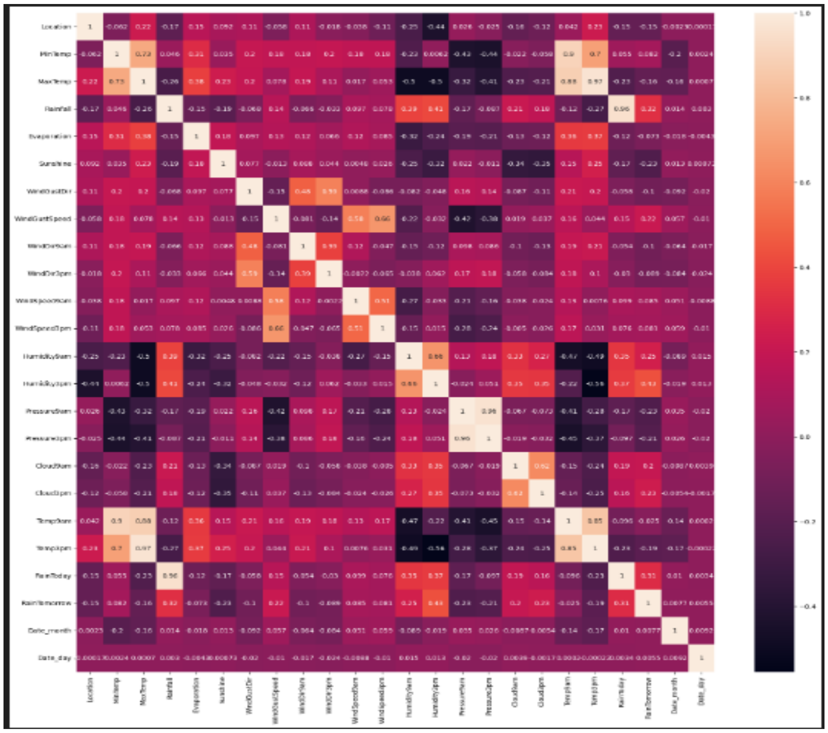
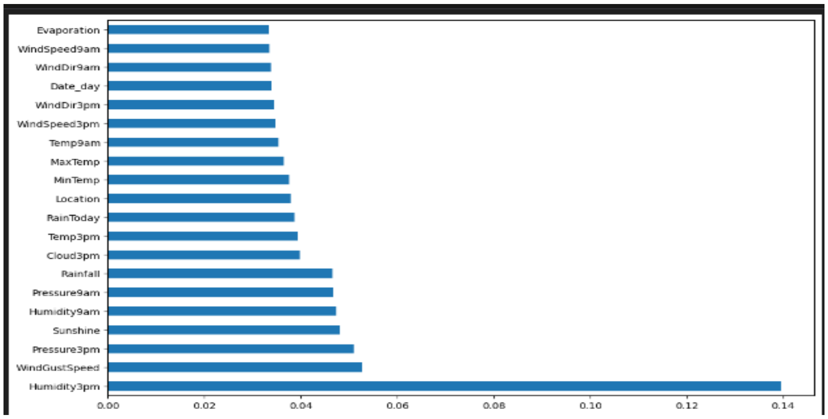
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>
Dimension:
145460 rows × 23 columns

Descriptive statistics:

<table>
<tr><th></th><th>Date</th><th>Location</th><th>MinTemp</th><th>MaxTemp</th><th>Rainfall</th><th>Evaporation</th><th>Sunshine</th><th>WindGustDir</th><th>WindGustSpeed</th><th>WindDir9am</th><th>...</th><th>Pressure9am</th><th>Pres</th></tr>
<tr><td>0</td><td>2008-12-01</td><td>30</td><td>13.4</td><td>22.9</td><td>0.6</td><td>2.4</td><td>8.3</td><td>4.0</td><td>44.0</td><td>5.0</td><td>...</td><td>1007.7</td><td></td></tr>
<tr><td>1</td><td>2008-12-02</td><td>30</td><td>7.4</td><td>25.1</td><td>0.0</td><td>3.6</td><td>10.0</td><td>2.0</td><td>44.0</td><td>0.0</td><td>...</td><td>1010.6</td><td></td></tr>
<tr><td>2</td><td>2008-12-03</td><td>30</td><td>12.9</td><td>25.7</td><td>0.0</td><td>2.6</td><td>4.4</td><td>5.0</td><td>46.0</td><td>5.0</td><td>...</td><td>1007.6</td><td></td></tr>
<tr><td>3</td><td>2008-12-04</td><td>30</td><td>9.2</td><td>28.0</td><td>0.0</td><td>14.6</td><td>8.9</td><td>11.0</td><td>24.0</td><td>13.0</td><td>...</td><td>1017.6</td><td></td></tr>
<tr><td>4</td><td>2008-12-05</td><td>30</td><td>17.5</td><td>32.3</td><td>1.0</td><td>5.4</td><td>3.0</td><td>4.0</td><td>41.0</td><td>12.0</td><td>...</td><td>1010.8</td><td></td></tr>
</table>
</td></tr>
<tr><td>Univariate Analysis</td><td>




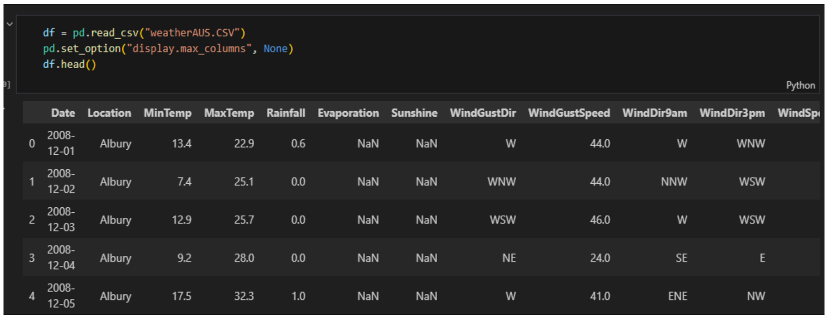
</td></tr>
</table>

Bivariate Analysis	
Multivariate Analysis	 

Outliers and Anomalies	-

Data Preprocessing Code Screenshots

Loading Data



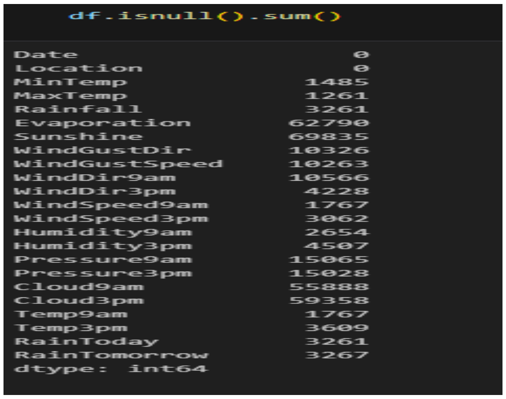
```python
df = pd.read_csv("weatherAUS.CSV")
pd.set_option("display.max_columns", None)
df.head()
```

Python

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSp
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	44.0	W	WNW	
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	44.0	NNW	WSW	
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	46.0	W	WSW	
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	24.0	SE	E	
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	41.0	ENE	NW	

Handling Missing Data

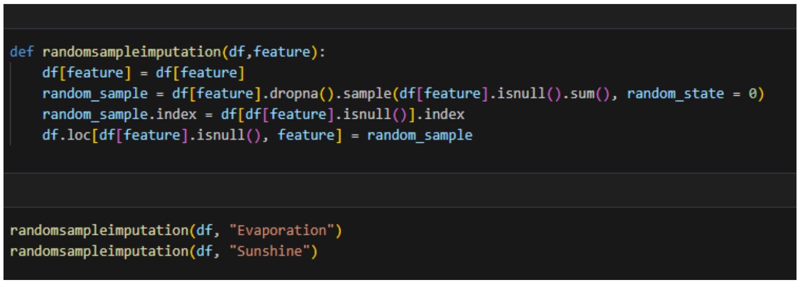
Identifying missing values.



```
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
Humidity3pm        4507
Pressure9am       15065
Pressure3pm       15028
Cloud9am          55888
Cloud3pm          59358
Temp9am            1767
Temp3pm            3609
RainToday          3261
RainTomorrow       3267
dtype: int64
```

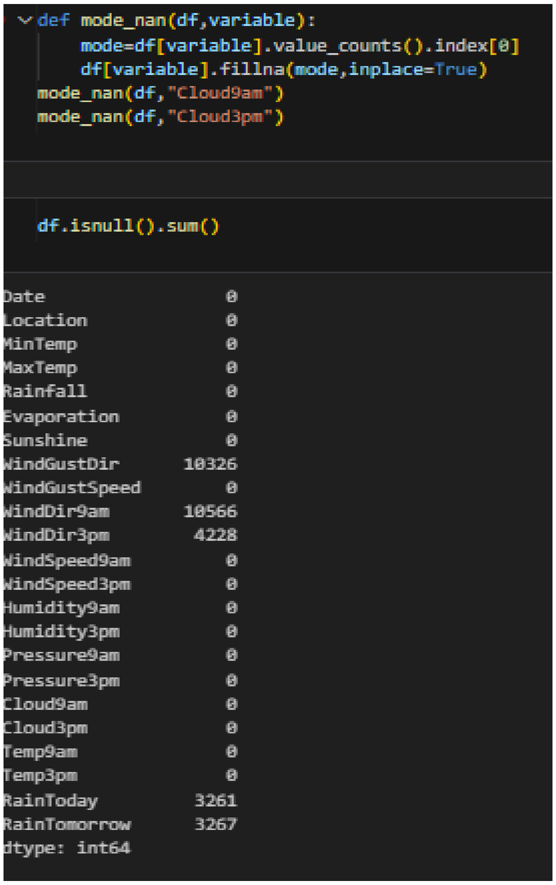
Handling missing values



```python
def randomsampleimputation(df,feature):
    df[feature] = df[feature]
    random_sample = df[feature].dropna().sample(df[feature].isnull().sum(), random_state = 0)
    random_sample.index = df[df[feature].isnull()].index
    df.loc[df[feature].isnull(), feature] = random_sample
```

```python
randomsampleimputation(df, "Evaporation")
randomsampleimputation(df, "Sunshine")
```

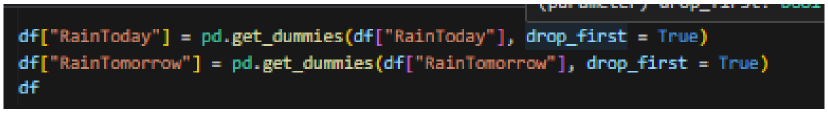
Data Transformation



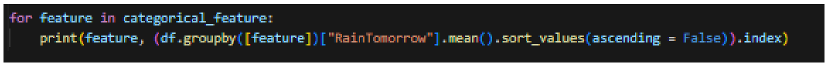
```
def mode_nan(df,variable):
    mode=df[variable].value_counts().index[0]
    df[variable].fillna(mode,inplace=True)
mode_nan(df,"Cloud9am")
mode_nan(df,"Cloud3pm")
```

```
df.isnull().sum()
```

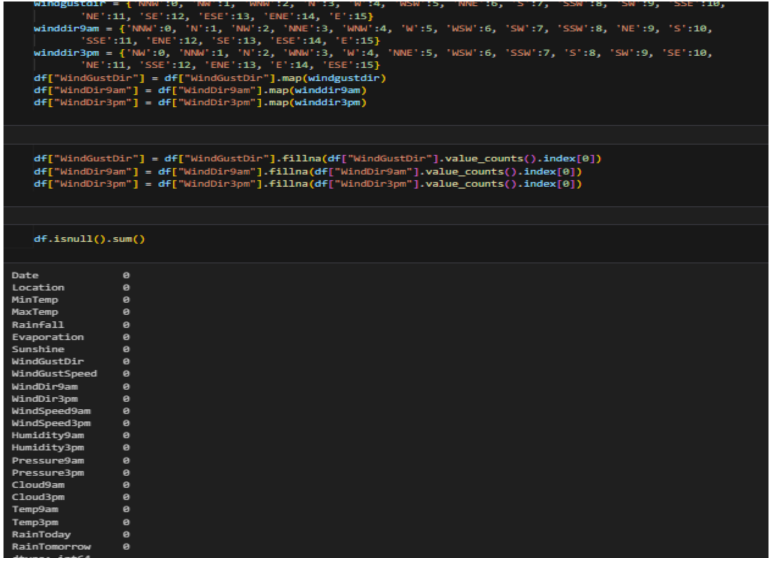
```
Date                 0
Location             0
MinTemp              0
MaxTemp              0
Rainfall             0
Evaporation          0
Sunshine             0
WindGustDir      10326
WindGustSpeed        0
WindDir9am       10566
WindDir3pm        4228
WindSpeed9am         0
WindSpeed3pm         0
Humidity9am          0
Humidity3pm          0
Pressure9am          0
Pressure3pm          0
Cloud9am             0
Cloud3pm             0
Temp9am              0
Temp3pm              0
RainToday         3261
RainTomorrow      3267
dtype: int64
```



```
(parameter) drop_first: bool
df["RainToday"] = pd.get_dummies(df["RainToday"], drop_first = True)
df["RainTomorrow"] = pd.get_dummies(df["RainTomorrow"], drop_first = True)
df
```



```
for feature in categorical_feature:
    print(feature, (df.groupby([feature])["RainTomorrow"].mean().sort_values(ascending = False)).index)
```

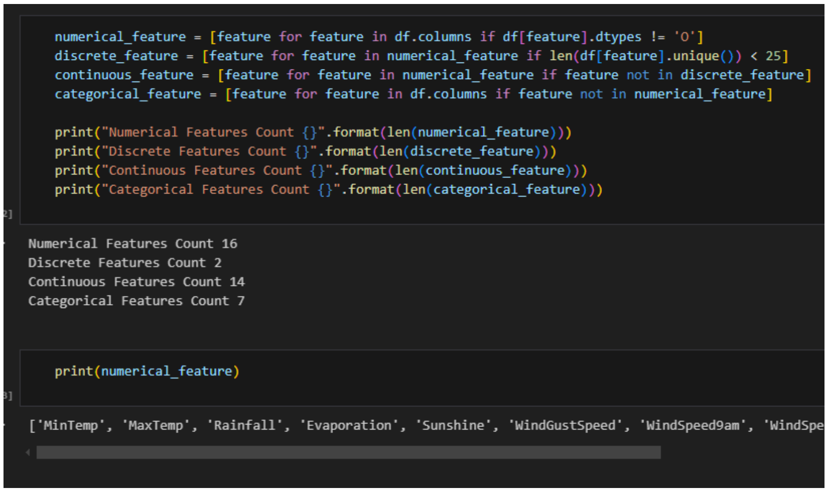
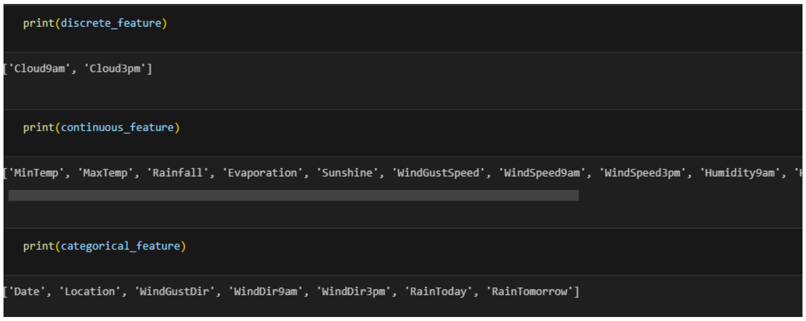


```
windgustdir = {'NNW':0, 'NW':1, 'WNW':2, 'N':3, 'W':4, 'WSW':5, 'NNE':6, 'S':7, 'SSW':8, 'SW':9, 'SSE':10,
        'NE':11, 'SE':12, 'ESE':13, 'ENE':14, 'E':15}
winddir9am = {'NNW':0, 'N':1, 'NW':2, 'NNE':3, 'WNW':4, 'W':5, 'WSW':6, 'SW':7, 'SSW':8, 'NE':9, 'S':10,
        'SSE':11, 'ENE':12, 'SE':13, 'ESE':14, 'E':15}
winddir3pm = {'NW':0, 'NNW':1, 'N':2, 'WNW':3, 'W':4, 'NNE':5, 'WSW':6, 'SSW':7, 'S':8, 'SW':9, 'SE':10,
        'NE':11, 'SSE':12, 'ENE':13, 'E':14, 'ESE':15}
df["WindGustDir"] = df["WindGustDir"].map(windgustdir)
df["WindDir9am"] = df["WindDir9am"].map(winddir9am)
df["WindDir3pm"] = df["WindDir3pm"].map(winddir3pm)
```

```
df["WindGustDir"] = df["WindGustDir"].fillna(df["WindGustDir"].value_counts().index[0])
df["WindDir9am"] = df["WindDir9am"].fillna(df["WindDir9am"].value_counts().index[0])
df["WindDir3pm"] = df["WindDir3pm"].fillna(df["WindDir3pm"].value_counts().index[0])
```

```
df.isnull().sum()
```

```
Date             0
Location         0
MinTemp          0
MaxTemp          0
Rainfall         0
Evaporation      0
Sunshine         0
WindGustDir      0
WindGustSpeed    0
WindDir9am       0
WindDir3pm       0
WindSpeed9am     0
WindSpeed3pm     0
Humidity9am      0
Humidity3pm      0
Pressure9am      0
Pressure3pm      0
Cloud9am         0
Cloud3pm         0
Temp9am          0
Temp3pm          0
RainToday        0
RainTomorrow     0
dtype: int64
```

Feature Engineering	
	`numerical_feature = [feature for feature in df.columns if df[feature].dtypes != 'O']`
	`discrete_feature = [feature for feature in numerical_feature if len(df[feature].unique()) < 25]`
	`continuous_feature = [feature for feature in numerical_feature if feature not in discrete_feature]`
	`categorical_feature = [feature for feature in df.columns if feature not in numerical_feature]`
	`print("Numerical Features Count {}".format(len(numerical_feature)))`
	`print("Discrete Features Count {}".format(len(discrete_feature)))`
	`print("Continuous Features Count {}".format(len(continuous_feature)))`
	`print("Categorical Features Count {}".format(len(categorical_feature)))`
	`Numerical Features Count 16`
	`Discrete Features Count 2`
	`Continuous Features Count 14`
	`Categorical Features Count 7`
	`print(numerical_feature)`
	`['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am', 'WindSpe`
	
	`print(discrete_feature)`
	`['Cloud9am', 'Cloud3pm']`
	`print(continuous_feature)`
	`['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'H`
	`print(categorical_feature)`
	`['Date', 'Location', 'WindGustDir', 'WindDir9am', 'WindDir3pm', 'RainToday', 'RainTomorrow']`

Save Processed Data	-