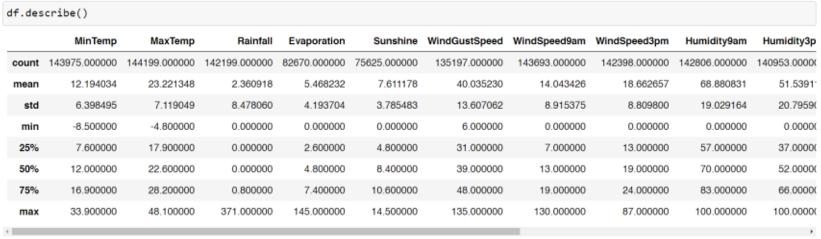
Data Exploration and Preprocessing Report

Date	15 April 2024
Team ID	Team-738164
Project Title	Rainfall Prediction Using Machine Learning
Maximum Marks	6 Marks

Data Exploration Screenshots:

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>

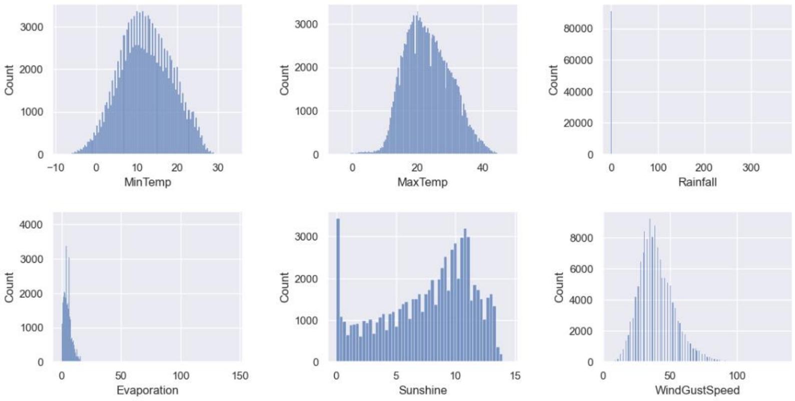
Dimensions:
145460 rows x 23 columns
Descriptive Statistics:



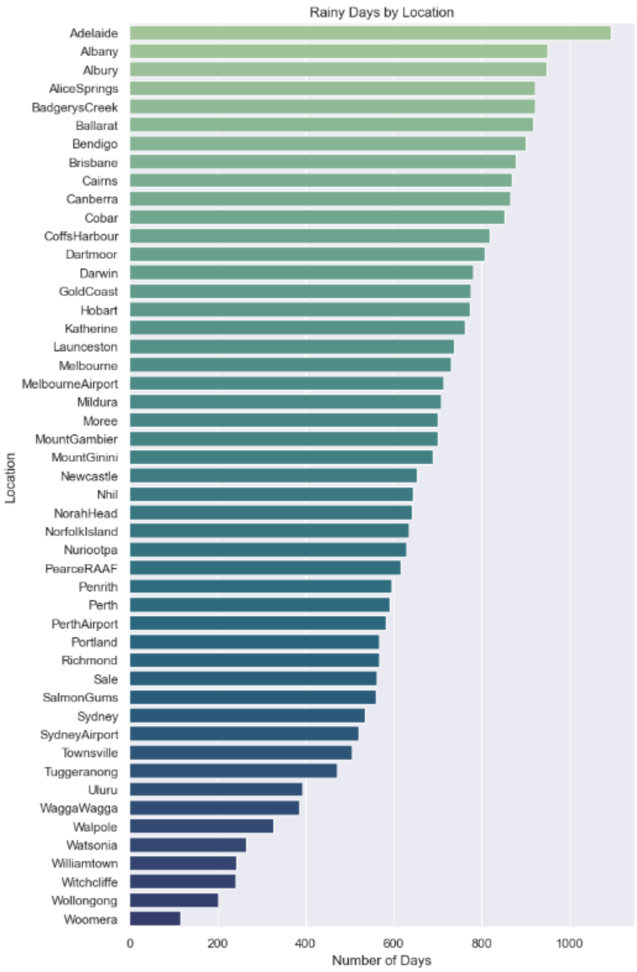
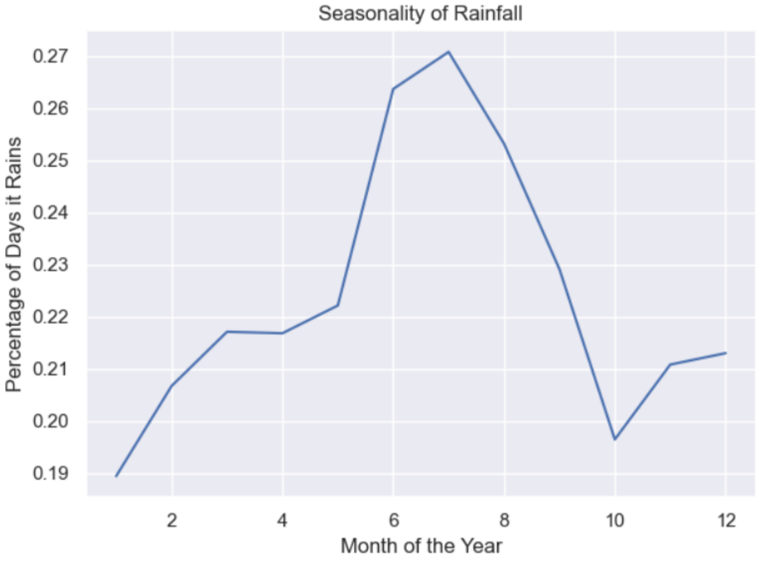
```
df.describe()
```

```
       MinTemp        MaxTemp        Rainfall       Evaporation   Sunshine      WindGustSpeed  WindSpeed9am   WindSpeed3pm   Humidity9am    Humidity3p
count  143975.000000  144199.000000  142199.000000  82670.000000  75625.000000  135197.000000  143693.000000  142398.000000  142806.000000  140953.0000(
mean   12.194034      23.221348      2.360918       5.468232      7.611178      40.035230      14.043426      18.662657      68.880831      51.5391
std    6.398495       7.119049       8.478060       4.193704      3.785483      13.607062      8.915375       8.809800       19.029164      20.7959(
min    -8.500000      -4.800000      0.000000       0.000000      0.000000      6.000000       0.000000       0.000000       0.000000       0.0000(
25%    7.600000       17.900000      0.000000       2.600000      4.800000      31.000000      7.000000       13.000000      57.000000      37.0000(
50%    12.000000      22.600000      0.000000       4.800000      8.400000      39.000000      13.000000      19.000000      70.000000      52.0000(
75%    16.900000      28.200000      0.800000       7.400000      10.600000     48.000000      19.000000      24.000000      83.000000      66.0000(
max    33.900000      48.100000      371.000000     145.000000    14.500000     135.000000     130.000000     87.000000      100.000000     100.0000(
```

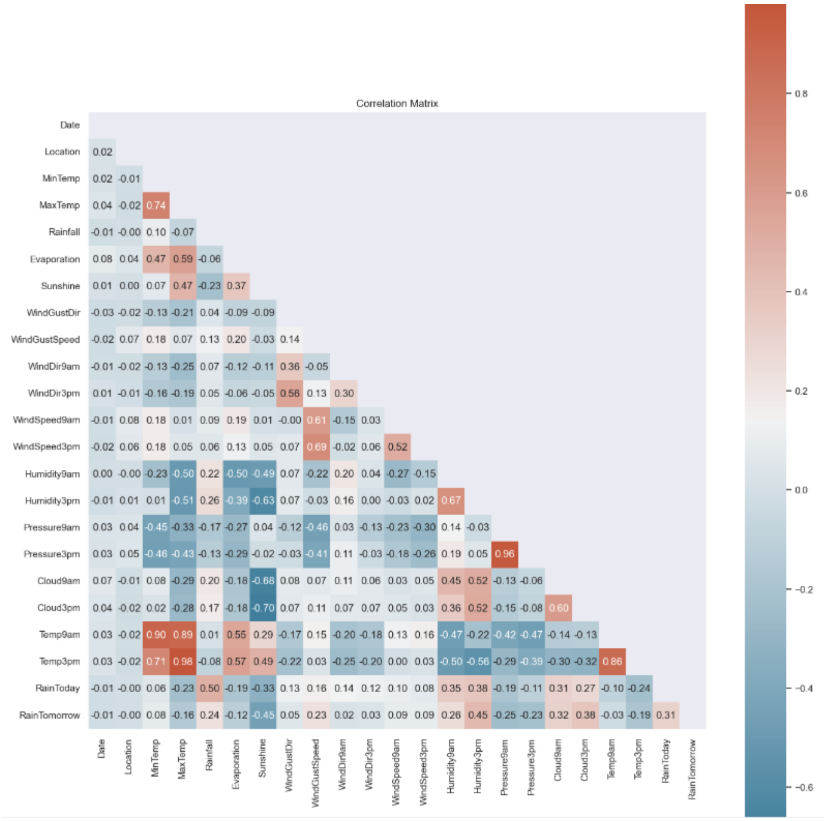
</td></tr>
<tr><td>Univariate Analysis</td><td>



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

Bivariate Analysis	 

Multivariate Analysis

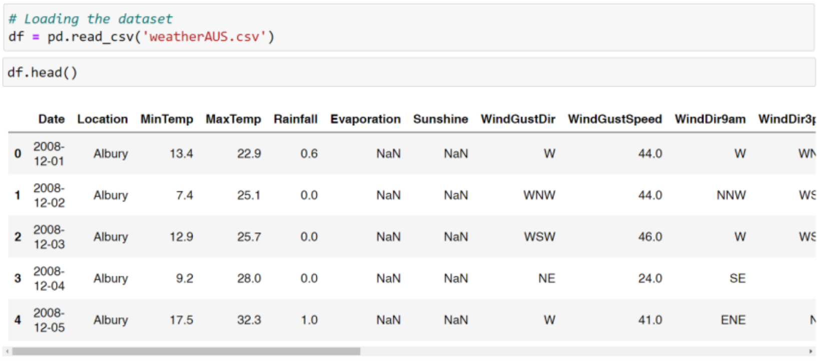


Outliers and Anomalies

1. Multiple columns have clear outliers (e.g., the max Rainfall value is 371.0 despite the 75th percentile being 0.8)
2. Not seeing any values that are immediate cause for concern (such as a negative value for minimum Rainfall)

Data Preprocessing Code Screenshots

Loading Data

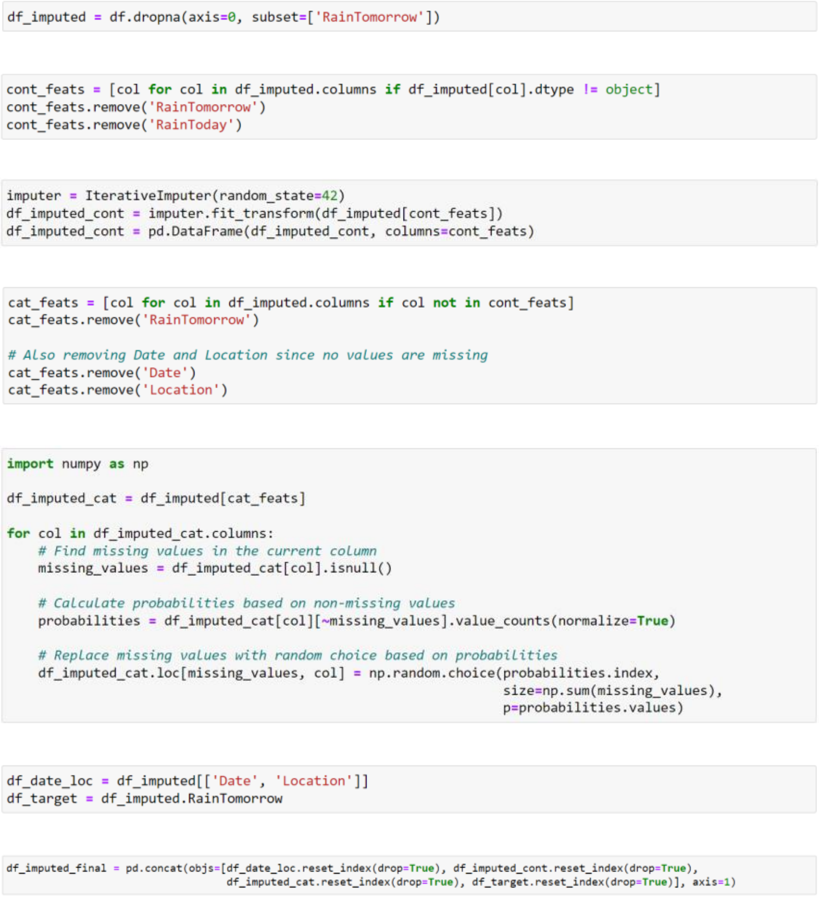


```
# Loading the dataset
df = pd.read_csv('weatherAUS.csv')
```

```
df.head()
```

```
   Date   Location  MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustDir  WindGustSpeed  WindDir9am  WindDir3
0  2008-  Albury       13.4     22.9       0.6          NaN       NaN            W           44.0           W        WN
   12-01
1  2008-  Albury        7.4     25.1       0.0          NaN       NaN          WNW           44.0         NNW        WS
   12-02
2  2008-  Albury       12.9     25.7       0.0          NaN       NaN          WSW           46.0           W        WS
   12-03
3  2008-  Albury        9.2     28.0       0.0          NaN       NaN           NE           24.0          SE
   12-04
4  2008-  Albury       17.5     32.3       1.0          NaN       NaN            W           41.0         ENE         N
   12-05
```

Handling Missing Data



```
df_imputed = df.dropna(axis=0, subset=['RainTomorrow'])
```

```
cont_feats = [col for col in df_imputed.columns if df_imputed[col].dtype != object]
cont_feats.remove('RainTomorrow')
cont_feats.remove('RainToday')
```

```
imputer = IterativeImputer(random_state=42)
df_imputed_cont = imputer.fit_transform(df_imputed[cont_feats])
df_imputed_cont = pd.DataFrame(df_imputed_cont, columns=cont_feats)
```

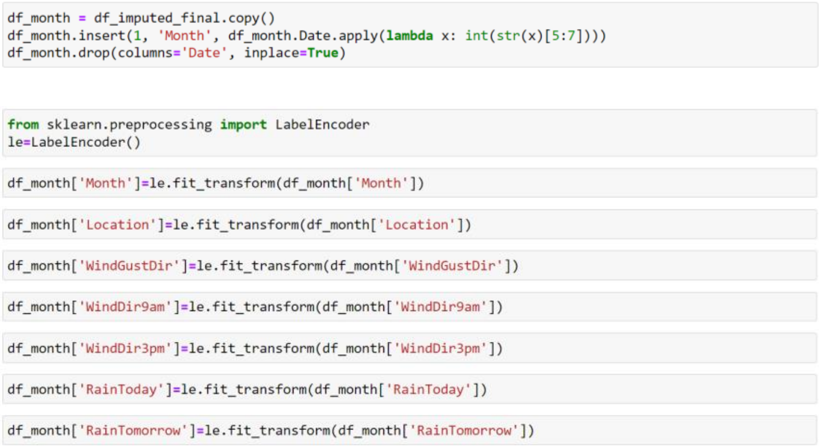
```
cat_feats = [col for col in df_imputed.columns if col not in cont_feats]
cat_feats.remove('RainTomorrow')

# Also removing Date and Location since no values are missing
cat_feats.remove('Date')
cat_feats.remove('Location')
```

```
import numpy as np

df_imputed_cat = df_imputed[cat_feats]

for col in df_imputed_cat.columns:
    # Find missing values in the current column
    missing_values = df_imputed_cat[col].isnull()

    # Calculate probabilities based on non-missing values
    probabilities = df_imputed_cat[col][~missing_values].value_counts(normalize=True)

    # Replace missing values with random choice based on probabilities
    df_imputed_cat.loc[missing_values, col] = np.random.choice(probabilities.index,
                                                               size=np.sum(missing_values),
                                                               p=probabilities.values)
```

```
df_date_loc = df_imputed[['Date', 'Location']]
df_target = df_imputed.RainTomorrow
```

```
df_imputed_final = pd.concat(objs=[df_date_loc.reset_index(drop=True), df_imputed_cont.reset_index(drop=True),
                                   df_imputed_cat.reset_index(drop=True), df_target.reset_index(drop=True)], axis=1)
```

Data Transformation



```
df_month = df_imputed_final.copy()
df_month.insert(1, 'Month', df_month.Date.apply(lambda x: int(str(x)[5:7])))
df_month.drop(columns='Date', inplace=True)
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
```

```
df_month['Month']=le.fit_transform(df_month['Month'])
```

```
df_month['Location']=le.fit_transform(df_month['Location'])
```

```
df_month['WindGustDir']=le.fit_transform(df_month['WindGustDir'])
```

```
df_month['WindDir9am']=le.fit_transform(df_month['WindDir9am'])
```

```
df_month['WindDir3pm']=le.fit_transform(df_month['WindDir3pm'])
```

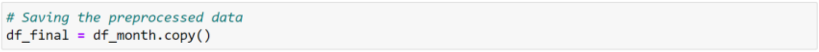
```
df_month['RainToday']=le.fit_transform(df_month['RainToday'])
```

```
df_month['RainTomorrow']=le.fit_transform(df_month['RainTomorrow'])
```

Feature Engineering

Attached the codes in final submission.

Save Processed Data



```
# Saving the preprocessed data
df_final = df_month.copy()
```