Final Project Submission

Please fill out:

- Student name: Justin Grisanti
- Student pace: self-paced
- Scheduled project review date/time: 1/5/2022 @ 2pm
- Instructor name: Claude Fried
- Blog post URL: https://justingrisanti.github.io/predicting_rain_patterns_in_australia

Total Time to Run: approx. 27 minutes

Section 1: Business Understanding

The purpose of this section is to define the business problem and understand the stakeholders for the work that I am performing. The Bureau of Meteorology is responsible for predicting weather patterns throughout the entire Australian region. According to their website, their forecast accuracy for rain varies much more than their forecasts for temperature and wind.

According to their analyses, they've underpredicted rainfall each year for the past five years. The goal is to create a classification model that allows the Bureau of Meteorology to improve their predictions of whether or not it will rain the next day. This will allow them to inform the public better so that citizens can prepare accordingly for the possibility of rain.

The stakeholders of this project are the Bureau of Meteorology and citizens of Australia.

The main purpose of this classification model is predictive, meaning that given charactaristics of rain data on a given day, the model should be able to predict whether it will rain the next day or not. My model is not meant to replace the Bureau of Meteorology's current system of predicting rain for the region of Australia, however, it is meant serve as an input to strengthen their predictions and assumptions, and to reduce the risk of failing to predict that it will rain the next day.

Section 2: Data Understanding

After scanning the data, we have weather-related information for a period of 12/1/2008 to 6/25/2017—8 years, 6 months and 24 days worth of data.

As shown below, this data has many different weather-related metrics, such as the wind speed, humidity, pressure, whether it was sunny or cloudy, and temperature. These seem like appropriate parameters to run a classification-based model in order to predict whether or not it will rain the next day, and I do not sense any limitations from this data.

In [134…

```
# Import Libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import tree
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, GridSearchCV, cross_v
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.metrics import accuracy_score
from sklearn.impute import SimpleImputer
from sklearn.metrics import make_scorer, classification_report, log_loss, f
from sklearn.model_selection import StratifiedKFold, cross_val_score
from sklearn.base import clone
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
import sklearn.preprocessing as preprocessing
from scipy import stats
import seaborn as sns
import math
from imblearn.over_sampling import SMOTE
import statsmodels.api as sm
from statsmodels.sandbox.predict_functional import predict_functional
sns.set_context("talk")
sns.set_theme(style='darkgrid')
import warnings
warnings.filterwarnings('ignore')
```

In [4]:

```
# Import data from csv

rain_data = pd.read_csv('data/WeatherAUS.csv')
```

In [5]:

```
rain_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145460 entries, 0 to 145459
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           145460 non-null  object 
 1   Location       145460 non-null  object 
 2   MinTemp        143975 non-null  float64
 3   MaxTemp        144199 non-null  float64
 4   Rainfall       142199 non-null  float64
 5   Evaporation    82670 non-null   float64
 6   Sunshine       75625 non-null   float64
 7   WindGustDir    135134 non-null  object 
 8   WindGustSpeed  135197 non-null  float64
 9   WindDir9am     134894 non-null  object 
 10  WindDir3pm     141232 non-null  object 
 11  WindSpeed9am   143693 non-null  float64
 12  WindSpeed3pm   142398 non-null  float64
 13  Humidity9am    142806 non-null  float64
 14  Humidity3pm    140953 non-null  float64
 15  Pressure9am    130395 non-null  float64
 16  Pressure3pm    130432 non-null  float64
 17  Cloud9am       89572 non-null   float64
 18  Cloud3pm       86102 non-null   float64
 19  Temp9am        143693 non-null  float64
 20  Temp3pm        141851 non-null  float64
 21  RainToday      142199 non-null  object 
 22  RainTomorrow   142193 non-null  object 
dtypes: float64(16), object(7)
memory usage: 25.5+ MB
```

Please see the following column descriptions:

> **MinTemp:** The minimum temperature in degrees celsius
>
> **MaxTemp:** The maximum temperature in degrees celsius
>
> **Rainfall:** The amount of rainfall recorded for the day in mm
>
> **Evaporation:** The so-called Class A pan evaporation (mm) in the 24 hours to 9am
>
> **Sunshine:** The number of hours of bright sunshine in the day.
>
> **WindGustDir:** The direction of the strongest wind gust in the 24 hours to midnight
>
> **WindGustSpeed:** The speed (km/h) of the strongest wind gust in the 24 hours to midnight

> **WindDir9am:** Direction of the wind at 9am
>
> **WindDir3pm:** Direction of the wind at 3pm
>
> **WindSpeed9am:** Wind speed (km/hr) averaged over 10 minutes prior to 9am
>
> **WindSpeed3pm:** Wind speed (km/hr) averaged over 10 minutes prior to 3pm
>
> **Humidity9am:** Humidity (percent) at 9am
>
> **Humidity3pm:** Humidity (percent) at 3pm
>
> **Pressure9am:** Atmospheric pressure (hpa) reduced to mean sea level at 9am
>
> **Pressure3pm:** Atmospheric pressure (hpa) reduced to mean sea level at 3pm
>
> **Cloud9am:** Fraction of sky obscured by cloud at 9am. This is measured in "oktas", which are a unit of eigths. It records how many eigths of the sky are obscured by cloud. A 0 measure indicates completely clear sky whilst an 8 indicates that it is completely overcast.
>
> **Cloud3pm:** Fraction of sky obscured by cloud (in "oktas": eighths) at 3pm. See Cload9am for a description of the values
>
> **Temp9am:** Temperature (degrees C) at 9am
>
> **Temp3pm:** Temperature (degrees C) at 3pm
>
> **RainToday:** Boolean: 1 if precipitation (mm) in the 24 hours to 9am exceeds 1mm, otherwise 0
>
> **RainTomorrow:** The amount of next day rain in mm. Used to create response variable RainTomorrow. A kind of measure of the "risk".

In [6]:

```
# To show all columns and rows

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)

rain_data.head()
```

Out[6]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	Winc
0	2008-12-01	Albury	13.4	22.9	0.6	NaN	NaN	W	
1	2008-12-02	Albury	7.4	25.1	0.0	NaN	NaN	WNW	
2	2008-12-03	Albury	12.9	25.7	0.0	NaN	NaN	WSW	
3	2008-12-04	Albury	9.2	28.0	0.0	NaN	NaN	NE	
4	2008-12-05	Albury	17.5	32.3	1.0	NaN	NaN	W	

In [7]:

```
rain_data[rain_data['RainTomorrow'].isna()].head()
```

Out[7]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	W
14	2008-12-15	Albury	8.4	24.6	0.0	NaN	NaN	NaN	
283	2009-09-10	Albury	2.6	NaN	0.0	NaN	NaN	NaN	
435	2010-02-09	Albury	22.1	35.1	0.0	NaN	NaN	NaN	
437	2010-02-11	Albury	21.5	35.0	0.0	NaN	NaN	NaN	
443	2010-02-17	Albury	15.5	30.6	0.0	NaN	NaN	NaN	

In [8]:

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

In [9]:

```
rain_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 142193 entries, 0 to 145458
Data columns (total 23 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           142193 non-null  object 
 1   Location       142193 non-null  object 
 2   MinTemp        141556 non-null  float64
 3   MaxTemp        141871 non-null  float64
 4   Rainfall       140787 non-null  float64
 5   Evaporation    81350 non-null   float64
 6   Sunshine       74377 non-null   float64
 7   WindGustDir    132863 non-null  object 
 8   WindGustSpeed  132923 non-null  float64
 9   WindDir9am     132180 non-null  object 
 10  WindDir3pm     138415 non-null  object 
 11  WindSpeed9am   140845 non-null  float64
 12  WindSpeed3pm   139563 non-null  float64
 13  Humidity9am    140419 non-null  float64
 14  Humidity3pm    138583 non-null  float64
 15  Pressure9am    128179 non-null  float64
 16  Pressure3pm    128212 non-null  float64
 17  Cloud9am       88536 non-null   float64
 18  Cloud3pm       85099 non-null   float64
 19  Temp9am        141289 non-null  float64
 20  Temp3pm        139467 non-null  float64
 21  RainToday      140787 non-null  object 
 22  RainTomorrow   142193 non-null  object 
dtypes: float64(16), object(7)
memory usage: 26.0+ MB
```

In [10]:

```
# Separate target variable from data and complete train-test split

y = rain_data['RainTomorrow']
X = rain_data.drop('RainTomorrow', axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42)
```

In [11]:

```
# Reset index and drop it

X_train.reset_index(inplace=True)
X_train = X_train.drop(columns=['index'],axis=1)
X_train.head()
```

Out[11]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	W
0	2009-04-12	Woomera	14.9	30.3	0.0	7.4	10.9	S	
1	2014-12-08	Witchcliffe	14.6	21.5	0.2	NaN	NaN	SSE	
2	2015-06-06	SalmonGums	9.0	23.7	0.0	NaN	NaN	W	
3	2014-01-09	Albany	15.3	24.0	0.0	8.2	12.1	NaN	
4	2014-12-14	Mildura	17.3	37.5	0.0	8.6	11.4	N	

In [12]:

```
# Reset index and drop it

X_test.reset_index(inplace=True)
X_test = X_test.drop(columns=['index'],axis=1)
X_test.head()
```

Out[12]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	W
0	2016-06-09	Ballarat	7.1	13.0	8.8	NaN	NaN	N	
1	2009-10-24	Walpole	13.2	18.3	0.0	NaN	NaN	E	
2	2015-09-21	PerthAirport	9.2	22.7	0.0	5.0	11.1	ENE	
3	2011-12-06	Cobar	15.3	26.1	0.0	10.4	NaN	E	
4	2014-03-15	Sale	11.9	31.8	0.0	5.0	4.1	NW	

In [13]:

```
X_train = X_train.replace('Yes', 1.0)
X_train = X_train.replace('No', 0.0)
X_test = X_test.replace('Yes', 1.0)
X_test = X_test.replace('No', 0.0)
y_train = y_train.replace('Yes', 1.0)
y_train = y_train.replace('No', 0.0)
y_test = y_test.replace('Yes', 1.0)
y_test = y_test.replace('No', 0.0)
```

In [14]:
```
# Inspect Target Variable

y_train.value_counts(normalize=True)
```

Out[14]:
```
0.0    0.775412
1.0    0.224588
Name: RainTomorrow, dtype: float64
```

In [15]:
```
# Did not need to stratify as the train-test split matched the population

y_test.value_counts(normalize=True)
```

Out[15]:
```
0.0    0.77704
1.0    0.22296
Name: RainTomorrow, dtype: float64
```

In [16]:
```
# Ensure we get a fair spread of data across the country

X_train['Location'].value_counts()
```

Out[16]:

```
Canberra            2578
Sydney              2429
Hobart              2416
Perth               2408
Brisbane            2401
Darwin              2382
Adelaide            2329
Tuggeranong         2298
Mildura             2295
Bendigo             2279
Launceston          2279
PerthAirport        2269
Woomera             2266
Ballarat            2262
Albany              2255
MountGambier        2254
Townsville          2248
CoffsHarbour        2247
MelbourneAirport    2246
Watsonia            2243
Sale                2241
Witchcliffe         2240
GoldCoast           2240
AliceSprings        2237
Cobar               2236
NorfolkIsland       2236
Portland            2236
WaggaWagga          2236
Albury              2234
Penrith             2233
BadgerysCreek       2232
Cairns              2232
Newcastle           2229
Nuriootpa           2224
SydneyAirport       2224
Wollongong          2213
SalmonGums          2206
Richmond            2201
Dartmoor            2200
MountGinini         2192
NorahHead           2179
Moree               2134
Walpole             2133
PearceRAAF          2087
Williamtown         1905
Melbourne           1834
Katherine           1193
Uluru               1138
Nhil                1135
Name: Location, dtype: int64
```

In [17]:

```
# Plot each column in a histogram to see what type of distribution there is
columns_with_nulls = X_train.drop(['Location','Date'], axis=1)
columns_with_nulls.hist(figsize=(20,20))
plt.savefig('Visualizations/ColumnsHist.png', bbox_inches = 'tight')
```

In [18]:

```
X_train.info(1, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106644 entries, 0 to 106643
Data columns (total 22 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           106644 non-null  object 
 1   Location       106644 non-null  object 
 2   MinTemp        106150 non-null  float64
 3   MaxTemp        106387 non-null  float64
 4   Rainfall       105540 non-null  float64
 5   Evaporation    60909 non-null   float64
 6   Sunshine       55677 non-null   float64
 7   WindGustDir    99665 non-null   object 
 8   WindGustSpeed  99715 non-null   float64
 9   WindDir9am     99120 non-null   object 
 10  WindDir3pm     103822 non-null  object 
 11  WindSpeed9am   105646 non-null  float64
 12  WindSpeed3pm   104681 non-null  float64
 13  Humidity9am    105326 non-null  float64
 14  Humidity3pm    103917 non-null  float64
 15  Pressure9am    96098 non-null   float64
 16  Pressure3pm    96123 non-null   float64
 17  Cloud9am       66285 non-null   float64
 18  Cloud3pm       63757 non-null   float64
 19  Temp9am        105967 non-null  float64
 20  Temp3pm        104579 non-null  float64
 21  RainToday      105540 non-null  float64
dtypes: float64(17), object(5)
memory usage: 17.9+ MB
```

As we can see, Evaporation, Sunshine, and cloud data all have a large amount of nulls. Let's take a deeper look into these variables.

In [19]:

```
X_train.head()
```

Out[19]:

	Date	Location	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	\
0	2009-04-12	Woomera	14.9	30.3	0.0	7.4	10.9	S	
1	2014-12-08	Witchcliffe	14.6	21.5	0.2	NaN	NaN	SSE	
2	2015-06-06	SalmonGums	9.0	23.7	0.0	NaN	NaN	W	
3	2014-01-09	Albany	15.3	24.0	0.0	8.2	12.1	NaN	
4	2014-12-14	Mildura	17.3	37.5	0.0	8.6	11.4	N	

The following code is to create meaningful information about our null values, so that we can impute them in a more educated manner.

In [20]:
```
# Dropping NAs to obtain distribution for data that isn't NA

rain_data_cloud_dropna = X_train.dropna(axis=0, subset=['Cloud3pm','Cloud9a
```

In [21]:
```
# Dropping NAs to obtain distribution for data that isn't NA

rain_data_sunshine_dropna = X_train.dropna(axis=0, subset=['Sunshine'])
```

In [22]:
```
# Checking null sunshine data against rain data to see what the population

rain_data_sunshine_nulls = X_train[X_train['Sunshine'].isna()]
rain_data_sunshine_nulls['RainToday'].value_counts()
```

Out[22]:
```
0.0    39031
1.0    11143
Name: RainToday, dtype: int64
```

In [23]:
```
# Removing nulls from sunshine data and biforcating the population to get m

sunshine_when_rain = rain_data_sunshine_dropna[rain_data_sunshine_dropna['R
sunshine_no_rain = rain_data_sunshine_dropna[rain_data_sunshine_dropna['Rai
```

In [24]:
```
# Checking evaporation mean when sunshine is greater than zero

evaporation_test = X_train.loc[X_train['Sunshine']>0.0,'Evaporation']
```

In [25]:
```
# Checking the mean of cloud data when it doesn't rain and humidity is high
# (Population is not normal without humidity check)

cloud9_no_rain_lower_humidity = X_train.loc[(X_train['RainToday']==0) & (X_
cloud9_no_rain_higher_humidity = X_train.loc[(X_train['RainToday']==0) & (X
cloud3_no_rain_lower_humidity = X_train.loc[(X_train['RainToday']==0) & (X_
cloud3_no_rain_higher_humidity = X_train.loc[(X_train['RainToday']==0) & (X
```

In [26]:
```
# Checking the mean of cloud data when it does rain

cloud9_rain = X_train.loc[X_train['RainToday']==1,'Cloud9am']
cloud3_rain = X_train.loc[X_train['RainToday']==1,'Cloud3pm']
```

In [27]:

```
# Checking null data to see how many records have both null cloud data and

test = []

for index in range(0,106643,1):
    if pd.isna(X_train['Cloud3pm'].loc[index]) and pd.isna(X_train['Cloud9a
        if pd.notna(X_train['Sunshine'].loc[index]):
            test.append('Sunshine')
        elif pd.isna(X_train['Sunshine'].loc[index]):
            test.append('Neither')
        else:
            pass
    elif pd.notna(X_train['Cloud3pm'].loc[index]) or pd.notna(X_train['Clou
        if pd.notna(X_train['Sunshine'].loc[index]):
            test.append('Both')
        elif pd.isna(X_train['Sunshine'].loc[index]):
            test.append('Clouds')
        else:
            pass
    else:
        pass
```

In [28]:

```
print("There are " + str(test.count('Sunshine')) + " records of sunshine da
print("There are " + str(test.count('Clouds')) + " records of cloud data wi
print("There are " + str(test.count('Neither')) + " records of neither suns
print("There are " + str(test.count('Both')) + " records of both sunshine a
```

```
There are 5580 records of sunshine data with no cloud data.
There are 19220 records of cloud data with no sunshine data.
There are 31746 records of neither sunshine or cloud data.
There are 50097 records of both sunshine and cloud data.
```

In [29]:

```
test_data = {'Sunshine data with Cloud Nulls': 8258, 'Cloud Data with Sunsh
```

In [30]:
```
# Checking the records to see the relationship between cloud data and sunsh

test2 = []

for index in range(0,106643,1):
    if X_train['Cloud3pm'].loc[index]== 0  or X_train['Cloud9am'].loc[index
        if X_train['Sunshine'].loc[index] != 0:
            test2.append('Sunshine')
        elif X_train['Sunshine'].loc[index]==0:
            test2.append('Neither')
        else:
            pass
    elif X_train['Cloud3pm'].loc[index]!=0 or X_train['Cloud9am'].loc[index
        if X_train['Sunshine'].loc[index]!=0:
            test2.append('Both')
        elif X_train['Sunshine'].loc[index]==0:
            test2.append('Clouds')
        else:
            pass
    else:
        pass
```

In [31]:
```
print("There are " + str(test2.count('Sunshine')) + " records of sunshine d
print("There are " + str(test2.count('Clouds')) + " records of cloud data w
print("There are " + str(test2.count('Neither')) + " records of no sunshine
print("There are " + str(test2.count('Both')) + " records of sunshine and c
```

```
There are 7701 records of sunshine data with 0 cloud coverage.
There are 1712 records of cloud data with 0 sunshine hours.
There are 4 records of no sunshine or cloud coverage.
There are 97226 records of sunshine and cloud coverage.
```

In [32]:
```
test2_data = {'Sunshine with 0 Clouds': 10350, 'Clouds with 0 sunshine': 23
```

In [33]:
```
X_train['RainToday'].value_counts(normalize=True)
```

Out[33]:
```
0.0    0.775687
1.0    0.224313
Name: RainToday, dtype: float64
```

In [34]:
```
rain_data_today = {'Did Not Rain':113580,'Rained':31880}
```

In [35]:

```
# Create subplot for all of our tests/findings

fig, (ax1, ax2, ax3, ax4, ax5, ax6, ax7, ax8) = plt.subplots(8, 1, figsize=
fig.suptitle('Plots of Cloud data and Sunshine Data')
fig.tight_layout(pad=5.0)

# This is a distribution for sunshine data that eliminates cloud nulls.
ax1.set_title('Chart 1: Dist. of Sunshine Data w/o cloud nulls')
ax1.hist(rain_data_cloud_dropna['Sunshine'])
ax1.set_xlabel('Hours of Sunshine')
ax1.set_ylabel('Sunshine Data')

# This is a distribution for cloud data that eliminates sunshine nulls. It
# hard to impute these values without further biforcation.
ax2.set_title('Chart 2: Dist. of 9am Cloud Data w/o Sunshine nulls')
ax2.hist(rain_data_sunshine_dropna['Cloud9am'])
ax2.set_xlabel('9am Cloud Coverage (0-8)')
ax2.set_ylabel('Cloud Data')

# This is a distribution for cloud data that eliminates sunshine nulls. It
# hard to impute these values without further biforcation.
ax3.set_title('Chart 3: Dist. of 3pm Cloud Data w/o Sunshine nulls')
ax3.hist(rain_data_sunshine_dropna['Cloud3pm'])
ax3.set_xlabel('3pm Cloud Coverage (0-8)')
ax3.set_ylabel('Cloud Data')

# Here we plot the population of nulls and their relationships. Perhaps the
# clouds and sunshine that we can impute. When both records are null, this
ax4.set_title('Chart 4: # of non-nulls by Category (Sunshine vs Cloud Data)
ax4.bar(test_data.keys(),test_data.values())
ax4.set_xlabel('Type')

# Checking the data for zeros.
ax5.set_title('Chart 5: Non-Zero Data by Category')
ax5.bar(test2_data.keys(),test2_data.values())
ax5.set_xlabel('Type')

# Here we find the distribution for days it does rain, and the relevant mea
ax6.set_title('Chart 6: Sunshine Data On Day it Rains')
ax6.hist(sunshine_when_rain['Sunshine'],bins=14)

# Here we find the distribution for days it doesn't rain, and the relevant
ax7.set_title("Chart 7: Sunshine Data On Day it Doesn't Rain")
ax7.hist(sunshine_no_rain['Sunshine'])

# Here is the distribution for evaporation when sunshine is greater than ze
ax8.set_title("Chart 8: Evaporation Data when Sunshine > 0")
ax8.hist(evaporation_test,bins=100)
plt.savefig('Visualizations/Subplot1.png', bbox_inches = 'tight')
plt.show()
```

Plots of Cloud data and Sunshine Data

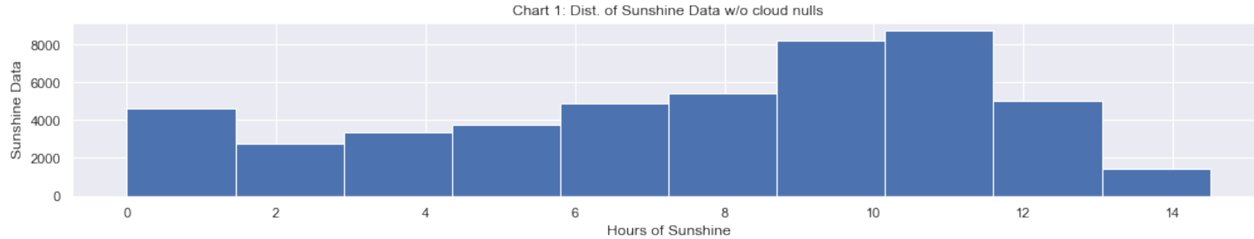


Chart 1: Dist. of Sunshine Data w/o cloud nulls
Sunshine Data
Hours of Sunshine

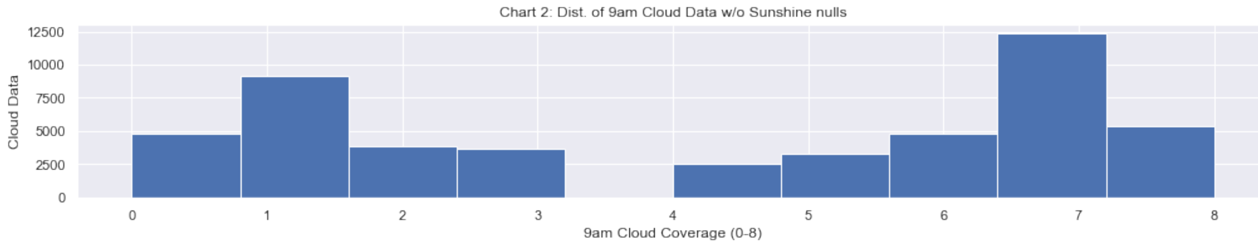


Chart 2: Dist. of 9am Cloud Data w/o Sunshine nulls
Cloud Data
9am Cloud Coverage (0-8)

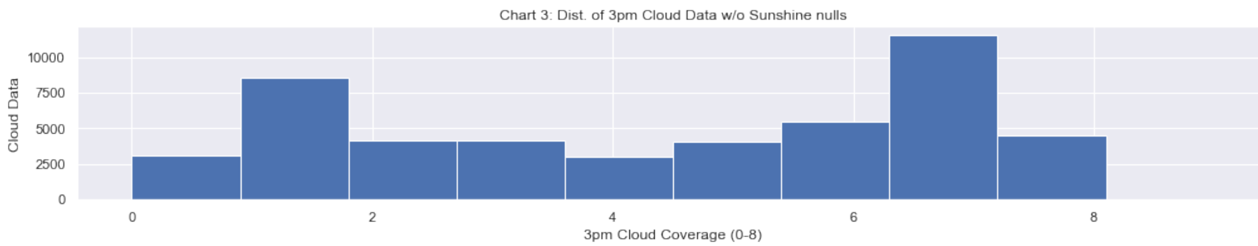


Chart 3: Dist. of 3pm Cloud Data w/o Sunshine nulls
Cloud Data
3pm Cloud Coverage (0-8)

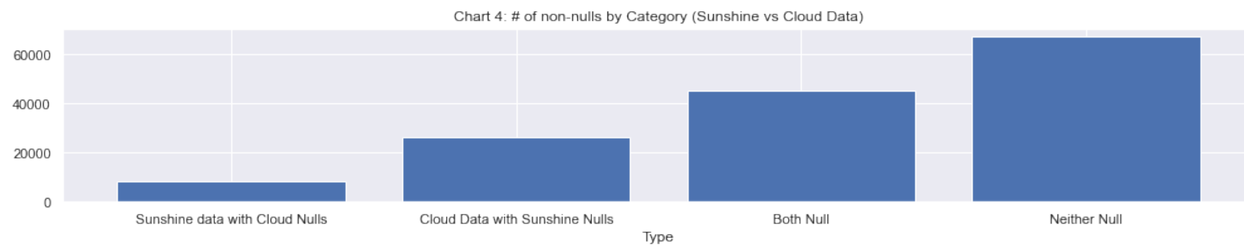


Chart 4: # of non-nulls by Category (Sunshine vs Cloud Data)
Sunshine data with Cloud Nulls
Cloud Data with Sunshine Nulls
Both Null
Neither Null
Type

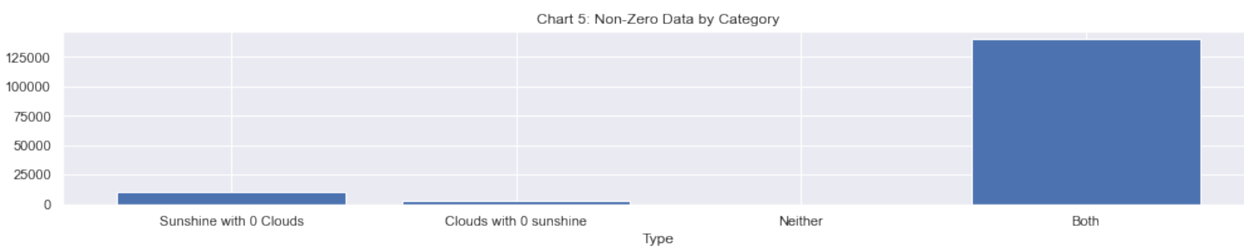


Chart 5: Non-Zero Data by Category
Sunshine with 0 Clouds
Clouds with 0 sunshine
Neither
Both
Type

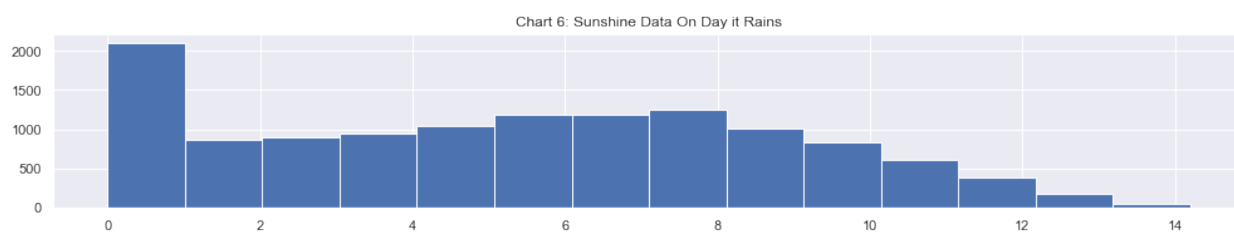


Chart 6: Sunshine Data On Day it Rains

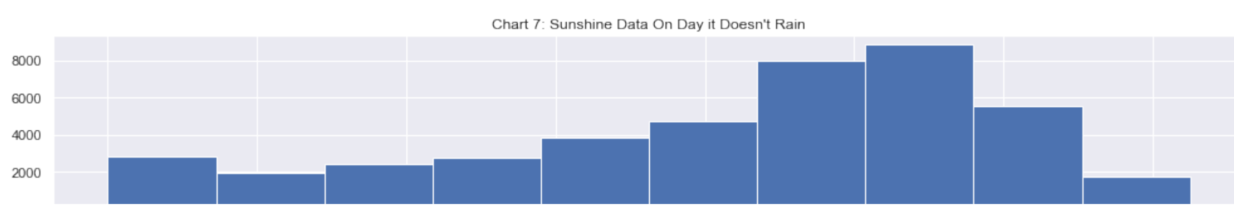
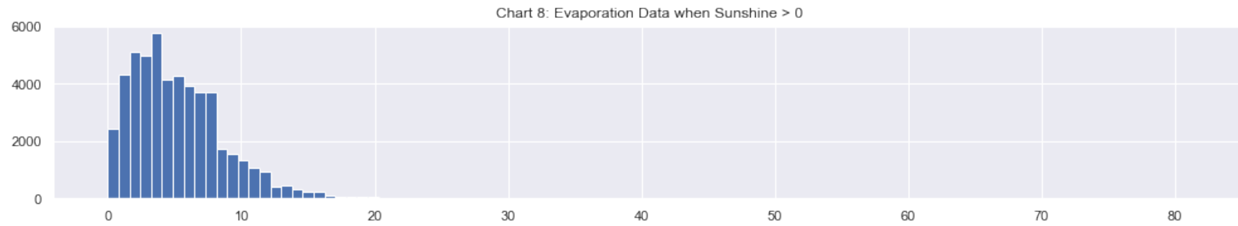


Chart 7: Sunshine Data On Day it Doesn't Rain



Here are the actionable insights gained from the charts above:

- **Chart 1**: For Sunshine Data overall and with cloud nulls filtered out, it appears to be normally distributed with a slight skew to the left. There are also many zeroes, which can be explained by rainy days.

- **Chart 2 and 3**: For Cloud Data overall and with Sunshine nulls removed, the data appears to be hump shaped, with one large hump at a 1 on the scale, and another large hump at a 7 on the scale.

- **Chart 4**: This chart is designed to look at the overall null population, to further understand what data we will need to impute. For our cloud data with sunshine nulls, I would recommend looking further into the rain data to determine how we should handle imputing sunshine values. For sunshine data with cloud nulls, I will impute based off of both the humps; if it rained, I will use the 7 hump, if it did not rain, I will use the 1 hump. If both are null, I will have to use only rain data to determine how I would like to handle these values.

- **Chart 5**: I made this chart to see situations where there are clouds with 0 sunshine, and sunshine with 0 clouds. These values appear to be minimal.

- **Chart 6 and 7**: I created charts here to see how many hours of sunshine there are when it rains, and when it doesn't rain. This could help us biforcate our population of sunshine data so that I am not imputing the mean onto data that doesn't represent the mean.

- **Chart 8**: This shows evaporation data when Sunshine > 0. We can impute the mean here onto our evaporation nulls.

We need to dive deeper into cloud data in order to figure out how to replace nulls. Please see graphs below for more details about our cloud data:

In [36]:

```
# Now I am going to make a subplot looking into the results of our cloud te

fig, ((ax1, ax4), (ax2, ax5), (ax3, ax6)) = plt.subplots(nrows=3, ncols=2,
fig.suptitle('Plots of Cloud data and Sunshine Data')
fig.tight_layout(pad=5.0)

# This is a graph for our cloud data when there is low humidity and it does
ax1.set_title('Chart 9: 9am Cloud Data- Low Humidity, No Rain')
ax1.hist(cloud9_no_rain_lower_humidity)

# This is a graph for our cloud data when there is high humidity and it doe
ax2.set_title('Chart 10: 9am Cloud Data- High Humidity, No Rain')
ax2.hist(cloud9_no_rain_higher_humidity)

# This is a graph for our cloud data when it does rain.
ax3.set_title('Chart 11: 9am Cloud Data- Rains')
ax3.hist(cloud9_rain)

# This is a graph for our cloud data when there is low humidity and it does
ax4.set_title('Chart 9: 3pm Cloud Data- Low Humidity, No Rain')
ax4.hist(cloud3_no_rain_lower_humidity)

# This is a graph for our cloud data when there is high humidity and it doe
ax5.set_title('Chart 10: 3pm Cloud Data- High Humidity, No Rain')
ax5.hist(cloud3_no_rain_higher_humidity)

# This is a graph for our cloud data when it does rain.
ax6.set_title('Chart 11: 3pm Cloud Data- Rains')
ax6.hist(cloud3_rain)

plt.savefig('Visualizations/Subplot2.png', bbox_inches = 'tight')
```

Let's breakdown our Cloud data:

- **Chart 9**: As we can see here, in our data when it doesn't rain and humidity is less than 70%, our Cloud data has a mean closer to 1. I will fill the nulls meeting this criteria using a lower cloud coverage.

- **Chart 10**: As we can see here, in our data when it doesn't rain and humidity is greater than 70%, our Cloud data has a mean closer to 7 or 8. I will fill the nulls meeting this criteria using a higher cloud coverage.

- **Chart 11**: As we can see here, in our data when it does rain, our Cloud data has a mean closer to 7 or 8. I will fill the nulls meeting this criteria using a higher cloud coverage.

Now that I have an understanding of the data, I will perform the train/test split, and begin imputing these null values.

Section 3: Data Preparation

Before imputing my data, I am going to create numeric columns for our categorical fields using OneHotEncoding.

In [37]:
```
X_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106644 entries, 0 to 106643
Data columns (total 22 columns):
 #   Column         Non-Null Count   Dtype  
---  ------         --------------   -----  
 0   Date           106644 non-null  object 
 1   Location       106644 non-null  object 
 2   MinTemp        106150 non-null  float64
 3   MaxTemp        106387 non-null  float64
 4   Rainfall       105540 non-null  float64
 5   Evaporation    60909 non-null   float64
 6   Sunshine       55677 non-null   float64
 7   WindGustDir    99665 non-null   object 
 8   WindGustSpeed  99715 non-null   float64
 9   WindDir9am     99120 non-null   object 
 10  WindDir3pm     103822 non-null  object 
 11  WindSpeed9am   105646 non-null  float64
 12  WindSpeed3pm   104681 non-null  float64
 13  Humidity9am    105326 non-null  float64
 14  Humidity3pm    103917 non-null  float64
 15  Pressure9am    96098 non-null   float64
 16  Pressure3pm    96123 non-null   float64
 17  Cloud9am       66285 non-null   float64
 18  Cloud3pm       63757 non-null   float64
 19  Temp9am        105967 non-null  float64
 20  Temp3pm        104579 non-null  float64
 21  RainToday      105540 non-null  float64
dtypes: float64(17), object(5)
memory usage: 17.9+ MB
```

In [38]:
```
X_train = X_train.drop(['Date'],axis=1)
```

In [39]:
```
# Impute sunshine data based off of our findings from above

X_train['Sunshine'] = np.where(((X_train['Sunshine'].isna()) & (X_train['Ra
X_train['Sunshine'] = np.where(((X_train['Sunshine'].isna()) & (X_train['Ra
```

In [40]:
```
X_train['Sunshine'].isna().any()
```

Out[40]:
```
True
```

In [41]:
```
X_train['Sunshine'].value_counts()
```

Out[41]:
```
11.0    39843
0.0     12859
```

```
10.7        815
10.5        779
10.8        771
10.3        750
10.9        732
10.2        732
9.8         726
11.1        708
10.6        708
10.4        705
10.0        704
10.1        704
9.2         698
11.2        686
9.9         667
9.5         647
9.4         639
9.6         629
9.7         617
9.3         605
9.0         565
8.8         558
11.3        557
9.1         555
8.4         548
8.9         526
8.7         525
8.2         522
11.4        515
8.0         505
11.6        491
8.3         480
8.6         468
7.2         463
8.5         462
7.8         457
13.0        448
8.1         448
12.0        446
11.7        431
13.1        425
7.3         423
13.2        421
11.5        420
7.7         418
12.7        418
11.9        418
7.6         417
11.8        414
6.8         411
7.1         408
6.9         406
7.5         403
```

```
7.4         401
0.1         399
6.3         398
6.1         396
12.2        394
7.0         388
0.2         388
12.5        383
6.6         380
7.9         370
6.2         366
6.0         366
12.4        365
6.5         364
12.6        363
12.3        359
5.7         356
6.4         356
12.1        353
5.8         353
6.7         343
12.8        342
5.0         340
5.5         339
5.2         337
5.6         325
0.3         325
13.3        319
5.9         317
12.9        316
5.1         310
5.4         310
4.8         307
3.8         301
5.3         300
4.3         295
3.0         295
4.4         293
3.2         292
3.9         291
4.0         291
4.5         291
4.7         290
4.1         285
4.9         279
4.6         273
0.7         263
3.6         262
2.7         255
0.4         253
2.3         252
4.2         251
2.8         245
```

```
1.0      244
3.5      239
1.6      236
2.0      235
2.2      234
2.4      233
1.2      233
2.1      230
3.3      229
0.9      229
13.4     227
0.5      226
0.6      226
1.7      224
2.9      224
0.8      223
3.7      220
1.5      217
1.9      216
2.6      215
1.4      213
1.3      213
3.1      212
1.8      211
2.5      210
3.4      205
1.1      201
13.5     147
13.6     126
13.7      88
13.8      48
13.9      14
14.0      10
14.1       5
14.3       4
14.2       2
14.5       1
Name: Sunshine, dtype: int64
```

In [42]:

```
# Impute zero evaporation when sunshine is zero

X_train['Evaporation'] = np.where(((X_train['Evaporation'].isna()) & (X_tra
```

In [43]:

```
evaporation_test.sum()/len(evaporation_test)
```

Out[43]:

```
5.037441856155372
```

In [44]:
```
# Impute mean evaporation when sunshine is not zero

X_train['Evaporation'] = np.where(((X_train['Evaporation'].isna()) & (X_tra
```

In [45]:
```
X_train['Evaporation'].isna().any()
```

Out[45]:
```
True
```

In [46]:
```
X_train['Evaporation'].value_counts().head()
```

Out[46]:
```
5.0    36568
0.0     9910
4.0     2494
8.0     1938
2.2     1567
Name: Evaporation, dtype: int64
```

In [47]:
```
print(cloud9_no_rain_lower_humidity.value_counts())
print(cloud9_no_rain_higher_humidity.value_counts())
print(cloud9_rain.value_counts())
print(cloud3_no_rain_lower_humidity.value_counts())
print(cloud3_no_rain_higher_humidity.value_counts())
print(cloud3_rain.value_counts())
```

```
1.0    7669
0.0    5159
7.0    5003
2.0    2975
6.0    2564
3.0    2482
5.0    1997
4.0    1642
8.0    1574
Name: Cloud9am, dtype: int64
7.0    5366
8.0    4206
1.0    2952
6.0    1884
5.0    1282
2.0    1261
3.0    1179
0.0    1132
4.0     949
Name: Cloud9am, dtype: int64
8.0    4938
7.0    4617
6.0    1631
1.0    1132
```

```
5.0     955
3.0     794
4.0     710
2.0     691
0.0     174
Name: Cloud9am, dtype: int64
1.0    9839
7.0    7523
2.0    4320
6.0    4268
3.0    3943
0.0    3560
5.0    3474
8.0    2850
4.0    2834
9.0       1
Name: Cloud3pm, dtype: int64
8.0    2529
7.0    1878
6.0     540
5.0     282
1.0     223
4.0     199
3.0     187
2.0     184
0.0      31
Name: Cloud3pm, dtype: int64
7.0    4215
8.0    3907
6.0    1865
5.0    1302
3.0     988
1.0     967
4.0     915
2.0     827
0.0      69
Name: Cloud3pm, dtype: int64
```

In [48]:

```
# Imputing cloud data based on our findings from section 2

X_train['Cloud9am'] = np.where(((X_train['Cloud9am'].isna()) & (X_train['Ra
X_train['Cloud3pm'] = np.where(((X_train['Cloud3pm'].isna()) & (X_train['Ra
X_train['Cloud9am'] = np.where(((X_train['Cloud9am'].isna()) & (X_train['Ra
X_train['Cloud9am'] = np.where(((X_train['Cloud9am'].isna()) & (X_train['Ra
X_train['Cloud3pm'] = np.where(((X_train['Cloud3pm'].isna()) & (X_train['Ra
X_train['Cloud3pm'] = np.where(((X_train['Cloud3pm'].isna()) & (X_train['Ra
```

In [49]:

```
X_train['Cloud9am'].value_counts()
```

Out[49]:
```
1.0    29221
7.0    28120
8.0    18799
0.0     6427
6.0     5999
2.0     4847
3.0     4385
5.0     4177
4.0     3260
Name: Cloud9am, dtype: int64
```

In [50]:
```
X_train['Cloud3pm'].value_counts()
```

Out[50]:
```
1.0    38957
7.0    22131
8.0    13165
6.0     6642
2.0     5346
3.0     5126
5.0     5049
4.0     3952
0.0     3720
9.0        1
Name: Cloud3pm, dtype: int64
```

In [51]:
```
# Imputing remaining columns based on mean, from our charts in section 2

imputer = SimpleImputer(strategy='most_frequent')
imputer = imputer.fit(X_train)
X_train.iloc[:,:] = imputer.transform(X_train)
```

In [52]:
```
categorical_data = X_train[['Location','WindGustDir','WindDir9am','WindDir3

ohe = OneHotEncoder()

# Fit the dummy variables to an array
X = ohe.fit_transform(categorical_data.values).toarray()
y = ohe.get_feature_names(['Location','WindGustDir','WindDir9am','WindDir3p

# To add this back into the original dataframe
dfOneHot = pd.DataFrame(X, columns = y)
X_train = pd.concat([X_train, dfOneHot], axis=1)

# Dropping the base columns, and to avoid multicollinearity, dropping one o
X_train = X_train.drop(['Location','WindGustDir','WindDir9am','WindDir3pm']

# Printing to verify
print(X_train.head())
```

```
   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustSpeed  \
```

```
0     14.9     30.3        0.0             7.4      10.9            33.0
1     14.6     21.5        0.2             5.0      11.0            46.0
2      9.0     23.7        0.0             5.0      11.0            28.0
3     15.3     24.0        0.0             8.2      12.1            35.0
4     17.3     37.5        0.0             8.6      11.4            39.0

   WindSpeed9am  WindSpeed3pm  Humidity9am  Humidity3pm  Pressure9am  \
0          15.0          11.0         19.0         12.0       1021.1
1          26.0          28.0         65.0         57.0       1012.6
2          11.0          15.0         59.0         45.0       1017.9
3           4.0          15.0         63.0         82.0       1018.1
4           9.0          15.0         26.0         12.0       1009.6

   Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday  \
0       1017.8       1.0       1.0     22.2     29.7        0.0
1       1013.5       1.0       1.0     17.5     18.6        0.0
2       1015.1       1.0       1.0     14.6     23.1        0.0
3       1016.7       3.0       3.0     21.8     21.8        0.0
4       1006.2       7.0       4.0     23.8     35.7        0.0

   Location_Adelaide  Location_Albany  Location_Albury  Location_AliceSpring
s  \
0                0.0              0.0              0.0                    0.
0
1                0.0              0.0              0.0                    0.
0
2                0.0              0.0              0.0                    0.
0
3                0.0              1.0              0.0                    0.
0
4                0.0              0.0              0.0                    0.
0

   Location_BadgerysCreek  Location_Ballarat  Location_Bendigo  \
0                     0.0                0.0               0.0
1                     0.0                0.0               0.0
2                     0.0                0.0               0.0
3                     0.0                0.0               0.0
4                     0.0                0.0               0.0

   Location_Brisbane  Location_Cairns  Location_Canberra  Location_Cobar  \
0                0.0              0.0                0.0             0.0
1                0.0              0.0                0.0             0.0
2                0.0              0.0                0.0             0.0
3                0.0              0.0                0.0             0.0
4                0.0              0.0                0.0             0.0

   Location_CoffsHarbour  Location_Dartmoor  Location_Darwin  \
0                    0.0                0.0              0.0
1                    0.0                0.0              0.0
2                    0.0                0.0              0.0
3                    0.0                0.0              0.0
4                    0.0                0.0              0.0
```

```
   Location_GoldCoast  Location_Hobart  Location_Katherine  \
0                 0.0              0.0                 0.0   
1                 0.0              0.0                 0.0   
2                 0.0              0.0                 0.0   
3                 0.0              0.0                 0.0   
4                 0.0              0.0                 0.0   

   Location_Launceston  Location_Melbourne  Location_MelbourneAirport  \
0                  0.0                 0.0                        0.0   
1                  0.0                 0.0                        0.0   
2                  0.0                 0.0                        0.0   
3                  0.0                 0.0                        0.0   
4                  0.0                 0.0                        0.0   

   Location_Mildura  Location_Moree  Location_MountGambier  \
0               0.0             0.0                    0.0   
1               0.0             0.0                    0.0   
2               0.0             0.0                    0.0   
3               0.0             0.0                    0.0   
4               1.0             0.0                    0.0   

   Location_MountGinini  Location_Newcastle  Location_Nhil  \
0                   0.0                 0.0            0.0   
1                   0.0                 0.0            0.0   
2                   0.0                 0.0            0.0   
3                   0.0                 0.0            0.0   
4                   0.0                 0.0            0.0   

   Location_NorahHead  Location_NorfolkIsland  Location_Nuriootpa  \
0                 0.0                     0.0                0.0   
1                 0.0                     0.0                0.0   
2                 0.0                     0.0                0.0   
3                 0.0                     0.0                0.0   
4                 0.0                     0.0                0.0   

   Location_PearceRAAF  Location_Penrith  Location_Perth  \
0                  0.0               0.0             0.0   
1                  0.0               0.0             0.0   
2                  0.0               0.0             0.0   
3                  0.0               0.0             0.0   
4                  0.0               0.0             0.0   

   Location_PerthAirport  Location_Portland  Location_Richmond  Location_Sal
e  \
0                    0.0                0.0                0.0             0.
0   
1                    0.0                0.0                0.0             0.
0   
2                    0.0                0.0                0.0             0.
0   
3                    0.0                0.0                0.0             0.
0   
```

```
4                    0.0                 0.0                 0.0              0.
0

   Location_SalmonGums  Location_Sydney  Location_SydneyAirport  \
0                  0.0              0.0                     0.0
1                  0.0              0.0                     0.0
2                  1.0              0.0                     0.0
3                  0.0              0.0                     0.0
4                  0.0              0.0                     0.0

   Location_Townsville  Location_Tuggeranong  Location_Uluru  \
0                  0.0                   0.0             0.0
1                  0.0                   0.0             0.0
2                  0.0                   0.0             0.0
3                  0.0                   0.0             0.0
4                  0.0                   0.0             0.0

   Location_WaggaWagga  Location_Walpole  Location_Watsonia  \
0                  0.0               0.0                0.0
1                  0.0               0.0                0.0
2                  0.0               0.0                0.0
3                  0.0               0.0                0.0
4                  0.0               0.0                0.0

   Location_Williamtown  Location_Witchcliffe  Location_Wollongong  \
0                   0.0                   0.0                  0.0
1                   0.0                   1.0                  0.0
2                   0.0                   0.0                  0.0
3                   0.0                   0.0                  0.0
4                   0.0                   0.0                  0.0

   Location_Woomera  WindGustDir_E  WindGustDir_ENE  WindGustDir_ESE  \
0               1.0            0.0              0.0              0.0
1               0.0            0.0              0.0              0.0
2               0.0            0.0              0.0              0.0
3               0.0            0.0              0.0              0.0
4               0.0            0.0              0.0              0.0

   WindGustDir_N  WindGustDir_NE  WindGustDir_NNE  WindGustDir_NNW  \
0            0.0             0.0              0.0              0.0
1            0.0             0.0              0.0              0.0
2            0.0             0.0              0.0              0.0
3            0.0             0.0              0.0              0.0
4            1.0             0.0              0.0              0.0

   WindGustDir_NW  WindGustDir_S  WindGustDir_SE  WindGustDir_SSE  \
0             0.0            1.0             0.0              0.0
1             0.0            0.0             0.0              1.0
2             0.0            0.0             0.0              0.0
3             0.0            0.0             0.0              0.0
4             0.0            0.0             0.0              0.0

   WindGustDir_SSW  WindGustDir_SW  WindGustDir_W  WindGustDir_WNW  \
```

```
0            0.0             0.0           0.0              0.0
1            0.0             0.0           0.0              0.0
2            0.0             0.0           1.0              0.0
3            0.0             0.0           1.0              0.0
4            0.0             0.0           0.0              0.0

   WindGustDir_WSW  WindDir9am_E  WindDir9am_ENE  WindDir9am_ESE  \
0              0.0           0.0             0.0             1.0
1              0.0           0.0             0.0             0.0
2              0.0           0.0             0.0             0.0
3              0.0           0.0             0.0             0.0
4              0.0           0.0             0.0             0.0

   WindDir9am_N  WindDir9am_NE  WindDir9am_NNE  WindDir9am_NNW  WindDir9am_N
W  \
0           0.0            0.0             0.0             0.0            0.
0
1           0.0            0.0             0.0             0.0            0.
0
2           0.0            0.0             0.0             0.0            0.
0
3           0.0            0.0             0.0             0.0            0.
0
4           0.0            1.0             0.0             0.0            0.
0

   WindDir9am_S  WindDir9am_SE  WindDir9am_SSE  WindDir9am_SSW  WindDir9am_S
W  \
0           0.0            0.0             0.0             0.0            0.
0
1           1.0            0.0             0.0             0.0            0.
0
2           0.0            0.0             0.0             0.0            0.
0
3           0.0            0.0             0.0             1.0            0.
0
4           0.0            0.0             0.0             0.0            0.
0

   WindDir9am_W  WindDir9am_WNW  WindDir9am_WSW  WindDir3pm_E  WindDir3pm_EN
E  \
0           0.0             0.0             0.0           0.0            0.
0
1           0.0             0.0             0.0           0.0            0.
0
2           0.0             1.0             0.0           0.0            0.
0
3           0.0             0.0             0.0           0.0            0.
0
4           0.0             0.0             0.0           0.0            0.
0

   WindDir3pm_ESE  WindDir3pm_N  WindDir3pm_NE  WindDir3pm_NNE  \
```

```
0             0.0           0.0            0.0             0.0
1             0.0           0.0            0.0             0.0
2             0.0           0.0            0.0             0.0
3             0.0           0.0            0.0             0.0
4             0.0           0.0            0.0             0.0

   WindDir3pm_NNW  WindDir3pm_NW  WindDir3pm_S  WindDir3pm_SE  WindDir3pm_SS
E  \
0             0.0            0.0           0.0            0.0            0.
0
1             0.0            0.0           0.0            0.0            1.
0
2             0.0            0.0           0.0            0.0            0.
0
3             0.0            0.0           0.0            0.0            0.
0
4             0.0            1.0           0.0            0.0            0.
0

   WindDir3pm_SSW  WindDir3pm_SW  WindDir3pm_W  WindDir3pm_WNW  WindDir3pm_W
SW
0             0.0            1.0           0.0             0.0             0
.0
1             0.0            0.0           0.0             0.0             0
.0
2             0.0            0.0           1.0             0.0             0
.0
3             1.0            0.0           0.0             0.0             0
.0
4             0.0            0.0           0.0             0.0             0
.0
```

In [53]:

```
# Imputing remaining columns based on mean, from our charts in section 2

imputer = SimpleImputer(strategy='most_frequent', missing_values=np.nan)
imputer = imputer.fit(X_train)
X_train.iloc[:,:] = imputer.transform(X_train)
```

In [54]:

```
X_train.info(1, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 106644 entries, 0 to 106643
Data columns (total 114 columns):
 #   Column                       Non-Null Count   Dtype
---  ------                       --------------   -----
 0   MinTemp                      106644 non-null  float64
 1   MaxTemp                      106644 non-null  float64
 2   Rainfall                     106644 non-null  float64
 3   Evaporation                  106644 non-null  float64
 4   Sunshine                     106644 non-null  float64
 5   WindGustSpeed                106644 non-null  float64
```

```
 6   WindSpeed9am              106644 non-null  float64
 7   WindSpeed3pm              106644 non-null  float64
 8   Humidity9am               106644 non-null  float64
 9   Humidity3pm               106644 non-null  float64
 10  Pressure9am               106644 non-null  float64
 11  Pressure3pm               106644 non-null  float64
 12  Cloud9am                  106644 non-null  float64
 13  Cloud3pm                  106644 non-null  float64
 14  Temp9am                   106644 non-null  float64
 15  Temp3pm                   106644 non-null  float64
 16  RainToday                 106644 non-null  float64
 17  Location_Adelaide         106644 non-null  float64
 18  Location_Albany           106644 non-null  float64
 19  Location_Albury           106644 non-null  float64
 20  Location_AliceSprings     106644 non-null  float64
 21  Location_BadgerysCreek    106644 non-null  float64
 22  Location_Ballarat         106644 non-null  float64
 23  Location_Bendigo          106644 non-null  float64
 24  Location_Brisbane         106644 non-null  float64
 25  Location_Cairns           106644 non-null  float64
 26  Location_Canberra         106644 non-null  float64
 27  Location_Cobar            106644 non-null  float64
 28  Location_CoffsHarbour     106644 non-null  float64
 29  Location_Dartmoor         106644 non-null  float64
 30  Location_Darwin           106644 non-null  float64
 31  Location_GoldCoast        106644 non-null  float64
 32  Location_Hobart           106644 non-null  float64
 33  Location_Katherine        106644 non-null  float64
 34  Location_Launceston       106644 non-null  float64
 35  Location_Melbourne        106644 non-null  float64
 36  Location_MelbourneAirport 106644 non-null  float64
 37  Location_Mildura          106644 non-null  float64
 38  Location_Moree            106644 non-null  float64
 39  Location_MountGambier     106644 non-null  float64
 40  Location_MountGinini      106644 non-null  float64
 41  Location_Newcastle        106644 non-null  float64
 42  Location_Nhil             106644 non-null  float64
 43  Location_NorahHead        106644 non-null  float64
 44  Location_NorfolkIsland    106644 non-null  float64
 45  Location_Nuriootpa        106644 non-null  float64
 46  Location_PearceRAAF       106644 non-null  float64
 47  Location_Penrith          106644 non-null  float64
 48  Location_Perth            106644 non-null  float64
 49  Location_PerthAirport     106644 non-null  float64
 50  Location_Portland         106644 non-null  float64
 51  Location_Richmond         106644 non-null  float64
 52  Location_Sale             106644 non-null  float64
 53  Location_SalmonGums       106644 non-null  float64
 54  Location_Sydney           106644 non-null  float64
 55  Location_SydneyAirport    106644 non-null  float64
 56  Location_Townsville       106644 non-null  float64
 57  Location_Tuggeranong      106644 non-null  float64
 58  Location_Uluru            106644 non-null  float64
```

```
 59  Location_WaggaWagga        106644 non-null  float64
 60  Location_Walpole           106644 non-null  float64
 61  Location_Watsonia          106644 non-null  float64
 62  Location_Williamtown       106644 non-null  float64
 63  Location_Witchcliffe       106644 non-null  float64
 64  Location_Wollongong        106644 non-null  float64
 65  Location_Woomera           106644 non-null  float64
 66  WindGustDir_E              106644 non-null  float64
 67  WindGustDir_ENE            106644 non-null  float64
 68  WindGustDir_ESE            106644 non-null  float64
 69  WindGustDir_N              106644 non-null  float64
 70  WindGustDir_NE             106644 non-null  float64
 71  WindGustDir_NNE            106644 non-null  float64
 72  WindGustDir_NNW            106644 non-null  float64
 73  WindGustDir_NW             106644 non-null  float64
 74  WindGustDir_S              106644 non-null  float64
 75  WindGustDir_SE             106644 non-null  float64
 76  WindGustDir_SSE            106644 non-null  float64
 77  WindGustDir_SSW            106644 non-null  float64
 78  WindGustDir_SW             106644 non-null  float64
 79  WindGustDir_W              106644 non-null  float64
 80  WindGustDir_WNW            106644 non-null  float64
 81  WindGustDir_WSW            106644 non-null  float64
 82  WindDir9am_E               106644 non-null  float64
 83  WindDir9am_ENE             106644 non-null  float64
 84  WindDir9am_ESE             106644 non-null  float64
 85  WindDir9am_N               106644 non-null  float64
 86  WindDir9am_NE              106644 non-null  float64
 87  WindDir9am_NNE             106644 non-null  float64
 88  WindDir9am_NNW             106644 non-null  float64
 89  WindDir9am_NW              106644 non-null  float64
 90  WindDir9am_S               106644 non-null  float64
 91  WindDir9am_SE              106644 non-null  float64
 92  WindDir9am_SSE             106644 non-null  float64
 93  WindDir9am_SSW             106644 non-null  float64
 94  WindDir9am_SW              106644 non-null  float64
 95  WindDir9am_W               106644 non-null  float64
 96  WindDir9am_WNW             106644 non-null  float64
 97  WindDir9am_WSW             106644 non-null  float64
 98  WindDir3pm_E               106644 non-null  float64
 99  WindDir3pm_ENE             106644 non-null  float64
 100 WindDir3pm_ESE             106644 non-null  float64
 101 WindDir3pm_N               106644 non-null  float64
 102 WindDir3pm_NE              106644 non-null  float64
 103 WindDir3pm_NNE             106644 non-null  float64
 104 WindDir3pm_NNW             106644 non-null  float64
 105 WindDir3pm_NW              106644 non-null  float64
 106 WindDir3pm_S               106644 non-null  float64
 107 WindDir3pm_SE              106644 non-null  float64
 108 WindDir3pm_SSE             106644 non-null  float64
 109 WindDir3pm_SSW             106644 non-null  float64
 110 WindDir3pm_SW              106644 non-null  float64
 111 WindDir3pm_W               106644 non-null  float64
```

```
 112 WindDir3pm_WNW              106644 non-null  float64
 113 WindDir3pm_WSW              106644 non-null  float64
dtypes: float64(114)
memory usage: 92.8 MB
```

In [55]:

```
y_train.count()
```

Out[55]:

```
106644
```

In [56]:

```
X_test.info(1, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35549 entries, 0 to 35548
Data columns (total 22 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           35549 non-null  object
 1   Location       35549 non-null  object
 2   MinTemp        35406 non-null  float64
 3   MaxTemp        35484 non-null  float64
 4   Rainfall       35247 non-null  float64
 5   Evaporation    20441 non-null  float64
 6   Sunshine       18700 non-null  float64
 7   WindGustDir    33198 non-null  object
 8   WindGustSpeed  33208 non-null  float64
 9   WindDir9am     33060 non-null  object
 10  WindDir3pm     34593 non-null  object
 11  WindSpeed9am   35199 non-null  float64
 12  WindSpeed3pm   34882 non-null  float64
 13  Humidity9am    35093 non-null  float64
 14  Humidity3pm    34666 non-null  float64
 15  Pressure9am    32081 non-null  float64
 16  Pressure3pm    32089 non-null  float64
 17  Cloud9am       22251 non-null  float64
 18  Cloud3pm       21342 non-null  float64
 19  Temp9am        35322 non-null  float64
 20  Temp3pm        34888 non-null  float64
 21  RainToday      35247 non-null  float64
dtypes: float64(17), object(5)
memory usage: 6.0+ MB
```

In [57]:

```
X_test.reset_index(inplace=True)
```

In [58]:

```
X_test = X_test.drop(columns=['index'],axis=1)
```

In [59]:

```
# Performing same imputations as train data

X_test['Sunshine'] = np.where(((X_test['Sunshine'].isna()) & (X_test['RainT
X_test['Sunshine'] = np.where(((X_test['Sunshine'].isna()) & (X_test['RainT
X_test['Evaporation'] = np.where(((X_test['Evaporation'].isna()) & (X_test[
X_test['Evaporation'] = np.where(((X_test['Evaporation'].isna()) & (X_test[
X_test['Cloud9am'] = np.where(((X_test['Cloud9am'].isna()) & (X_test['RainT
X_test['Cloud3pm'] = np.where(((X_test['Cloud3pm'].isna()) & (X_test['RainT
X_test['Cloud9am'] = np.where(((X_test['Cloud9am'].isna()) & (X_test['RainT
X_test['Cloud9am'] = np.where(((X_test['Cloud9am'].isna()) & (X_test['RainT
X_test['Cloud3pm'] = np.where(((X_test['Cloud3pm'].isna()) & (X_test['RainT
X_test['Cloud3pm'] = np.where(((X_test['Cloud3pm'].isna()) & (X_test['RainT
```

In [60]:

```
# Performing same imputations as train data

imputer = SimpleImputer(strategy='most_frequent')
imputer = imputer.fit(X_test)
X_test.iloc[:,:] = imputer.transform(X_test)
```

In [61]:

```
categorical_data = X_test[['Location','WindGustDir','WindDir9am','WindDir3p

ohe = OneHotEncoder()

# Fit the dummy variables to an array
X = ohe.fit_transform(categorical_data.values).toarray()
y = ohe.get_feature_names(['Location','WindGustDir','WindDir9am','WindDir3p

# To add this back into the original dataframe
dfOneHot = pd.DataFrame(X, columns = y)
X_test = pd.concat([X_test, dfOneHot], axis=1)

# Dropping the country column
X_test = X_test.drop(['Date','Location','WindGustDir','WindDir9am','WindDir

# Printing to verify
print(X_test.head())
```

```
   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustSpeed  \
0      7.1     13.0       8.8          0.0       0.0           41.0
1     13.2     18.3       0.0          5.0       7.0           48.0
2      9.2     22.7       0.0          5.0      11.1           52.0
3     15.3     26.1       0.0         10.4       7.0           44.0
4     11.9     31.8       0.0          5.0       4.1           72.0

   WindSpeed9am  WindSpeed3pm  Humidity9am  Humidity3pm  Pressure9am  \
0          24.0          22.0        100.0         98.0       1001.7
1          24.0          20.0         73.0         73.0       1027.6
2          26.0          20.0         45.0         25.0       1030.1
3          24.0          19.0         48.0         40.0       1013.2
```

```
4          6.0          19.0          89.0          25.0        1006.7

   Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday  \
0       1005.4       8.0       8.0      8.6     11.5        1.0
1       1023.8       7.0       8.0     14.2     17.0        0.0
2       1025.9       0.0       0.0     15.1     22.5        0.0
3       1009.8       7.0       7.0     17.5     24.3        0.0
4       1001.0       7.0       6.0     16.2     27.4        0.0

   Location_Adelaide  Location_Albany  Location_Albury  Location_AliceSpring
s  \
0                0.0              0.0              0.0                   0.
0
1                0.0              0.0              0.0                   0.
0
2                0.0              0.0              0.0                   0.
0
3                0.0              0.0              0.0                   0.
0
4                0.0              0.0              0.0                   0.
0

   Location_BadgerysCreek  Location_Ballarat  Location_Bendigo  \
0                     0.0                1.0               0.0
1                     0.0                0.0               0.0
2                     0.0                0.0               0.0
3                     0.0                0.0               0.0
4                     0.0                0.0               0.0

   Location_Brisbane  Location_Cairns  Location_Canberra  Location_Cobar  \
0                0.0              0.0                0.0             0.0
1                0.0              0.0                0.0             0.0
2                0.0              0.0                0.0             0.0
3                0.0              0.0                0.0             1.0
4                0.0              0.0                0.0             0.0

   Location_CoffsHarbour  Location_Dartmoor  Location_Darwin  \
0                    0.0                0.0              0.0
1                    0.0                0.0              0.0
2                    0.0                0.0              0.0
3                    0.0                0.0              0.0
4                    0.0                0.0              0.0

   Location_GoldCoast  Location_Hobart  Location_Katherine  \
0                 0.0              0.0                 0.0
1                 0.0              0.0                 0.0
2                 0.0              0.0                 0.0
3                 0.0              0.0                 0.0
4                 0.0              0.0                 0.0

   Location_Launceston  Location_Melbourne  Location_MelbourneAirport  \
0                  0.0                 0.0                        0.0
1                  0.0                 0.0                        0.0
```

```
2                0.0                 0.0                           0.0
3                0.0                 0.0                           0.0
4                0.0                 0.0                           0.0

   Location_Mildura  Location_Moree  Location_MountGambier  \
0               0.0             0.0                    0.0
1               0.0             0.0                    0.0
2               0.0             0.0                    0.0
3               0.0             0.0                    0.0
4               0.0             0.0                    0.0

   Location_MountGinini  Location_Newcastle  Location_Nhil  \
0                   0.0                 0.0            0.0
1                   0.0                 0.0            0.0
2                   0.0                 0.0            0.0
3                   0.0                 0.0            0.0
4                   0.0                 0.0            0.0

   Location_NorahHead  Location_NorfolkIsland  Location_Nuriootpa  \
0                 0.0                     0.0                0.0
1                 0.0                     0.0                0.0
2                 0.0                     0.0                0.0
3                 0.0                     0.0                0.0
4                 0.0                     0.0                0.0

   Location_PearceRAAF  Location_Penrith  Location_Perth  \
0                  0.0               0.0             0.0
1                  0.0               0.0             0.0
2                  0.0               0.0             0.0
3                  0.0               0.0             0.0
4                  0.0               0.0             0.0

   Location_PerthAirport  Location_Portland  Location_Richmond  Location_Sal
e  \
0                    0.0                0.0                0.0             0.
0
1                    0.0                0.0                0.0             0.
0
2                    1.0                0.0                0.0             0.
0
3                    0.0                0.0                0.0             0.
0
4                    0.0                0.0                0.0             1.
0

   Location_SalmonGums  Location_Sydney  Location_SydneyAirport  \
0                  0.0              0.0                     0.0
1                  0.0              0.0                     0.0
2                  0.0              0.0                     0.0
3                  0.0              0.0                     0.0
4                  0.0              0.0                     0.0

   Location_Townsville  Location_Tuggeranong  Location_Uluru  \
```

```
0                    0.0                    0.0              0.0
1                    0.0                    0.0              0.0
2                    0.0                    0.0              0.0
3                    0.0                    0.0              0.0
4                    0.0                    0.0              0.0

   Location_WaggaWagga  Location_Walpole  Location_Watsonia  \
0                  0.0               0.0                0.0
1                  0.0               1.0                0.0
2                  0.0               0.0                0.0
3                  0.0               0.0                0.0
4                  0.0               0.0                0.0

   Location_Williamtown  Location_Witchcliffe  Location_Wollongong  \
0                   0.0                   0.0                  0.0
1                   0.0                   0.0                  0.0
2                   0.0                   0.0                  0.0
3                   0.0                   0.0                  0.0
4                   0.0                   0.0                  0.0

   Location_Woomera  WindGustDir_E  WindGustDir_ENE  WindGustDir_ESE  \
0               0.0            0.0              0.0              0.0
1               0.0            1.0              0.0              0.0
2               0.0            0.0              1.0              0.0
3               0.0            1.0              0.0              0.0
4               0.0            0.0              0.0              0.0

   WindGustDir_N  WindGustDir_NE  WindGustDir_NNE  WindGustDir_NNW  \
0            1.0             0.0              0.0              0.0
1            0.0             0.0              0.0              0.0
2            0.0             0.0              0.0              0.0
3            0.0             0.0              0.0              0.0
4            0.0             0.0              0.0              0.0

   WindGustDir_NW  WindGustDir_S  WindGustDir_SE  WindGustDir_SSE  \
0             0.0            0.0             0.0              0.0
1             0.0            0.0             0.0              0.0
2             0.0            0.0             0.0              0.0
3             0.0            0.0             0.0              0.0
4             1.0            0.0             0.0              0.0

   WindGustDir_SSW  WindGustDir_SW  WindGustDir_W  WindGustDir_WNW  \
0              0.0             0.0            0.0              0.0
1              0.0             0.0            0.0              0.0
2              0.0             0.0            0.0              0.0
3              0.0             0.0            0.0              0.0
4              0.0             0.0            0.0              0.0

   WindGustDir_WSW  WindDir9am_E  WindDir9am_ENE  WindDir9am_ESE  \
0              0.0           0.0             0.0             0.0
1              0.0           0.0             0.0             1.0
2              0.0           0.0             1.0             0.0
3              0.0           1.0             0.0             0.0
```

```
4              0.0             1.0              0.0              0.0

   WindDir9am_N  WindDir9am_NE  WindDir9am_NNE  WindDir9am_NNW  WindDir9am_N
W  \
0           1.0            0.0             0.0             0.0             0.
0
1           0.0            0.0             0.0             0.0             0.
0
2           0.0            0.0             0.0             0.0             0.
0
3           0.0            0.0             0.0             0.0             0.
0
4           0.0            0.0             0.0             0.0             0.
0

   WindDir9am_S  WindDir9am_SE  WindDir9am_SSE  WindDir9am_SSW  WindDir9am_S
W  \
0           0.0            0.0             0.0             0.0             0.
0
1           0.0            0.0             0.0             0.0             0.
0
2           0.0            0.0             0.0             0.0             0.
0
3           0.0            0.0             0.0             0.0             0.
0
4           0.0            0.0             0.0             0.0             0.
0

   WindDir9am_W  WindDir9am_WNW  WindDir9am_WSW  WindDir3pm_E  WindDir3pm_EN
E  \
0           0.0             0.0             0.0           0.0             0.
0
1           0.0             0.0             0.0           0.0             0.
0
2           0.0             0.0             0.0           0.0             0.
0
3           0.0             0.0             0.0           0.0             0.
0
4           0.0             0.0             0.0           0.0             0.
0

   WindDir3pm_ESE  WindDir3pm_N  WindDir3pm_NE  WindDir3pm_NNE  \
0             0.0           0.0            0.0             0.0
1             1.0           0.0            0.0             0.0
2             1.0           0.0            0.0             0.0
3             0.0           0.0            1.0             0.0
4             0.0           1.0            0.0             0.0

   WindDir3pm_NNW  WindDir3pm_NW  WindDir3pm_S  WindDir3pm_SE  WindDir3pm_SS
E  \
0             0.0            0.0           0.0            0.0             0.
0
1             0.0            0.0           0.0            0.0             0.
```

```
0
2            0.0          0.0          0.0          0.0          0.
0
3            0.0          0.0          0.0          0.0          0.
0
4            0.0          0.0          0.0          0.0          0.
0

   WindDir3pm_SSW  WindDir3pm_SW  WindDir3pm_W  WindDir3pm_WNW  WindDir3pm_W
SW
0            0.0          0.0          0.0          1.0          0
.0
1            0.0          0.0          0.0          0.0          0
.0
2            0.0          0.0          0.0          0.0          0
.0
3            0.0          0.0          0.0          0.0          0
.0
4            0.0          0.0          0.0          0.0          0
.0
```

In [62]:

```python
X_test.info(1, null_counts=True)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35549 entries, 0 to 35548
Data columns (total 114 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   MinTemp                       35549 non-null  float64
 1   MaxTemp                       35549 non-null  float64
 2   Rainfall                      35549 non-null  float64
 3   Evaporation                   35549 non-null  float64
 4   Sunshine                      35549 non-null  float64
 5   WindGustSpeed                 35549 non-null  float64
 6   WindSpeed9am                  35549 non-null  float64
 7   WindSpeed3pm                  35549 non-null  float64
 8   Humidity9am                   35549 non-null  float64
 9   Humidity3pm                   35549 non-null  float64
 10  Pressure9am                   35549 non-null  float64
 11  Pressure3pm                   35549 non-null  float64
 12  Cloud9am                      35549 non-null  float64
 13  Cloud3pm                      35549 non-null  float64
 14  Temp9am                       35549 non-null  float64
 15  Temp3pm                       35549 non-null  float64
 16  RainToday                     35549 non-null  float64
 17  Location_Adelaide             35549 non-null  float64
 18  Location_Albany               35549 non-null  float64
 19  Location_Albury               35549 non-null  float64
 20  Location_AliceSprings         35549 non-null  float64
 21  Location_BadgerysCreek        35549 non-null  float64
 22  Location_Ballarat             35549 non-null  float64
 23  Location_Bendigo              35549 non-null  float64
```

```
 24  Location_Brisbane          35549 non-null  float64
 25  Location_Cairns            35549 non-null  float64
 26  Location_Canberra          35549 non-null  float64
 27  Location_Cobar             35549 non-null  float64
 28  Location_CoffsHarbour      35549 non-null  float64
 29  Location_Dartmoor          35549 non-null  float64
 30  Location_Darwin            35549 non-null  float64
 31  Location_GoldCoast         35549 non-null  float64
 32  Location_Hobart            35549 non-null  float64
 33  Location_Katherine         35549 non-null  float64
 34  Location_Launceston        35549 non-null  float64
 35  Location_Melbourne         35549 non-null  float64
 36  Location_MelbourneAirport  35549 non-null  float64
 37  Location_Mildura           35549 non-null  float64
 38  Location_Moree             35549 non-null  float64
 39  Location_MountGambier      35549 non-null  float64
 40  Location_MountGinini       35549 non-null  float64
 41  Location_Newcastle         35549 non-null  float64
 42  Location_Nhil              35549 non-null  float64
 43  Location_NorahHead         35549 non-null  float64
 44  Location_NorfolkIsland     35549 non-null  float64
 45  Location_Nuriootpa         35549 non-null  float64
 46  Location_PearceRAAF        35549 non-null  float64
 47  Location_Penrith           35549 non-null  float64
 48  Location_Perth             35549 non-null  float64
 49  Location_PerthAirport      35549 non-null  float64
 50  Location_Portland          35549 non-null  float64
 51  Location_Richmond          35549 non-null  float64
 52  Location_Sale              35549 non-null  float64
 53  Location_SalmonGums        35549 non-null  float64
 54  Location_Sydney            35549 non-null  float64
 55  Location_SydneyAirport     35549 non-null  float64
 56  Location_Townsville        35549 non-null  float64
 57  Location_Tuggeranong       35549 non-null  float64
 58  Location_Uluru             35549 non-null  float64
 59  Location_WaggaWagga        35549 non-null  float64
 60  Location_Walpole           35549 non-null  float64
 61  Location_Watsonia          35549 non-null  float64
 62  Location_Williamtown       35549 non-null  float64
 63  Location_Witchcliffe       35549 non-null  float64
 64  Location_Wollongong        35549 non-null  float64
 65  Location_Woomera           35549 non-null  float64
 66  WindGustDir_E              35549 non-null  float64
 67  WindGustDir_ENE            35549 non-null  float64
 68  WindGustDir_ESE            35549 non-null  float64
 69  WindGustDir_N              35549 non-null  float64
 70  WindGustDir_NE             35549 non-null  float64
 71  WindGustDir_NNE            35549 non-null  float64
 72  WindGustDir_NNW            35549 non-null  float64
 73  WindGustDir_NW             35549 non-null  float64
 74  WindGustDir_S              35549 non-null  float64
 75  WindGustDir_SE             35549 non-null  float64
 76  WindGustDir_SSE            35549 non-null  float64
```

```
 77  WindGustDir_SSW           35549 non-null  float64
 78  WindGustDir_SW            35549 non-null  float64
 79  WindGustDir_W             35549 non-null  float64
 80  WindGustDir_WNW           35549 non-null  float64
 81  WindGustDir_WSW           35549 non-null  float64
 82  WindDir9am_E              35549 non-null  float64
 83  WindDir9am_ENE            35549 non-null  float64
 84  WindDir9am_ESE            35549 non-null  float64
 85  WindDir9am_N              35549 non-null  float64
 86  WindDir9am_NE             35549 non-null  float64
 87  WindDir9am_NNE            35549 non-null  float64
 88  WindDir9am_NNW            35549 non-null  float64
 89  WindDir9am_NW             35549 non-null  float64
 90  WindDir9am_S              35549 non-null  float64
 91  WindDir9am_SE             35549 non-null  float64
 92  WindDir9am_SSE            35549 non-null  float64
 93  WindDir9am_SSW            35549 non-null  float64
 94  WindDir9am_SW             35549 non-null  float64
 95  WindDir9am_W              35549 non-null  float64
 96  WindDir9am_WNW            35549 non-null  float64
 97  WindDir9am_WSW            35549 non-null  float64
 98  WindDir3pm_E              35549 non-null  float64
 99  WindDir3pm_ENE            35549 non-null  float64
 100 WindDir3pm_ESE            35549 non-null  float64
 101 WindDir3pm_N              35549 non-null  float64
 102 WindDir3pm_NE             35549 non-null  float64
 103 WindDir3pm_NNE            35549 non-null  float64
 104 WindDir3pm_NNW            35549 non-null  float64
 105 WindDir3pm_NW             35549 non-null  float64
 106 WindDir3pm_S              35549 non-null  float64
 107 WindDir3pm_SE             35549 non-null  float64
 108 WindDir3pm_SSE            35549 non-null  float64
 109 WindDir3pm_SSW            35549 non-null  float64
 110 WindDir3pm_SW             35549 non-null  float64
 111 WindDir3pm_W              35549 non-null  float64
 112 WindDir3pm_WNW            35549 non-null  float64
 113 WindDir3pm_WSW            35549 non-null  float64
dtypes: float64(114)
memory usage: 30.9 MB
```

In [63]:

```
y_test.count()
```

Out[63]:

```
35549
```

In [64]:

```
# Re-distributing our population using SMOTE, to account for accuracy bias

smote = SMOTE()
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

Section 4: Modeling

To begin, I will build a logistic model. After that, I will be building a kNN model and a decision tree model, and will pick the best model from the three.

The metric I focus on the most will be recall, because if there are false negatives in my model, then it could rain on citizens who expected a dry day. As the Bureau of Meteorology has been underestimating rain over the past five years, we are trying to minimize false negatives as much as possible.

Logistic Regression using sklearn

In [65]:
```
# Create a Baseline Model

logreg = LogisticRegression()
model_log = logreg.fit(X_train_resampled, y_train_resampled)
model_log.score(X_test,y_test)
y_pred_lr = model_log.predict(X_test)
```

In [66]:
```
recall_score(y_test,y_pred_lr)
```

Out[66]:
```
0.7696189755235933
```

I will now run various models through a Pipeline to see if I can find a combination of parameters to bring down our performance metric.

In [67]:
```
# Create pipeline

lr_pipeline_1 = Pipeline([('ss', StandardScaler()),
                          ('lr', LogisticRegression())])
```

In [68]:
```
# Create grid using GridSearchCV

lr_grid = [{'lr__random_state':[42],
         'lr__C': [1,1e2,1e3],
         'lr__solver': ['saga','lbfgs']}]
```

In [69]:

```
# Create grid using GridSearchCV

lr_gridsearch = GridSearchCV(estimator=lr_pipeline_1,
                             param_grid=lr_grid,
                             scoring='recall',
                             cv=3)
```

In [70]:

```
y_train_resampled.value_counts()
```

Out[70]:

```
1.0    82693
0.0    82693
Name: RainTomorrow, dtype: int64
```

In [71]:

```
# Fit the training data
model = lr_gridsearch.fit(X_train_resampled, y_train_resampled)

# Print the recall score for train and test data
print('Train Score: ',lr_gridsearch.score(X_train_resampled,y_train_resampl
print('Test Score: ',lr_gridsearch.score(X_test, y_test))

# Print best parameters
print('\n',lr_gridsearch.best_estimator_)
```

```
Train Score:  0.7777079075617042
Test Score:  0.7847590209437295

 Pipeline(steps=[('ss', StandardScaler()),
                ('lr',
                 LogisticRegression(C=100.0, random_state=42, solver='saga')
)])
```

In [72]:

```
# Created Scaled Data for recreating model
scaler = StandardScaler()
X_train_resampled_scaled = scaler.fit_transform(X_train_resampled)
X_train_resampled_scaled_df = pd.DataFrame(X_train_resampled_scaled, index=
X_test_scaled = scaler.fit_transform(X_test)
X_test_scaled_df = pd.DataFrame(X_test_scaled, index=X_test.index, columns=

# Using the best parameters, recreate the model and generate coefficients a
lr_best = LogisticRegression(C=1, random_state=42, solver = 'saga')
lr_best.fit(X_train_resampled_scaled_df, y_train_resampled)
print(lr_best.coef_)
print(X_train_resampled_scaled_df.columns)
```

```
[[ 2.98694625e-01 -4.30394966e-01  1.09644382e-01  2.46900736e-02
  -2.64234913e-01  7.78800429e-01 -3.01588637e-02 -2.00382806e-01
   1.49932197e-01  1.18572290e+00  1.20799097e+00 -1.62919078e+00
   1.35206569e-02  2.48333720e-01  2.07717288e-01  9.28356206e-03
  -1.18581928e-03  1.18795736e-01  2.75949378e-02  7.78068318e-02
  -2.70332635e-02  2.77204226e-02 -4.98475836e-02  3.04884331e-02
   5.29469889e-02  2.90608471e-03  3.85179112e-02  2.24822865e-02
   9.57659693e-03  3.08828624e-02 -1.22904721e-01 -3.77978078e-02
  -7.64037745e-02 -7.91030693e-02 -4.55859229e-02  1.05438522e-02
  -5.40842029e-02 -2.06062340e-02  7.35075435e-04  2.56287412e-02
  -1.61668610e-01 -6.20655051e-03 -8.72958973e-03 -1.02229971e-01
  -3.70468309e-02  2.13856053e-02  7.36384946e-02  6.35133810e-02
   7.29805728e-02  6.23112555e-02  2.83255414e-02  2.07225806e-02
  -3.97263080e-02  1.01257787e-01  1.22255755e-02 -9.26359812e-03
  -1.17207566e-01  6.30692137e-02  1.56931666e-02  3.65262915e-02
   1.39179288e-02 -1.87863481e-02  5.46955406e-02  8.12112839e-02
  -1.25994635e-01 -5.93366268e-02 -2.90299103e-02 -2.44137143e-02
   1.89662403e-03  1.24648106e-02 -4.14733834e-02 -4.08677166e-02
   1.80467183e-02  1.34858148e-02  2.69967271e-02  6.09186269e-03
   1.07715285e-03  3.00104720e-03  3.13016361e-03  5.01863798e-02
  -1.99023094e-02 -1.58641044e-02 -4.19252176e-02  3.31275147e-02
  -5.52844586e-02  1.86380754e-02  4.06191671e-02  8.21581320e-02
   8.43341839e-04  1.87954293e-02 -6.19742190e-02 -5.01738642e-02
  -7.40474826e-02 -9.51555309e-03  1.87842900e-02  2.67928526e-02
   1.66944022e-02  2.70807773e-02 -2.21068140e-02 -1.04529795e-02
  -3.44424042e-02  5.86390288e-02 -6.32264668e-02  1.81139359e-02
   9.79289392e-02  8.19919654e-02 -2.90221738e-02  2.77129023e-02
  -2.30543942e-02 -2.86834480e-02 -5.57892426e-02 -3.11505378e-02
   3.50338351e-02 -3.20131679e-02]]
Index(['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine',
       'WindGustSpeed', 'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am',
       'Humidity3pm',
       ...
       'WindDir3pm_NNW', 'WindDir3pm_NW', 'WindDir3pm_S', 'WindDir3pm_SE',
       'WindDir3pm_SSE', 'WindDir3pm_SSW', 'WindDir3pm_SW', 'WindDir3pm_W',
       'WindDir3pm_WNW', 'WindDir3pm_WSW'],
      dtype='object', length=114)
```

In [73]:

```
# Credit: https://stackoverflow.com/questions/34649969/how-to-find-the-feat
# Create a dictionary to pair the column headers with its relevant coeffici

coef_dict = {}
for coef, feat in zip(lr_best.coef_[0,:],X_train_resampled_scaled_df.column
    coef_dict[feat] = coef
coef_dict
```

Out[73]:

```
{'MinTemp': 0.29869462480686215,
 'MaxTemp': -0.43039496567083335,
 'Rainfall': 0.10964438161645215,
 'Evaporation': 0.024690073584378934,
 'Sunshine': -0.26423491319738746,
 'WindGustSpeed': 0.7788004292599717,
```

```
'WindSpeed9am': -0.03015886368146996,
'WindSpeed3pm': -0.20038280642712103,
'Humidity9am': 0.1499321966262893,
'Humidity3pm': 1.1857228985839068,
'Pressure9am': 1.2079909697573015,
'Pressure3pm': -1.6291907752287695,
'Cloud9am': 0.013520656850079422,
'Cloud3pm': 0.24833371992596276,
'Temp9am': 0.20771728805380213,
'Temp3pm': 0.009283562061695585,
'RainToday': -0.0011858192808979467,
'Location_Adelaide': 0.11879573605924158,
'Location_Albany': 0.027594937840594927,
'Location_Albury': 0.07780683176135311,
'Location_AliceSprings': -0.02703326349546255,
'Location_BadgerysCreek': 0.02772042261551849,
'Location_Ballarat': -0.049847583580396065,
'Location_Bendigo': 0.030488433059040233,
'Location_Brisbane': 0.05294698891050664,
'Location_Cairns': 0.0029060847101391365,
'Location_Canberra': 0.03851791122709859,
'Location_Cobar': 0.022482286450269128,
'Location_CoffsHarbour': 0.009576596925515532,
'Location_Dartmoor': 0.03088286240683609,
'Location_Darwin': -0.12290472126057628,
'Location_GoldCoast': -0.037797807825104994,
'Location_Hobart': -0.0764037744586181,
'Location_Katherine': -0.07910306933539243,
'Location_Launceston': -0.045585922896354485,
'Location_Melbourne': 0.01054385222286126,
'Location_MelbourneAirport': -0.054084202909823666,
'Location_Mildura': -0.02060623398764053,
'Location_Moree': 0.0007350754345888617,
'Location_MountGambier': 0.025628741248558982,
'Location_MountGinini': -0.16166860954861087,
'Location_Newcastle': -0.00620655051350016,
'Location_Nhil': -0.008729589732659568,
'Location_NorahHead': -0.1022299711045101,
'Location_NorfolkIsland': -0.037046830908105045,
'Location_Nuriootpa': 0.021385605321269926,
'Location_PearceRAAF': 0.07363849456798978,
'Location_Penrith': 0.06351338097902647,
'Location_Perth': 0.07298057278437195,
'Location_PerthAirport': 0.062311255539962224,
'Location_Portland': 0.028325541448905176,
'Location_Richmond': 0.020722580596679148,
'Location_Sale': -0.03972630800045375,
'Location_SalmonGums': 0.1012577874390231,
'Location_Sydney': 0.012225575488476371,
'Location_SydneyAirport': -0.009263598116329102,
'Location_Townsville': -0.11720756601130876,
'Location_Tuggeranong': 0.0630692136811133,
'Location_Uluru': 0.01569316661976984,
```

```
'Location_WaggaWagga': 0.03652629154395293,
'Location_Walpole': 0.013917928845141245,
'Location_Watsonia': -0.018786348149100667,
'Location_Williamtown': 0.05469554061532822,
'Location_Witchcliffe': 0.08121128393779953,
'Location_Wollongong': -0.125994634726998,
'Location_Woomera': -0.059336626831893655,
'WindGustDir_E': -0.0290299103097l103,
'WindGustDir_ENE': -0.024413714288822393,
'WindGustDir_ESE': 0.0018966240274322578,
'WindGustDir_N': 0.012464810599040638,
'WindGustDir_NE': -0.04147338339691926,
'WindGustDir_NNE': -0.04086771660429363,
'WindGustDir_NNW': 0.0180467182846937,
'WindGustDir_NW': 0.013485814838324717,
'WindGustDir_S': 0.026996727138188414,
'WindGustDir_SE': 0.006091862686492851,
'WindGustDir_SSE': 0.0010771528457470202,
'WindGustDir_SSW': 0.0030010471988638995,
'WindGustDir_SW': 0.0031301636128424027,
'WindGustDir_W': 0.05018637984093656,
'WindGustDir_WNW': -0.01990230940398417,
'WindGustDir_WSW': -0.01586410443328936,
'WindDir9am_E': -0.04192521764646852,
'WindDir9am_ENE': 0.03312751465121465,
'WindDir9am_ESE': -0.05528445858547368,
'WindDir9am_N': 0.01863807536494281,
'WindDir9am_NE': 0.040619167050716734,
'WindDir9am_NNE': 0.08215813201062991,
'WindDir9am_NNW': 0.0008433418390891563,
'WindDir9am_NW': 0.0187954293156464,
'WindDir9am_S': -0.0619742190313133,
'WindDir9am_SE': -0.050173864164514394,
'WindDir9am_SSE': -0.07404748259611575,
'WindDir9am_SSW': -0.009515553086083841,
'WindDir9am_SW': 0.018784289976742367,
'WindDir9am_W': 0.026792852608973138,
'WindDir9am_WNW': 0.016694402207246137,
'WindDir9am_WSW': 0.02708077728062469,
'WindDir3pm_E': -0.02210681395939944,
'WindDir3pm_ENE': -0.010452979540971411,
'WindDir3pm_ESE': -0.03444240419676403,
'WindDir3pm_N': 0.058639028834843246,
'WindDir3pm_NE': -0.06322646675712619,
'WindDir3pm_NNE': 0.018113935916101567,
'WindDir3pm_NNW': 0.09792893924316423,
'WindDir3pm_NW': 0.08199196544564676,
'WindDir3pm_S': -0.029022173840670365,
'WindDir3pm_SE': 0.027712902252948944,
'WindDir3pm_SSE': -0.023054394191851554,
'WindDir3pm_SSW': -0.028683448035253516,
'WindDir3pm_SW': -0.05578924258890517,
'WindDir3pm_W': -0.031150537820773594,
```

```
 'WindDir3pm_WNW': 0.03503383508209787,
 'WindDir3pm_WSW': -0.03201316788621604}
```

In [74]:

```
# Sort the list and format it as a numbered list
sorted_coef = sorted(coef_dict.items(), key=lambda x: x[1], reverse=True)

for i in enumerate(sorted_coef,start=1):
    print(i[0], i[1])
```

```
1 ('Pressure9am', 1.2079909697573015)
2 ('Humidity3pm', 1.1857228985839068)
3 ('WindGustSpeed', 0.7788004292599717)
4 ('MinTemp', 0.29869462480686215)
5 ('Cloud3pm', 0.24833371992596276)
6 ('Temp9am', 0.20771728805380213)
7 ('Humidity9am', 0.1499321966262893)
8 ('Location_Adelaide', 0.11879573605924158)
9 ('Rainfall', 0.10964438161645215)
10 ('Location_SalmonGums', 0.1012577874390231)
11 ('WindDir3pm_NNW', 0.09792893924316423)
12 ('WindDir9am_NNE', 0.08215813201062991)
13 ('WindDir3pm_NW', 0.08199196544564676)
14 ('Location_Witchcliffe', 0.08121128393779953)
15 ('Location_Albury', 0.07780683176135311)
16 ('Location_PearceRAAF', 0.07363849456798978)
17 ('Location_Perth', 0.07298057278437195)
18 ('Location_Penrith', 0.06351338097902647)
19 ('Location_Tuggeranong', 0.0630692136811133)
20 ('Location_PerthAirport', 0.062311255539962224)
21 ('WindDir3pm_N', 0.058639028834843246)
22 ('Location_Williamtown', 0.05469554061532822)
23 ('Location_Brisbane', 0.05294698891050664)
24 ('WindGustDir_W', 0.05018637984093656)
25 ('WindDir9am_NE', 0.040619167050716734)
26 ('Location_Canberra', 0.03851791122709859)
27 ('Location_WaggaWagga', 0.03652629154395293)
28 ('WindDir3pm_WNW', 0.03503383508209787)
29 ('WindDir9am_ENE', 0.03312751465121465)
30 ('Location_Dartmoor', 0.03088286240683609)
31 ('Location_Bendigo', 0.030488433059040233)
32 ('Location_Portland', 0.028325541448905176)
33 ('Location_BadgerysCreek', 0.02772042261551849)
34 ('WindDir3pm_SE', 0.027712902252948944)
35 ('Location_Albany', 0.027594937840594927)
36 ('WindDir9am_WSW', 0.02708077728062469)
37 ('WindGustDir_S', 0.026996727138188414)
38 ('WindDir9am_W', 0.026792852608973138)
39 ('Location_MountGambier', 0.025628741248558982)
40 ('Evaporation', 0.024690073584378934)
41 ('Location_Cobar', 0.022482286450269128)
42 ('Location_Nuriootpa', 0.021385605321269926)
43 ('Location_Richmond', 0.020722580596679148)
```

```
44 ('WindDir9am_NW', 0.0187954293156464)
45 ('WindDir9am_SW', 0.018784289976742367)
46 ('WindDir9am_N', 0.01863807536494281)
47 ('WindDir3pm_NNE', 0.018113935916101567)
48 ('WindGustDir_NNW', 0.0180467182846937)
49 ('WindDir9am_WNW', 0.016694402207246137)
50 ('Location_Uluru', 0.01569316661976984)
51 ('Location_Walpole', 0.013917928845141245)
52 ('Cloud9am', 0.013520656850079422)
53 ('WindGustDir_NW', 0.013485814838324717)
54 ('WindGustDir_N', 0.012464810599040638)
55 ('Location_Sydney', 0.012225575488476371)
56 ('Location_Melbourne', 0.01054385222286126)
57 ('Location_CoffsHarbour', 0.009576596925515532)
58 ('Temp3pm', 0.009283562061695585)
59 ('WindGustDir_SE', 0.006091862686492851)
60 ('WindGustDir_SW', 0.0031301636128424027)
61 ('WindGustDir_SSW', 0.0030010471988638995)
62 ('Location_Cairns', 0.0029060847101391365)
63 ('WindGustDir_ESE', 0.0018966240274322578)
64 ('WindGustDir_SSE', 0.0010771528457470202)
65 ('WindDir9am_NNW', 0.0008433418390891563)
66 ('Location_Moree', 0.0007350754345888617)
67 ('RainToday', -0.0011858192808979467)
68 ('Location_Newcastle', -0.00620655051350016)
69 ('Location_Nhil', -0.008729589732659568)
70 ('Location_SydneyAirport', -0.009263598116329102)
71 ('WindDir9am_SSW', -0.009515553086083841)
72 ('WindDir3pm_ENE', -0.010452979540971411)
73 ('WindGustDir_WSW', -0.01586410443328936)
74 ('Location_Watsonia', -0.018786348149100667)
75 ('WindGustDir_WNW', -0.01990230940398417)
76 ('Location_Mildura', -0.02060623398764053)
77 ('WindDir3pm_E', -0.02210681395939944)
78 ('WindDir3pm_SSE', -0.023054394191851554)
79 ('WindGustDir_ENE', -0.024413714288822393)
80 ('Location_AliceSprings', -0.02703326349546255)
81 ('WindDir3pm_SSW', -0.028683448035253516)
82 ('WindDir3pm_S', -0.029022173840670365)
83 ('WindGustDir_E', -0.02902991030971103)
84 ('WindSpeed9am', -0.03015886368146996)
85 ('WindDir3pm_W', -0.031150537820773594)
86 ('WindDir3pm_WSW', -0.03201316788621604)
87 ('WindDir3pm_ESE', -0.03444240419676403)
88 ('Location_NorfolkIsland', -0.037046830908105045)
89 ('Location_GoldCoast', -0.037797807825104994)
90 ('Location_Sale', -0.03972630800045375)
91 ('WindGustDir_NNE', -0.04086771660429363)
92 ('WindGustDir_NE', -0.04147338339691926)
93 ('WindDir9am_E', -0.04192521764646852)
94 ('Location_Launceston', -0.045585922896354485)
95 ('Location_Ballarat', -0.049847583580396065)
96 ('WindDir9am_SE', -0.050173864164514394)
```

```
97 ('Location_MelbourneAirport', -0.05408420290982366 6)
98 ('WindDir9am_ESE', -0.05528445858547368)
99 ('WindDir3pm_SW', -0.05578924258890517)
100 ('Location_Woomera', -0.059336626831893655)
101 ('WindDir9am_S', -0.0619742190313133)
102 ('WindDir3pm_NE', -0.06322646675712619)
103 ('WindDir9am_SSE', -0.07404748259611575)
104 ('Location_Hobart', -0.0764037744586181)
105 ('Location_Katherine', -0.07910306933539243)
106 ('Location_NorahHead', -0.1022299711045101)
107 ('Location_Townsville', -0.11720756601130876)
108 ('Location_Darwin', -0.12290472126057628)
109 ('Location_Wollongong', -0.125994634726998)
110 ('Location_MountGinini', -0.16166860954861087)
111 ('WindSpeed3pm', -0.20038280642712103)
112 ('Sunshine', -0.26423491319738746)
113 ('MaxTemp', -0.43039496567083335)
114 ('Pressure3pm', -1.6291907752287695)
```

In [75]:

```python
# Create a dictionary linking the variables together
odds_ratio = {}
for coef, feat in zip(lr_best.coef_[0,:],X_train_resampled_scaled_df.column
    odds_ratio[feat] = math.pow(math.e,coef)
odds_ratio
```

Out[75]:

```
{'MinTemp': 1.3480978849567584,
 'MaxTemp': 0.6502522166946768,
 'Rainfall': 1.1158811720322361,
 'Evaporation': 1.0249973975221616,
 'Sunshine': 0.767793153725344,
 'WindGustSpeed': 2.178857004107454,
 'WindSpeed9am': 0.9702913772436258,
 'WindSpeed3pm': 0.8184173976646817,
 'Humidity9am': 1.161755469117522,
 'Humidity3pm': 3.2730520511451826,
 'Pressure9am': 3.3467541636053464,
 'Pressure3pm': 0.19608818936074196,
 'Cloud9am': 1.013612474274870 2,
 'Cloud3pm': 1.2818876522731042,
 'Temp9am': 1.230865140075883,
 'Temp3pm': 1.0093267879839354,
 'RainToday': 0.9988148835249578,
 'Location_Adelaide': 1.1261398650266843,
 'Location_Albany': 1.0279792046005727,
 'Location_Albury': 1.0809138402795362,
 'Location_AliceSprings': 0.9733288646655827,
 'Location_BadgerysCreek': 1.0281082084335273,
 'Location_Ballarat': 0.9513744185332733,
 'Location_Bendigo': 1.0309579649498999,
 'Location_Brisbane': 1.0543737501366142,
 'Location_Cairns': 1.002910311467757,
 'Location_Canberra': 1.039269342780495,
```

```
 'Location_Cobar': 1.0227369177027397,
 'Location_CoffsHarbour': 1.009622599261194,
 'Location_Dartmoor': 1.0313646852335552,
 'Location_Darwin': 0.884347918101725,
 'Location_GoldCoast': 0.9629076135945033,
 'Location_Hobart': 0.9264420573883629,
 'Location_Katherine': 0.9239446891722006,
 'Location_Launceston': 0.9554375050872553,
 'Location_Melbourne': 1.0105996345137365,
 'Location_MelbourneAirport': 0.9473523333201846,
 'Location_Mildura': 0.9796046236410991,
 'Location_Moree': 1.0007353456687462,
 'Location_MountGambier': 1.0259599811374636,
 'Location_MountGinini': 0.850723079337614,
 'Location_Newcastle': 0.9938126703355229,
 'Location_Nhil': 0.9913084025032038,
 'Location_NorahHead': 0.9028219048427744,
 'Location_NorfolkIsland': 0.9636310065772348,
 'Location_Nuriootpa': 1.0216159162280989,
 'Location_PearceRAAF': 1.076417604340212,
 'Location_Penrith': 1.0655737441259927,
 'Location_Perth': 1.0757096386686031,
 'Location_PerthAirport': 1.064293560445831,
 'Location_Portland': 1.0287305243417588,
 'Location_Richmond': 1.020938784119075,
 'Location_Sale': 0.9610524355234273,
 'Location_SalmonGums': 1.1065618627473688,
 'Location_Sydney': 1.012300613318244,
 'Location_SydneyAirport': 0.9907791768235165,
 'Location_Townsville': 0.8894005646751094,
 'Location_Tuggeranong': 1.0651005562104154,
 'Location_Uluru': 1.015816951034504,
 'Location_WaggaWagga': 1.0372015732900246,
 'Location_Walpole': 1.0140152341220519,
 'Location_Watsonia': 0.9813890154252112,
 'Location_Williamtown': 1.0562189899333894,
 'Location_Witchcliffe': 1.0846000309251744,
 'Location_Wollongong': 0.8816195769004408,
 'Location_Woomera': 0.9423894821706464,
 'WindGustDir_E': 0.9713874095338174,
 'WindGustDir_ENE': 0.9758818899485817,
 'WindGustDir_ESE': 1.0018984237564061,
 'WindGustDir_N': 1.012542820138428,
 'WindGustDir_NE': 0.9593748703017724,
 'WindGustDir_NNE': 0.9599561078027521,
 'WindGustDir_NNW': 1.0182105443287268,
 'WindGustDir_NW': 1.0135771585923652,
 'WindGustDir_S': 1.0273644403361437,
 'WindGustDir_SE': 1.006110455818416,
 'WindGustDir_SSE': 1.0010777331832257,
 'WindGustDir_SSW': 1.0030055548491046,
 'WindGustDir_SW': 1.0031350676904842,
 'WindGustDir_W': 1.0514670503761183,
```

```
 'WindGustDir_WNW': 0.9802944341767567,
 'WindGustDir_WSW': 0.9842610676828084,
 'WindDir9am_E': 0.9589414897926647,
 'WindDir9am_ENE': 1.0336823404809454,
 'WindDir9am_ESE': 0.9462159504167157,
 'WindDir9am_N': 1.0188128484141403,
 'WindDir9am_NE': 1.041455409477669,
 'WindDir9am_NNE': 1.0856274687112317,
 'WindDir9am_NNW': 1.0008436975518067,
 'WindDir9am_NW': 1.018973175254566,
 'WindDir9am_S': 0.9399071181947156,
 'WindDir9am_SE': 0.9510640541679657,
 'WindDir9am_SSE': 0.928627599145198,
 'WindDir9am_SSW': 0.9905295765313237,
 'WindDir9am_SW': 1.018961824630252,
 'WindDir9am_W': 1.0271550082441976,
 'WindDir9am_WNW': 1.0168345324505967,
 'WindDir9am_WSW': 1.0274507940926598,
 'WindDir3pm_E': 0.9781357509187417,
 'WindDir3pm_ENE': 0.9896014629889949,
 'WindDir3pm_ESE': 0.9661439839228346,
 'WindDir3pm_N': 1.0603924005668284,
 'WindDir3pm_NE': 0.9387308582815918,
 'WindDir3pm_NNE': 1.0182789883300873,
 'WindDir3pm_NNW': 1.1028844105229478,
 'WindDir3pm_NW': 1.085447088710882,
 'WindDir3pm_S': 0.971394924671508,
 'WindDir3pm_SE': 1.028100476716112,
 'WindDir3pm_SSE': 0.9772093278159462,
 'WindDir3pm_SSW': 0.9717240169326193,
 'WindDir3pm_SW': 0.9457384362720439,
 'WindDir3pm_W': 0.9693296413203617,
 'WindDir3pm_WNW': 1.0356547496702946,
 'WindDir3pm_WSW': 0.9684938289786912}
```

In [76]:

```
# Sort the dictionary and format it as a numbered list
sorted_odds = sorted(odds_ratio.items(), key=lambda x: x[1], reverse=True)

for i in enumerate(sorted_odds,start=1):
    print(i[0], i[1])
```

```
1 ('Pressure9am', 3.3467541636053464)
2 ('Humidity3pm', 3.2730520511451826)
3 ('WindGustSpeed', 2.178857004107454)
4 ('MinTemp', 1.3480978849567584)
5 ('Cloud3pm', 1.2818876522731042)
6 ('Temp9am', 1.230865140075883)
7 ('Humidity9am', 1.161755469117522)
8 ('Location_Adelaide', 1.1261398650266843)
9 ('Rainfall', 1.1158811720322361)
10 ('Location_SalmonGums', 1.1065618627473688)
11 ('WindDir3pm_NNW', 1.1028844105229478)
```

```
12 ('WindDir9am_NNE', 1.0856274687112317)
13 ('WindDir3pm_NW', 1.085447088710882)
14 ('Location_Witchcliffe', 1.0846000309251744)
15 ('Location_Albury', 1.0809138402795362)
16 ('Location_PearceRAAF', 1.076417604340212)
17 ('Location_Perth', 1.0757096386686031)
18 ('Location_Penrith', 1.0655737441259927)
19 ('Location_Tuggeranong', 1.0651005562104154)
20 ('Location_PerthAirport', 1.064293560445831)
21 ('WindDir3pm_N', 1.0603924005668284)
22 ('Location_Williamtown', 1.0562189899333894)
23 ('Location_Brisbane', 1.0543737501366142)
24 ('WindGustDir_W', 1.0514670503761183)
25 ('WindDir9am_NE', 1.041455409477669)
26 ('Location_Canberra', 1.039269342780495)
27 ('Location_WaggaWagga', 1.0372015732900246)
28 ('WindDir3pm_WNW', 1.0356547496702946)
29 ('WindDir9am_ENE', 1.0336823404809454)
30 ('Location_Dartmoor', 1.0313646852335552)
31 ('Location_Bendigo', 1.0309579649498999)
32 ('Location_Portland', 1.0287305243417588)
33 ('Location_BadgerysCreek', 1.0281082084335273)
34 ('WindDir3pm_SE', 1.028100476716112)
35 ('Location_Albany', 1.0279792046005727)
36 ('WindDir9am_WSW', 1.0274507940926598)
37 ('WindGustDir_S', 1.0273644403361437)
38 ('WindDir9am_W', 1.0271550082441976)
39 ('Location_MountGambier', 1.0259599811374636)
40 ('Evaporation', 1.0249973975221616)
41 ('Location_Cobar', 1.0227369177027397)
42 ('Location_Nuriootpa', 1.0216159162280989)
43 ('Location_Richmond', 1.020938784119075)
44 ('WindDir9am_NW', 1.018973175254566)
45 ('WindDir9am_SW', 1.018961824630252)
46 ('WindDir9am_N', 1.0188128484141403)
47 ('WindDir3pm_NNE', 1.0182789883300873)
48 ('WindGustDir_NNW', 1.0182105443287268)
49 ('WindDir9am_WNW', 1.0168345324505967)
50 ('Location_Uluru', 1.015816951034504)
51 ('Location_Walpole', 1.0140152341220519)
52 ('Cloud9am', 1.0136124742748702)
53 ('WindGustDir_NW', 1.0135771585923652)
54 ('WindGustDir_N', 1.012542820138428)
55 ('Location_Sydney', 1.012300613318244)
56 ('Location_Melbourne', 1.0105996345137365)
57 ('Location_CoffsHarbour', 1.009622599261194)
58 ('Temp3pm', 1.0093267879839354)
59 ('WindGustDir_SE', 1.006110455818416)
60 ('WindGustDir_SW', 1.0031350676904842)
61 ('WindGustDir_SSW', 1.0030055548491046)
62 ('Location_Cairns', 1.002910311467757)
63 ('WindGustDir_ESE', 1.0018984237564061)
64 ('WindGustDir_SSE', 1.0010777331832257)
```

```
65 ('WindDir9am_NNW', 1.0008436975518067)
66 ('Location_Moree', 1.0007353456687462)
67 ('RainToday', 0.9988148835249578)
68 ('Location_Newcastle', 0.9938126703355229)
69 ('Location_Nhil', 0.9913084025032038)
70 ('Location_SydneyAirport', 0.9907791768235165)
71 ('WindDir9am_SSW', 0.9905295765313237)
72 ('WindDir3pm_ENE', 0.9896014629889949)
73 ('WindGustDir_WSW', 0.9842610676828084)
74 ('Location_Watsonia', 0.9813890154252112)
75 ('WindGustDir_WNW', 0.9802944341767567)
76 ('Location_Mildura', 0.9796046236410991)
77 ('WindDir3pm_E', 0.9781357509187417)
78 ('WindDir3pm_SSE', 0.9772093278159462)
79 ('WindGustDir_ENE', 0.9758818899485817)
80 ('Location_AliceSprings', 0.9733288646655827)
81 ('WindDir3pm_SSW', 0.9717240169326193)
82 ('WindDir3pm_S', 0.971394924671508)
83 ('WindGustDir_E', 0.9713874095338174)
84 ('WindSpeed9am', 0.9702913772436258)
85 ('WindDir3pm_W', 0.9693296413203617)
86 ('WindDir3pm_WSW', 0.9684938289786912)
87 ('WindDir3pm_ESE', 0.9661439839228346)
88 ('Location_NorfolkIsland', 0.9636310065772348)
89 ('Location_GoldCoast', 0.9629076135945033)
90 ('Location_Sale', 0.961052435523427 3)
91 ('WindGustDir_NNE', 0.9599561078027521)
92 ('WindGustDir_NE', 0.9593748703017724)
93 ('WindDir9am_E', 0.9589414897926647)
94 ('Location_Launceston', 0.9554375050872553)
95 ('Location_Ballarat', 0.9513744185332733)
96 ('WindDir9am_SE', 0.9510640541679657)
97 ('Location_MelbourneAirport', 0.9473523333201846)
98 ('WindDir9am_ESE', 0.9462159504167157)
99 ('WindDir3pm_SW', 0.9457384362720439)
100 ('Location_Woomera', 0.9423894821706464)
101 ('WindDir9am_S', 0.9399071181947156)
102 ('WindDir3pm_NE', 0.9387308582815918)
103 ('WindDir9am_SSE', 0.928627599145198)
104 ('Location_Hobart', 0.9264420573883629)
105 ('Location_Katherine', 0.9239446891722006)
106 ('Location_NorahHead', 0.9028219048427744)
107 ('Location_Townsville', 0.8894005646751094)
108 ('Location_Darwin', 0.884347918101725)
109 ('Location_Wollongong', 0.8816195769004408)
110 ('Location_MountGinini', 0.850723079337614)
111 ('WindSpeed3pm', 0.8184173976646817)
112 ('Sunshine', 0.767793153725344)
113 ('MaxTemp', 0.650252216694676 8)
114 ('Pressure3pm', 0.19608818936074196)
```

As we can see above, we have the odds ratios for all of our features. However, we need to determine if these variables are statistically significant. To do this, we will be running another logistic regression model through Statsmodels.

Logistic Regression using Statsmodels

First I am going to build a baseline model. I also need to add a constant to my dataset.

In [155…

```
# Add a constant to the X_train_resampled data, and fit the Logit model
X_train_resampled_sm = sm.add_constant(X_train_resampled_scaled_df)
logit_model=sm.Logit(y_train_resampled,X_train_resampled_sm)
result=logit_model.fit()
print(result.summary())
```

```
Warning: Maximum number of iterations has been exceeded.
         Current function value: 0.446867
         Iterations: 35
                           Logit Regression Results
=============================================================================
==
Dep. Variable:          RainTomorrow   No. Observations:               1653
86
Model:                         Logit   Df Residuals:                   1652
75
Method:                          MLE   Df Model:                          1
10
Date:               Sun, 09 Jan 2022   Pseudo R-squ.:                  0.35
53
Time:                       01:40:23   Log-Likelihood:                -7390
6.
converged:                     False   LL-Null:                   -1.1464e+
05
Covariance Type:           nonrobust   LLR p-value:                     0.0
00
=============================================================================
=================
                         coef    std err          z      P>|z|          [
0.025      0.975]
-----------------------------------------------------------------------------
-----------------
const                  0.0584      0.007      8.636      0.000
0.045      0.072
MinTemp                0.2989      0.022     13.423      0.000
0.255      0.343
MaxTemp               -0.4313      0.032    -13.458      0.000          -
0.494     -0.368
Rainfall               0.1097      0.012      9.153      0.000
0.086      0.133
Evaporation            0.0248      0.009      2.686      0.007
0.007      0.043
Sunshine              -0.2642      0.011    -23.841      0.000          -
```

```
0.286      -0.243
WindGustSpeed                 0.7790       0.011      67.832       0.000       
0.757       0.802
WindSpeed9am                 -0.0301       0.010      -3.070       0.002      -
0.049      -0.011
WindSpeed3pm                 -0.2005       0.010     -20.065       0.000      -
0.220      -0.181
Humidity9am                   0.1499       0.013      11.554       0.000       
0.124       0.175
Humidity3pm                   1.1859       0.016      72.456       0.000       
1.154       1.218
Pressure9am                   1.2113       0.032      37.278       0.000       
1.148       1.275
Pressure3pm                  -1.6325       0.033     -50.164       0.000      -
1.696      -1.569
Cloud9am                      0.0135       0.009       1.444       0.149      -
0.005       0.032
Cloud3pm                      0.2483       0.009      26.444       0.000       
0.230       0.267
Temp9am                       0.2083       0.031       6.775       0.000       
0.148       0.269
Temp3pm                       0.0094       0.032       0.296       0.767      -
0.053       0.071
RainToday                    -0.0011       0.010      -0.105       0.916      -
0.021       0.019
Location_Adelaide             0.1188     2.1e+05    5.67e-07       1.000    -4.1
1e+05    4.11e+05
Location_Albany               0.0276     2.2e+05    1.25e-07       1.000    -4.3
2e+05    4.32e+05
Location_Albury               0.0778    2.01e+05    3.86e-07       1.000    -3.9
5e+05    3.95e+05
Location_AliceSprings        -0.0271    1.85e+05   -1.46e-07       1.000    -3.6
3e+05    3.63e+05
Location_BadgerysCreek        0.0277       2e+05    1.38e-07       1.000    -3.9
2e+05    3.92e+05
Location_Ballarat            -0.0498    2.11e+05   -2.36e-07       1.000    -4.1
3e+05    4.13e+05
Location_Bendigo              0.0305       2e+05    1.52e-07       1.000    -3.9
3e+05    3.93e+05
Location_Brisbane             0.0529    2.19e+05    2.42e-07       1.000    -4.2
9e+05    4.29e+05
Location_Cairns               0.0028    2.24e+05    1.26e-08       1.000    -4.3
8e+05    4.38e+05
Location_Canberra             0.0386    2.13e+05    1.81e-07       1.000    -4.1
8e+05    4.18e+05
Location_Cobar                0.0225    1.91e+05    1.18e-07       1.000    -3.7
4e+05    3.74e+05
Location_CoffsHarbour         0.0095    2.16e+05    4.42e-08       1.000    -4.2
3e+05    4.23e+05
Location_Dartmoor             0.0310    2.16e+05    1.44e-07       1.000    -4.2
3e+05    4.23e+05
Location_Darwin              -0.1230    2.26e+05   -5.45e-07       1.000    -4.4
2e+05    4.42e+05
```

```
Location_GoldCoast           -0.0379   2.11e+05   -1.8e-07      1.000   -4.1
3e+05    4.13e+05
Location_Hobart              -0.0764   2.15e+05  -3.55e-07      1.000   -4.2
1e+05    4.21e+05
Location_Katherine           -0.0792   1.48e+05  -5.36e-07      1.000    -2.
9e+05     2.9e+05
Location_Launceston          -0.0456   2.11e+05  -2.16e-07      1.000   -4.1
3e+05    4.13e+05
Location_Melbourne            0.0106    1.9e+05   5.57e-08      1.000   -3.7
2e+05    3.72e+05
Location_MelbourneAirport    -0.0540   2.03e+05  -2.66e-07      1.000   -3.9
9e+05    3.99e+05
Location_Mildura             -0.0206   1.92e+05  -1.07e-07      1.000   -3.7
6e+05    3.76e+05
Location_Moree                0.0007   1.88e+05   3.81e-09      1.000   -3.6
9e+05    3.69e+05
Location_MountGambier         0.0257   2.15e+05    1.2e-07      1.000   -4.2
1e+05    4.21e+05
Location_MountGinini         -0.1618   2.18e+05  -7.42e-07      1.000   -4.2
8e+05    4.28e+05
Location_Newcastle           -0.0062   2.14e+05  -2.92e-08      1.000   -4.1
9e+05    4.19e+05
Location_Nhil                -0.0087    1.4e+05  -6.21e-08      1.000   -2.7
5e+05    2.75e+05
Location_NorahHead           -0.1023   2.08e+05  -4.93e-07      1.000   -4.0
7e+05    4.07e+05
Location_NorfolkIsland       -0.0371   2.24e+05  -1.66e-07      1.000   -4.3
9e+05    4.39e+05
Location_Nuriootpa            0.0214   2.02e+05   1.06e-07      1.000   -3.9
7e+05    3.97e+05
Location_PearceRAAF           0.0737   1.94e+05   3.79e-07      1.000   -3.8
1e+05    3.81e+05
Location_Penrith              0.0635   2.05e+05    3.1e-07      1.000   -4.0
2e+05    4.02e+05
Location_Perth                0.0730   2.11e+05   3.46e-07      1.000   -4.1
3e+05    4.13e+05
Location_PerthAirport         0.0623   2.04e+05   3.05e-07      1.000      -
4e+05       4e+05
Location_Portland             0.0284   2.23e+05   1.27e-07      1.000   -4.3
8e+05    4.38e+05
Location_Richmond             0.0207   1.98e+05   1.05e-07      1.000   -3.8
8e+05    3.88e+05
Location_Sale                -0.0397   2.02e+05  -1.96e-07      1.000   -3.9
7e+05    3.97e+05
Location_SalmonGums           0.1013   1.99e+05   5.08e-07      1.000   -3.9
1e+05    3.91e+05
Location_Sydney               0.0122   2.18e+05   5.61e-08      1.000   -4.2
6e+05    4.26e+05
Location_SydneyAirport       -0.0093    2.1e+05  -4.43e-08      1.000   -4.1
1e+05    4.11e+05
Location_Townsville          -0.1173   2.02e+05  -5.79e-07      1.000   -3.9
7e+05    3.97e+05
Location_Tuggeranong          0.0631   2.03e+05   3.11e-07      1.000   -3.9
```

```
8e+05    3.98e+05
Location_Uluru                 0.0157   1.33e+05   1.18e-07      1.000    -2.
6e+05     2.6e+05
Location_WaggaWagga            0.0366   1.98e+05   1.85e-07      1.000   -3.8
8e+05    3.88e+05
Location_Walpole               0.0139   2.16e+05   6.45e-08      1.000   -4.2
3e+05    4.23e+05
Location_Watsonia             -0.0187   2.07e+05  -9.03e-08      1.000   -4.0
7e+05    4.07e+05
Location_Williamtown           0.0547   1.92e+05   2.84e-07      1.000   -3.7
7e+05    3.77e+05
Location_Witchcliffe           0.0813   2.16e+05   3.75e-07      1.000   -4.2
4e+05    4.24e+05
Location_Wollongong           -0.1260   2.06e+05  -6.13e-07      1.000   -4.0
3e+05    4.03e+05
Location_Woomera              -0.0593   1.84e+05  -3.22e-07      1.000   -3.6
1e+05    3.61e+05
WindGustDir_E                 -0.0290        nan        nan        nan
nan         nan
WindGustDir_ENE               -0.0244        nan        nan        nan
nan         nan
WindGustDir_ESE                0.0019        nan        nan        nan
nan         nan
WindGustDir_N                  0.0124        nan        nan        nan
nan         nan
WindGustDir_NE                -0.0415        nan        nan        nan
nan         nan
WindGustDir_NNE               -0.0409        nan        nan        nan
nan         nan
WindGustDir_NNW                0.0180        nan        nan        nan
nan         nan
WindGustDir_NW                 0.0135        nan        nan        nan
nan         nan
WindGustDir_S                  0.0270        nan        nan        nan
nan         nan
WindGustDir_SE                 0.0061        nan        nan        nan
nan         nan
WindGustDir_SSE                0.0011        nan        nan        nan
nan         nan
WindGustDir_SSW                0.0030        nan        nan        nan
nan         nan
WindGustDir_SW                 0.0032        nan        nan        nan
nan         nan
WindGustDir_W                  0.0502        nan        nan        nan
nan         nan
WindGustDir_WNW               -0.0199        nan        nan        nan
nan         nan
WindGustDir_WSW               -0.0158        nan        nan        nan
nan         nan
WindDir9am_E                  -0.0419        nan        nan        nan
nan         nan
WindDir9am_ENE                 0.0331        nan        nan        nan
nan         nan
```

```
WindDir9am_ESE              -0.0553        nan        nan        nan
nan          nan
WindDir9am_N                 0.0186        nan        nan        nan
nan          nan
WindDir9am_NE                0.0406        nan        nan        nan
nan          nan
WindDir9am_NNE               0.0821        nan        nan        nan
nan          nan
WindDir9am_NNW               0.0008        nan        nan        nan
nan          nan
WindDir9am_NW                0.0188        nan        nan        nan
nan          nan
WindDir9am_S                -0.0620        nan        nan        nan
nan          nan
WindDir9am_SE               -0.0502        nan        nan        nan
nan          nan
WindDir9am_SSE              -0.0740        nan        nan        nan
nan          nan
WindDir9am_SSW              -0.0095        nan        nan        nan
nan          nan
WindDir9am_SW                0.0188        nan        nan        nan
nan          nan
WindDir9am_W                 0.0268        nan        nan        nan
nan          nan
WindDir9am_WNW               0.0167        nan        nan        nan
nan          nan
WindDir9am_WSW               0.0271        nan        nan        nan
nan          nan
WindDir3pm_E                -0.0221        nan        nan        nan
nan          nan
WindDir3pm_ENE              -0.0105        nan        nan        nan
nan          nan
WindDir3pm_ESE              -0.0345        nan        nan        nan
nan          nan
WindDir3pm_N                 0.0586        nan        nan        nan
nan          nan
WindDir3pm_NE               -0.0633        nan        nan        nan
nan          nan
WindDir3pm_NNE               0.0180        nan        nan        nan
nan          nan
WindDir3pm_NNW               0.0979        nan        nan        nan
nan          nan
WindDir3pm_NW                0.0820        nan        nan        nan
nan          nan
WindDir3pm_S                -0.0290        nan        nan        nan
nan          nan
WindDir3pm_SE                0.0277        nan        nan        nan
nan          nan
WindDir3pm_SSE              -0.0230        nan        nan        nan
nan          nan
WindDir3pm_SSW              -0.0286        nan        nan        nan
nan          nan
WindDir3pm_SW               -0.0557        nan        nan        nan
```

```
nan          nan
WindDir3pm_W                    -0.0311        nan        nan        nan
nan          nan
WindDir3pm_WNW                   0.0351        nan        nan        nan
nan          nan
WindDir3pm_WSW                  -0.0319        nan        nan        nan
nan          nan
============================================================================
=================
```

The location and wind data appears to be random and has no relationship to whether it will rain tomorrow. I will now create a DataFrame with the odds ratio and filter our the features with a p-value greater than .05.

In [78]:

```
# Creating a DataFrame sorted by p-value
logit_coefs = pd.DataFrame({
    'coef': result.params.values,
    'odds_ratio': np.exp(result.params.values),
    'pvalue': result.pvalues,
}).sort_values(by='pvalue', ascending=False)
logit_coefs
```

Out[78]:

	coef	odds_ratio	pvalue
Location_Moree	0.000716	1.000717	1.000000e+00
Location_Cairns	0.002813	1.002817	1.000000e+00
Location_Newcastle	-0.006242	0.993778	1.000000e+00
Location_CoffsHarbour	0.009540	1.009586	1.000000e+00
Location_SydneyAirport	-0.009302	0.990741	1.000000e+00
Location_Melbourne	0.010581	1.010637	1.000000e+00
Location_Sydney	0.012201	1.012275	1.000000e+00
Location_Nhil	-0.008696	0.991342	1.000000e+00
Location_Walpole	0.013935	1.014033	9.999999e-01
Location_Watsonia	-0.018741	0.981434	9.999999e-01
Location_Richmond	0.020721	1.020937	9.999999e-01
Location_Nuriootpa	0.021413	1.021644	9.999999e-01
Location_Mildura	-0.020570	0.979640	9.999999e-01
Location_Cobar	0.022497	1.022752	9.999999e-01
Location_Uluru	0.015688	1.015812	9.999999e-01
Location_MountGambier	0.025686	1.026019	9.999999e-01

Location_Albany	0.027594	1.027978	9.999999e-01
Location_Portland	0.028388	1.028795	9.999999e-01
Location_BadgerysCreek	0.027731	1.028120	9.999999e-01
Location_Dartmoor	0.030967	1.031451	9.999999e-01
Location_AliceSprings	-0.027075	0.973288	9.999999e-01
Location_Bendigo	0.030537	1.031008	9.999999e-01
Location_NorfolkIsland	-0.037097	0.963583	9.999999e-01
Location_GoldCoast	-0.037865	0.962842	9.999999e-01
Location_Canberra	0.038572	1.039325	9.999999e-01
Location_WaggaWagga	0.036555	1.037231	9.999999e-01
Location_Sale	-0.039694	0.961084	9.999998e-01
Location_Launceston	-0.045555	0.955468	9.999998e-01
Location_Ballarat	-0.049788	0.951431	9.999998e-01
Location_Brisbane	0.052886	1.054310	9.999998e-01
Location_MelbourneAirport	-0.054044	0.947390	9.999998e-01
Location_Williamtown	0.054685	1.056208	9.999998e-01
Location_PerthAirport	0.062323	1.064306	9.999998e-01
Location_Penrith	0.063528	1.065590	9.999998e-01
Location_Tuggeranong	0.063109	1.065143	9.999998e-01
Location_Woomera	-0.059347	0.942379	9.999997e-01
Location_Perth	0.073000	1.075731	9.999997e-01
Location_Hobart	-0.076359	0.926484	9.999997e-01
Location_Witchcliffe	0.081251	1.084643	9.999997e-01
Location_PearceRAAF	0.073660	1.076441	9.999997e-01
Location_Albury	0.077849	1.080960	9.999997e-01
Location_NorahHead	-0.102282	0.902775	9.999996e-01
Location_SalmonGums	0.101261	1.106566	9.999996e-01
Location_Katherine	-0.079177	0.923876	9.999996e-01
Location_Darwin	-0.123016	0.884249	9.999996e-01
Location_Adelaide	0.118829	1.126177	9.999995e-01
Location_Townsville	-0.117288	0.889329	9.999995e-01
Location_Wollongong	-0.126041	0.881579	9.999995e-01

Location_MountGinini	-0.161782	0.850627	9.999994e-01
RainToday	-0.001087	0.998914	9.164778e-01
Temp3pm	0.009367	1.009411	7.671700e-01
Cloud9am	0.013517	1.013608	1.486511e-01
Evaporation	0.024756	1.025065	7.240790e-03
WindSpeed9am	-0.030127	0.970322	2.139489e-03
Temp9am	0.208261	1.231534	1.241464e-11
const	0.058351	1.060087	5.830523e-18
Rainfall	0.109693	1.115935	5.556709e-20
Humidity9am	0.149929	1.161752	7.012236e-31
MinTemp	0.298949	1.348441	4.463301e-41
MaxTemp	-0.431276	0.649680	2.767674e-41
WindSpeed3pm	-0.200462	0.818353	1.504097e-89
Sunshine	-0.264226	0.767800	1.260142e-125
Cloud3pm	0.248318	1.281868	4.220952e-154
Pressure9am	1.211288	3.357808	3.673235e-304
Humidity3pm	1.185913	3.273676	0.000000e+00
WindGustSpeed	0.779013	2.179321	0.000000e+00
Pressure3pm	-1.632489	0.195442	0.000000e+00
WindGustDir_E	-0.029032	0.971385	NaN
WindGustDir_ENE	-0.024401	0.975894	NaN
WindGustDir_ESE	0.001909	1.001911	NaN
WindGustDir_N	0.012423	1.012501	NaN
WindGustDir_NE	-0.041481	0.959368	NaN
WindGustDir_NNE	-0.040900	0.959925	NaN
WindGustDir_NNW	0.018005	1.018168	NaN
WindGustDir_NW	0.013477	1.013568	NaN
WindGustDir_S	0.027029	1.027398	NaN
WindGustDir_SE	0.006116	1.006134	NaN
WindGustDir_SSE	0.001102	1.001102	NaN
WindGustDir_SSW	0.003018	1.003023	NaN

WindGustDir_SW	0.003158	1.003163	NaN
WindGustDir_W	0.050218	1.051500	NaN
WindGustDir_WNW	-0.019907	0.980290	NaN
WindGustDir_WSW	-0.015840	0.984285	NaN
WindDir9am_E	-0.041939	0.958928	NaN
WindDir9am_ENE	0.033111	1.033665	NaN
WindDir9am_ESE	-0.055297	0.946204	NaN
WindDir9am_N	0.018638	1.018813	NaN
WindDir9am_NE	0.040621	1.041458	NaN
WindDir9am_NNE	0.082142	1.085610	NaN
WindDir9am_NNW	0.000839	1.000839	NaN
WindDir9am_NW	0.018815	1.018993	NaN
WindDir9am_S	-0.061961	0.939920	NaN
WindDir9am_SE	-0.050178	0.951060	NaN
WindDir9am_SSE	-0.074044	0.928631	NaN
WindDir9am_SSW	-0.009486	0.990559	NaN
WindDir9am_SW	0.018831	1.019009	NaN
WindDir9am_W	0.026841	1.027204	NaN
WindDir9am_WNW	0.016735	1.016876	NaN
WindDir9am_WSW	0.027126	1.027498	NaN
WindDir3pm_E	-0.022128	0.978115	NaN
WindDir3pm_ENE	-0.010489	0.989566	NaN
WindDir3pm_ESE	-0.034452	0.966135	NaN
WindDir3pm_N	0.058561	1.060310	NaN
WindDir3pm_NE	-0.063282	0.938679	NaN
WindDir3pm_NNE	0.018045	1.018209	NaN
WindDir3pm_NNW	0.097895	1.102847	NaN
WindDir3pm_NW	0.081969	1.085422	NaN
WindDir3pm_S	-0.028965	0.971451	NaN
WindDir3pm_SE	0.027733	1.028121	NaN
WindDir3pm_SSE	-0.023020	0.977243	NaN
WindDir3pm_SSW	-0.028610	0.971796	NaN

WindDir3pm_SW	-0.055699	0.945824	NaN
WindDir3pm_W	-0.031093	0.969385	NaN
WindDir3pm_WNW	0.035052	1.035673	NaN
WindDir3pm_WSW	-0.031927	0.968577	NaN

In [79]:

```
# Filter out variables that were not statistically significant
logit_coefs = logit_coefs[logit_coefs.pvalue < 0.05]
logit_coefs = logit_coefs.sort_values(by='odds_ratio',ascending=False).drop
logit_coefs
```

Out[79]:

	coef	odds_ratio	pvalue
Pressure9am	1.211288	3.357808	3.673235e-304
Humidity3pm	1.185913	3.273676	0.000000e+00
WindGustSpeed	0.779013	2.179321	0.000000e+00
MinTemp	0.298949	1.348441	4.463301e-41
Cloud3pm	0.248318	1.281868	4.220952e-154
Temp9am	0.208261	1.231534	1.241464e-11
Humidity9am	0.149929	1.161752	7.012236e-31
Rainfall	0.109693	1.115935	5.556709e-20
Evaporation	0.024756	1.025065	7.240790e-03
WindSpeed9am	-0.030127	0.970322	2.139489e-03
WindSpeed3pm	-0.200462	0.818353	1.504097e-89
Sunshine	-0.264226	0.767800	1.260142e-125
MaxTemp	-0.431276	0.649680	2.767674e-41
Pressure3pm	-1.632489	0.195442	0.000000e+00

In [80]:

```
# Sort by odds ratio
logit_coefs['odds_ratio'].sort_values(ascending=False)
```

Out[80]:
```
Pressure9am      3.357808
Humidity3pm      3.273676
WindGustSpeed    2.179321
MinTemp          1.348441
Cloud3pm         1.281868
Temp9am          1.231534
Humidity9am      1.161752
Rainfall         1.115935
Evaporation      1.025065
WindSpeed9am     0.970322
WindSpeed3pm     0.818353
Sunshine         0.767800
MaxTemp          0.649680
Pressure3pm      0.195442
Name: odds_ratio, dtype: float64
```

The logit model from Statsmodel appears to confirm our findings above, as well as provides us with confirmation that these features are statistically significant.

Now that we have reviewed the features, let's look at our overall best sklearn logistic regression model according to its recall metric.

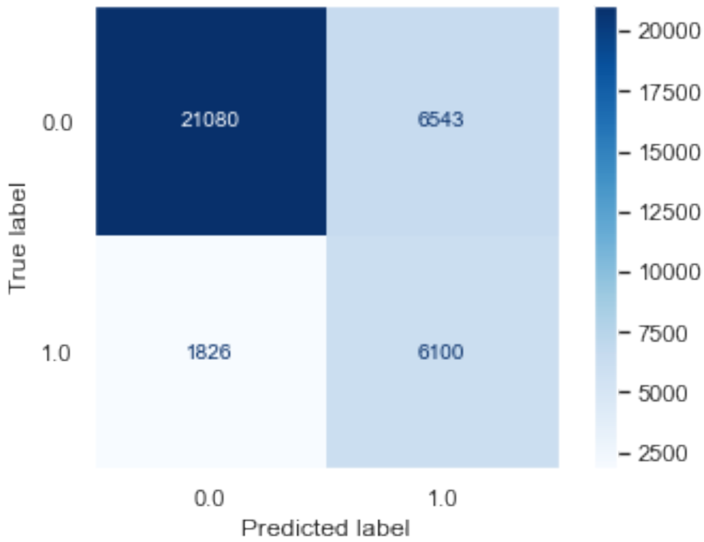
In [81]:
```
# Checking our our train and test data metrics

lr_best_test = lr_gridsearch.predict(X_test)
lr_best_train = lr_gridsearch.predict(X_train_resampled)

print("Training Data Results:\n")
print(classification_report(y_train_resampled, lr_best_train))
print("\nTest Data Results:\n")
print(classification_report(y_test, lr_best_test))
```

```
Training Data Results:

              precision    recall  f1-score   support

         0.0       0.78      0.80      0.79     82693
         1.0       0.79      0.78      0.79     82693

    accuracy                           0.79    165386
   macro avg       0.79      0.79      0.79    165386
weighted avg       0.79      0.79      0.79    165386


Test Data Results:

              precision    recall  f1-score   support

         0.0       0.93      0.78      0.85     27623
         1.0       0.50      0.78      0.61      7926

    accuracy                           0.78     35549
   macro avg       0.72      0.78      0.73     35549
weighted avg       0.83      0.78      0.79     35549
```

Based on our Training and Test Data results above, it does not appear that our model is overfitting for recall or f1-score. It does appear to be overfitting for precision, as the test data seems more sporadic 0 data and 1 data.

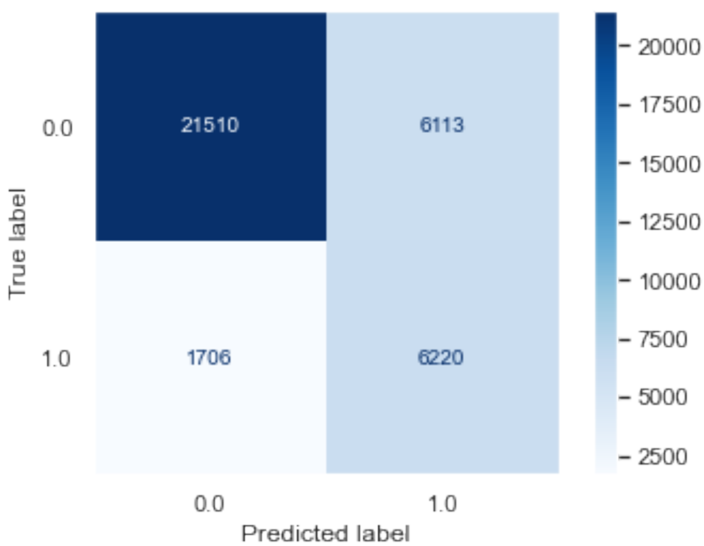
In [82]:

```
# Create confusion matrix for baseline model

plot_confusion_matrix(logreg, X_test, y_test,
                      cmap=plt.cm.Blues)
plt.grid(False)
plt.savefig('Visualizations/LogRegBase.png', bbox_inches = 'tight')
plt.show()
```



In [83]:

```
# Create confusion matrix for best model

plot_confusion_matrix(lr_gridsearch, X_test, y_test,
                      cmap=plt.cm.Blues)
plt.grid(False)
plt.savefig('Visualizations/LogRegBest.png', bbox_inches = 'tight')
plt.show()
```



k-Nearest Neighbors

In [84]:
```
# Create baseline model

knn_baseline_model = KNeighborsClassifier()
model_knn = knn_baseline_model.fit(X_train_resampled, y_train_resampled)
model_knn.score(X_test,y_test)
y_pred_knn = model_knn.predict(X_test)
```

In [85]:
```
recall_score(y_test,y_pred_knn)
```

Out[85]:
```
0.7453949028513752
```

Next, I will build a model using Pipeline and kNN to check score using a set of different parameters:

In [86]:
```
# Creating 3 different tests using different parameters

knn_pipeline = Pipeline([('ss', StandardScaler()),
                              ('knn', KNeighborsClassifier())])
```

In [87]:
```
# Created a function to iterate through the number of nearest neighbors fea
def find_best_k(X_train_resampled, y_train_resampled, X_test, y_test, min_k
    best_k = 0
    best_score = 0.0
    for k in range(min_k, max_k+1, 2):
        knn_pipeline_test = Pipeline([('ss', StandardScaler()),
                              ('knn', KNeighborsClassifier(n_neighbors=k))]
        knn_pipeline_test.fit(X_train_resampled, y_train_resampled)
        preds = knn_pipeline_test.predict(X_test)
        recall = recall_score(y_test, preds)
        if recall > best_score:
            best_k = k
            best_score = recall

    print("Best Value for k: {}".format(best_k))
    print("Recall Score: {}".format(best_score))
```

In [88]:
```
# Running the function to find the best value for k
find_best_k(X_train_resampled, y_train_resampled, X_test, y_test)
```

```
Best Value for k: 25
Recall Score: 0.653671461014383
```

In [89]:
```
# Create grid using GridSearchCV

knn_grid = [{'knn__n_neighbors':[25]}]
```

In [90]:
```
# Create grid using GridSearchCV

knn_gridsearch = GridSearchCV(estimator=knn_pipeline,
                              param_grid=knn_grid,
                              scoring='recall',
                              cv=3)
```

In [91]:
```
# Fit the training data
knn_gridsearch.fit(X_train_resampled, y_train_resampled)
print(knn_gridsearch.score(X_test, y_test))
```

```
0.653671461014383
```

In [92]:
```
print(knn_gridsearch.best_params_)
```

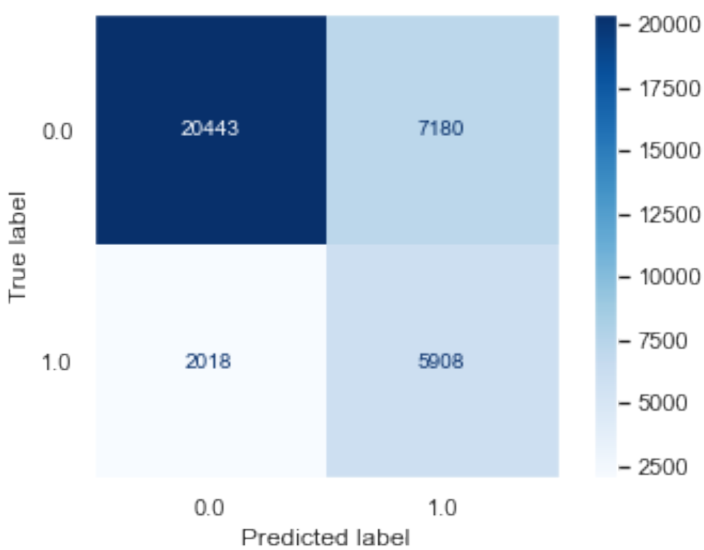
```
{'knn__n_neighbors': 25}
```

In [93]:
```
# Checking our our train and test data metrics

best_knn_test = knn_gridsearch.predict(X_test)
best_knn_train = knn_gridsearch.predict(X_train_resampled)

print("Training Data Results:\n")
print(classification_report(y_train_resampled, best_knn_train))
print("\nTest Data Results:\n")
print(classification_report(y_test, best_knn_test))
```

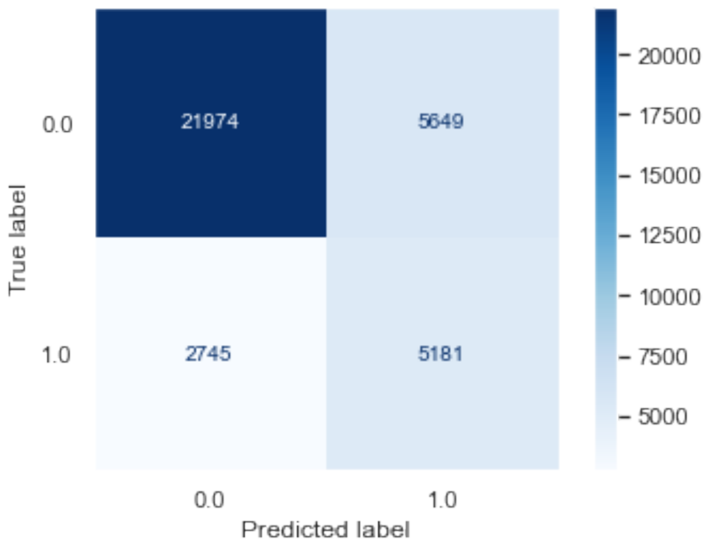
```
Training Data Results:

              precision    recall  f1-score   support

         0.0       0.84      0.82      0.83     82693
         1.0       0.82      0.84      0.83     82693

    accuracy                           0.83    165386
   macro avg       0.83      0.83      0.83    165386
weighted avg       0.83      0.83      0.83    165386


Test Data Results:

              precision    recall  f1-score   support

         0.0       0.89      0.80      0.84     27623
         1.0       0.48      0.65      0.55      7926

    accuracy                           0.76     35549
   macro avg       0.68      0.72      0.70     35549
weighted avg       0.80      0.76      0.78     35549
```

In [94]:

```python
# Create confusion matrix for baseline

plot_confusion_matrix(knn_baseline_model, X_test, y_test,
                     cmap=plt.cm.Blues,values_format = '.5g')
plt.grid(False)
plt.savefig('Visualizations/KNNbaseline.png', bbox_inches = 'tight')
plt.show()
```



In [95]:
```
# Create confusion matrix for best

plot_confusion_matrix(knn_gridsearch, X_test, y_test,
                      cmap=plt.cm.Blues)
plt.grid(False)
plt.savefig('Visualizations/KNNbest.png', bbox_inches = 'tight')
plt.show()
```



As we can see above, our best model seems to cap at a recall score of around .66, which was worse than the kNN baseline model. It also appears to be underfitting. Also, as kNN does not generate coefficients, Logistic Regression is still the best model type. I will now use a decision tree model to see if it generates a stronger model.

Decision Trees

In [96]:
```
# Create a Baseline Model

dt_base = DecisionTreeClassifier()
model_dt = dt_base.fit(X_train_resampled, y_train_resampled)
model_dt.score(X_test,y_test)
y_pred_dt = model_dt.predict(X_test)
```

In [97]:
```
recall_score(y_test,y_pred_dt)
```

Out[97]:
```
0.5643451930355791
```

In [98]:
```
# Create pipeline

dtree_pipeline = Pipeline([('ss', StandardScaler()),
                            ('dt', DecisionTreeClassifier())])
```

In [99]:
```
# Use GridSearchCV to create different models

dt_grid = [{'dt__criterion': ['gini','entropy'],
           'dt__max_leaf_nodes': [10,20,None],
           'dt__max_features': ['auto','sqrt','log2', None],
           'dt__random_state':[42]}]
```

In [100…
```
# Use GridSearchCV to create different models

dt_gridsearch = GridSearchCV(estimator=dtree_pipeline,
                             param_grid=dt_grid,
                             scoring='recall',
                             cv=3)
```

In [101…
```
# Fit model and print

dt_gridsearch.fit(X_train_resampled, y_train_resampled)
print(dt_gridsearch.score(X_test, y_test))
```

```
0.7641937925813778
```

In [102…
```
print(dt_gridsearch.best_params_)
```

```
{'dt__criterion': 'entropy', 'dt__max_features': 'auto', 'dt__max_leaf_nodes
': 10, 'dt__random_state': 42}
```

Now I will begin to look at the most important features in my decision tree model.

In [103…
```
# Using the best parameters, recreate the model and generate most important
dt_features_model = DecisionTreeClassifier(criterion='entropy',max_features
dt_features_model = dt_features_model.fit(X_train_resampled_scaled_df,y_tra
importance = dt_features_model.feature_importances_
importance
```

Out[103…

```
array([0.        , 0.07883961, 0.        , 0.05718609, 0.02757026,
       0.        , 0.        , 0.        , 0.06082803, 0.        ,
       0.15940519, 0.        , 0.54563903, 0.02967998, 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.01709123,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.        , 0.        , 0.        , 0.        , 0.        ,
       0.02376058, 0.        , 0.        , 0.        ])
```

In [104…

```
# Created a dict to keep the data together
importance_dict = {}
for coef, feat in zip(dt_features_model.feature_importances_,X_train_resamp
    importance_dict[feat] = coef
importance_dict
```

Out[104…

```
{'MinTemp': 0.0,
 'MaxTemp': 0.07883961419976189,
 'Rainfall': 0.0,
 'Evaporation': 0.05718608654046412,
 'Sunshine': 0.0275702604168548,
 'WindGustSpeed': 0.0,
 'WindSpeed9am': 0.0,
 'WindSpeed3pm': 0.0,
 'Humidity9am': 0.060828028282080567,
 'Humidity3pm': 0.0,
 'Pressure9am': 0.15940518885295288,
 'Pressure3pm': 0.0,
 'Cloud9am': 0.545639028965787,
 'Cloud3pm': 0.0296799751168835,
 'Temp9am': 0.0,
 'Temp3pm': 0.0,
 'RainToday': 0.0,
 'Location_Adelaide': 0.0,
 'Location_Albany': 0.0,
 'Location_Albury': 0.0,
 'Location_AliceSprings': 0.0,
 'Location_BadgerysCreek': 0.0,
```

```
 'Location_Ballarat': 0.0,
 'Location_Bendigo': 0.0,
 'Location_Brisbane': 0.0,
 'Location_Cairns': 0.0,
 'Location_Canberra': 0.0,
 'Location_Cobar': 0.0,
 'Location_CoffsHarbour': 0.0,
 'Location_Dartmoor': 0.0,
 'Location_Darwin': 0.0,
 'Location_GoldCoast': 0.0,
 'Location_Hobart': 0.0,
 'Location_Katherine': 0.0,
 'Location_Launceston': 0.0,
 'Location_Melbourne': 0.0,
 'Location_MelbourneAirport': 0.0,
 'Location_Mildura': 0.0,
 'Location_Moree': 0.0,
 'Location_MountGambier': 0.0,
 'Location_MountGinini': 0.0,
 'Location_Newcastle': 0.0,
 'Location_Nhil': 0.0,
 'Location_NorahHead': 0.0,
 'Location_NorfolkIsland': 0.0,
 'Location_Nuriootpa': 0.0,
 'Location_PearceRAAF': 0.0,
 'Location_Penrith': 0.0,
 'Location_Perth': 0.0,
 'Location_PerthAirport': 0.0,
 'Location_Portland': 0.0,
 'Location_Richmond': 0.0,
 'Location_Sale': 0.0,
 'Location_SalmonGums': 0.0,
 'Location_Sydney': 0.0,
 'Location_SydneyAirport': 0.0,
 'Location_Townsville': 0.0,
 'Location_Tuggeranong': 0.0,
 'Location_Uluru': 0.0,
 'Location_WaggaWagga': 0.0,
 'Location_Walpole': 0.0,
 'Location_Watsonia': 0.0,
 'Location_Williamtown': 0.0,
 'Location_Witchcliffe': 0.0,
 'Location_Wollongong': 0.0,
 'Location_Woomera': 0.0,
 'WindGustDir_E': 0.0,
 'WindGustDir_ENE': 0.0,
 'WindGustDir_ESE': 0.0,
 'WindGustDir_N': 0.017091234431553727,
 'WindGustDir_NE': 0.0,
 'WindGustDir_NNE': 0.0,
 'WindGustDir_NNW': 0.0,
 'WindGustDir_NW': 0.0,
 'WindGustDir_S': 0.0,
```

```
 'WindGustDir_SE': 0.0,
 'WindGustDir_SSE': 0.0,
 'WindGustDir_SSW': 0.0,
 'WindGustDir_SW': 0.0,
 'WindGustDir_W': 0.0,
 'WindGustDir_WNW': 0.0,
 'WindGustDir_WSW': 0.0,
 'WindDir9am_E': 0.0,
 'WindDir9am_ENE': 0.0,
 'WindDir9am_ESE': 0.0,
 'WindDir9am_N': 0.0,
 'WindDir9am_NE': 0.0,
 'WindDir9am_NNE': 0.0,
 'WindDir9am_NNW': 0.0,
 'WindDir9am_NW': 0.0,
 'WindDir9am_S': 0.0,
 'WindDir9am_SE': 0.0,
 'WindDir9am_SSE': 0.0,
 'WindDir9am_SSW': 0.0,
 'WindDir9am_SW': 0.0,
 'WindDir9am_W': 0.0,
 'WindDir9am_WNW': 0.0,
 'WindDir9am_WSW': 0.0,
 'WindDir3pm_E': 0.0,
 'WindDir3pm_ENE': 0.0,
 'WindDir3pm_ESE': 0.0,
 'WindDir3pm_N': 0.0,
 'WindDir3pm_NE': 0.0,
 'WindDir3pm_NNE': 0.0,
 'WindDir3pm_NNW': 0.0,
 'WindDir3pm_NW': 0.0,
 'WindDir3pm_S': 0.0,
 'WindDir3pm_SE': 0.0,
 'WindDir3pm_SSE': 0.0,
 'WindDir3pm_SSW': 0.0,
 'WindDir3pm_SW': 0.023760583193661658,
 'WindDir3pm_W': 0.0,
 'WindDir3pm_WNW': 0.0,
 'WindDir3pm_WSW': 0.0}
```

In [105…

```
# Sorted the dictionary by values
sorted_importance = sorted(importance_dict.items(), key=lambda x: x[1], reve

for i in enumerate(sorted_importance,start=1):
    print(i[0], i[1])
```

```
1 ('Cloud9am', 0.545639028965787)
2 ('Pressure9am', 0.15940518885295288)
3 ('MaxTemp', 0.07883961419976189)
4 ('Humidity9am', 0.06082028282080567)
5 ('Evaporation', 0.05718608654046412)
6 ('Cloud3pm', 0.0296799751168835)
```

```
7 ('Sunshine', 0.0275702604168548)
8 ('WindDir3pm_SW', 0.023760583193661658)
9 ('WindGustDir_N', 0.017091234431553727)
10 ('MinTemp', 0.0)
11 ('Rainfall', 0.0)
12 ('WindGustSpeed', 0.0)
13 ('WindSpeed9am', 0.0)
14 ('WindSpeed3pm', 0.0)
15 ('Humidity3pm', 0.0)
16 ('Pressure3pm', 0.0)
17 ('Temp9am', 0.0)
18 ('Temp3pm', 0.0)
19 ('RainToday', 0.0)
20 ('Location_Adelaide', 0.0)
21 ('Location_Albany', 0.0)
22 ('Location_Albury', 0.0)
23 ('Location_AliceSprings', 0.0)
24 ('Location_BadgerysCreek', 0.0)
25 ('Location_Ballarat', 0.0)
26 ('Location_Bendigo', 0.0)
27 ('Location_Brisbane', 0.0)
28 ('Location_Cairns', 0.0)
29 ('Location_Canberra', 0.0)
30 ('Location_Cobar', 0.0)
31 ('Location_CoffsHarbour', 0.0)
32 ('Location_Dartmoor', 0.0)
33 ('Location_Darwin', 0.0)
34 ('Location_GoldCoast', 0.0)
35 ('Location_Hobart', 0.0)
36 ('Location_Katherine', 0.0)
37 ('Location_Launceston', 0.0)
38 ('Location_Melbourne', 0.0)
39 ('Location_MelbourneAirport', 0.0)
40 ('Location_Mildura', 0.0)
41 ('Location_Moree', 0.0)
42 ('Location_MountGambier', 0.0)
43 ('Location_MountGinini', 0.0)
44 ('Location_Newcastle', 0.0)
45 ('Location_Nhil', 0.0)
46 ('Location_NorahHead', 0.0)
47 ('Location_NorfolkIsland', 0.0)
48 ('Location_Nuriootpa', 0.0)
49 ('Location_PearceRAAF', 0.0)
50 ('Location_Penrith', 0.0)
51 ('Location_Perth', 0.0)
52 ('Location_PerthAirport', 0.0)
53 ('Location_Portland', 0.0)
54 ('Location_Richmond', 0.0)
55 ('Location_Sale', 0.0)
56 ('Location_SalmonGums', 0.0)
57 ('Location_Sydney', 0.0)
58 ('Location_SydneyAirport', 0.0)
59 ('Location_Townsville', 0.0)
```

```
60 ('Location_Tuggeranong', 0.0)
61 ('Location_Uluru', 0.0)
62 ('Location_WaggaWagga', 0.0)
63 ('Location_Walpole', 0.0)
64 ('Location_Watsonia', 0.0)
65 ('Location_Williamtown', 0.0)
66 ('Location_Witchcliffe', 0.0)
67 ('Location_Wollongong', 0.0)
68 ('Location_Woomera', 0.0)
69 ('WindGustDir_E', 0.0)
70 ('WindGustDir_ENE', 0.0)
71 ('WindGustDir_ESE', 0.0)
72 ('WindGustDir_NE', 0.0)
73 ('WindGustDir_NNE', 0.0)
74 ('WindGustDir_NNW', 0.0)
75 ('WindGustDir_NW', 0.0)
76 ('WindGustDir_S', 0.0)
77 ('WindGustDir_SE', 0.0)
78 ('WindGustDir_SSE', 0.0)
79 ('WindGustDir_SSW', 0.0)
80 ('WindGustDir_SW', 0.0)
81 ('WindGustDir_W', 0.0)
82 ('WindGustDir_WNW', 0.0)
83 ('WindGustDir_WSW', 0.0)
84 ('WindDir9am_E', 0.0)
85 ('WindDir9am_ENE', 0.0)
86 ('WindDir9am_ESE', 0.0)
87 ('WindDir9am_N', 0.0)
88 ('WindDir9am_NE', 0.0)
89 ('WindDir9am_NNE', 0.0)
90 ('WindDir9am_NNW', 0.0)
91 ('WindDir9am_NW', 0.0)
92 ('WindDir9am_S', 0.0)
93 ('WindDir9am_SE', 0.0)
94 ('WindDir9am_SSE', 0.0)
95 ('WindDir9am_SSW', 0.0)
96 ('WindDir9am_SW', 0.0)
97 ('WindDir9am_W', 0.0)
98 ('WindDir9am_WNW', 0.0)
99 ('WindDir9am_WSW', 0.0)
100 ('WindDir3pm_E', 0.0)
101 ('WindDir3pm_ENE', 0.0)
102 ('WindDir3pm_ESE', 0.0)
103 ('WindDir3pm_N', 0.0)
104 ('WindDir3pm_NE', 0.0)
105 ('WindDir3pm_NNE', 0.0)
106 ('WindDir3pm_NNW', 0.0)
107 ('WindDir3pm_NW', 0.0)
108 ('WindDir3pm_S', 0.0)
109 ('WindDir3pm_SE', 0.0)
110 ('WindDir3pm_SSE', 0.0)
111 ('WindDir3pm_SSW', 0.0)
112 ('WindDir3pm_W', 0.0)
```

```
113 ('WindDir3pm_WNW', 0.0)
114 ('WindDir3pm_WSW', 0.0)
```

In [106…

```
# Calculated odds ratio by taking e and raising it by the importance factor
odds_ratio2 = {}
for coef, feat in zip(dt_features_model.feature_importances_,X_train_resamp
    odds_ratio2[feat] = math.pow(math.e,coef)
odds_ratio2
```

Out[106…

```
{'MinTemp': 1.0,
 'MaxTemp': 1.0820307657824524,
 'Rainfall': 1.0,
 'Evaporation': 1.0588528303236087,
 'Sunshine': 1.027953837035149,
 'WindGustSpeed': 1.0,
 'WindSpeed9am': 1.0,
 'WindSpeed3pm': 1.0,
 'Humidity9am': 1.0627161413512776,
 'Humidity3pm': 1.0,
 'Pressure9am': 1.172813061197659,
 'Pressure3pm': 1.0,
 'Cloud9am': 1.7257108092912632,
 'Cloud3pm': 1.0301248156235805,
 'Temp9am': 1.0,
 'Temp3pm': 1.0,
 'RainToday': 1.0,
 'Location_Adelaide': 1.0,
 'Location_Albany': 1.0,
 'Location_Albury': 1.0,
 'Location_AliceSprings': 1.0,
 'Location_BadgerysCreek': 1.0,
 'Location_Ballarat': 1.0,
 'Location_Bendigo': 1.0,
 'Location_Brisbane': 1.0,
 'Location_Cairns': 1.0,
 'Location_Canberra': 1.0,
 'Location_Cobar': 1.0,
 'Location_CoffsHarbour': 1.0,
 'Location_Dartmoor': 1.0,
 'Location_Darwin': 1.0,
 'Location_GoldCoast': 1.0,
 'Location_Hobart': 1.0,
 'Location_Katherine': 1.0,
 'Location_Launceston': 1.0,
 'Location_Melbourne': 1.0,
 'Location_MelbourneAirport': 1.0,
 'Location_Mildura': 1.0,
 'Location_Moree': 1.0,
 'Location_MountGambier': 1.0,
 'Location_MountGinini': 1.0,
 'Location_Newcastle': 1.0,
 'Location_Nhil': 1.0,
```

```
 'Location_NorahHead': 1.0,
 'Location_NorfolkIsland': 1.0,
 'Location_Nuriootpa': 1.0,
 'Location_PearceRAAF': 1.0,
 'Location_Penrith': 1.0,
 'Location_Perth': 1.0,
 'Location_PerthAirport': 1.0,
 'Location_Portland': 1.0,
 'Location_Richmond': 1.0,
 'Location_Sale': 1.0,
 'Location_SalmonGums': 1.0,
 'Location_Sydney': 1.0,
 'Location_SydneyAirport': 1.0,
 'Location_Townsville': 1.0,
 'Location_Tuggeranong': 1.0,
 'Location_Uluru': 1.0,
 'Location_WaggaWagga': 1.0,
 'Location_Walpole': 1.0,
 'Location_Watsonia': 1.0,
 'Location_Williamtown': 1.0,
 'Location_Witchcliffe': 1.0,
 'Location_Wollongong': 1.0,
 'Location_Woomera': 1.0,
 'WindGustDir_E': 1.0,
 'WindGustDir_ENE': 1.0,
 'WindGustDir_ESE': 1.0,
 'WindGustDir_N': 1.0172381252338765,
 'WindGustDir_NE': 1.0,
 'WindGustDir_NNE': 1.0,
 'WindGustDir_NNW': 1.0,
 'WindGustDir_NW': 1.0,
 'WindGustDir_S': 1.0,
 'WindGustDir_SE': 1.0,
 'WindGustDir_SSE': 1.0,
 'WindGustDir_SSW': 1.0,
 'WindGustDir_SW': 1.0,
 'WindGustDir_W': 1.0,
 'WindGustDir_WNW': 1.0,
 'WindGustDir_WSW': 1.0,
 'WindDir9am_E': 1.0,
 'WindDir9am_ENE': 1.0,
 'WindDir9am_ESE': 1.0,
 'WindDir9am_N': 1.0,
 'WindDir9am_NE': 1.0,
 'WindDir9am_NNE': 1.0,
 'WindDir9am_NNW': 1.0,
 'WindDir9am_NW': 1.0,
 'WindDir9am_S': 1.0,
 'WindDir9am_SE': 1.0,
 'WindDir9am_SSE': 1.0,
 'WindDir9am_SSW': 1.0,
 'WindDir9am_SW': 1.0,
 'WindDir9am_W': 1.0,
```

```
 'WindDir9am_WNW': 1.0,
 'WindDir9am_WSW': 1.0,
 'WindDir3pm_E': 1.0,
 'WindDir3pm_ENE': 1.0,
 'WindDir3pm_ESE': 1.0,
 'WindDir3pm_N': 1.0,
 'WindDir3pm_NE': 1.0,
 'WindDir3pm_NNE': 1.0,
 'WindDir3pm_NNW': 1.0,
 'WindDir3pm_NW': 1.0,
 'WindDir3pm_S': 1.0,
 'WindDir3pm_SE': 1.0,
 'WindDir3pm_SSE': 1.0,
 'WindDir3pm_SSW': 1.0,
 'WindDir3pm_SW': 1.0240451149279752,
 'WindDir3pm_W': 1.0,
 'WindDir3pm_WNW': 1.0,
 'WindDir3pm_WSW': 1.0}
```

In [107…

```
# Sort the list and format it as a numbered list
sorted_odds2 = sorted(odds_ratio2.items(), key=lambda x: x[1], reverse=True

for i in enumerate(sorted_odds2,start=1):
    print(i[0], i[1])
```

```
1 ('Cloud9am', 1.7257108092912632)
2 ('Pressure9am', 1.172813061197659)
3 ('MaxTemp', 1.0820307657824524)
4 ('Humidity9am', 1.0627161413512776)
5 ('Evaporation', 1.0588528303236087)
6 ('Cloud3pm', 1.0301248156235805)
7 ('Sunshine', 1.027953837035149)
8 ('WindDir3pm_SW', 1.0240451149279752)
9 ('WindGustDir_N', 1.0172381252338765)
10 ('MinTemp', 1.0)
11 ('Rainfall', 1.0)
12 ('WindGustSpeed', 1.0)
13 ('WindSpeed9am', 1.0)
14 ('WindSpeed3pm', 1.0)
15 ('Humidity3pm', 1.0)
16 ('Pressure3pm', 1.0)
17 ('Temp9am', 1.0)
18 ('Temp3pm', 1.0)
19 ('RainToday', 1.0)
20 ('Location_Adelaide', 1.0)
21 ('Location_Albany', 1.0)
22 ('Location_Albury', 1.0)
23 ('Location_AliceSprings', 1.0)
24 ('Location_BadgerysCreek', 1.0)
25 ('Location_Ballarat', 1.0)
26 ('Location_Bendigo', 1.0)
27 ('Location_Brisbane', 1.0)
```

```
28 ('Location_Cairns', 1.0)
29 ('Location_Canberra', 1.0)
30 ('Location_Cobar', 1.0)
31 ('Location_CoffsHarbour', 1.0)
32 ('Location_Dartmoor', 1.0)
33 ('Location_Darwin', 1.0)
34 ('Location_GoldCoast', 1.0)
35 ('Location_Hobart', 1.0)
36 ('Location_Katherine', 1.0)
37 ('Location_Launceston', 1.0)
38 ('Location_Melbourne', 1.0)
39 ('Location_MelbourneAirport', 1.0)
40 ('Location_Mildura', 1.0)
41 ('Location_Moree', 1.0)
42 ('Location_MountGambier', 1.0)
43 ('Location_MountGinini', 1.0)
44 ('Location_Newcastle', 1.0)
45 ('Location_Nhil', 1.0)
46 ('Location_NorahHead', 1.0)
47 ('Location_NorfolkIsland', 1.0)
48 ('Location_Nuriootpa', 1.0)
49 ('Location_PearceRAAF', 1.0)
50 ('Location_Penrith', 1.0)
51 ('Location_Perth', 1.0)
52 ('Location_PerthAirport', 1.0)
53 ('Location_Portland', 1.0)
54 ('Location_Richmond', 1.0)
55 ('Location_Sale', 1.0)
56 ('Location_SalmonGums', 1.0)
57 ('Location_Sydney', 1.0)
58 ('Location_SydneyAirport', 1.0)
59 ('Location_Townsville', 1.0)
60 ('Location_Tuggeranong', 1.0)
61 ('Location_Uluru', 1.0)
62 ('Location_WaggaWagga', 1.0)
63 ('Location_Walpole', 1.0)
64 ('Location_Watsonia', 1.0)
65 ('Location_Williamtown', 1.0)
66 ('Location_Witchcliffe', 1.0)
67 ('Location_Wollongong', 1.0)
68 ('Location_Woomera', 1.0)
69 ('WindGustDir_E', 1.0)
70 ('WindGustDir_ENE', 1.0)
71 ('WindGustDir_ESE', 1.0)
72 ('WindGustDir_NE', 1.0)
73 ('WindGustDir_NNE', 1.0)
74 ('WindGustDir_NNW', 1.0)
75 ('WindGustDir_NW', 1.0)
76 ('WindGustDir_S', 1.0)
77 ('WindGustDir_SE', 1.0)
78 ('WindGustDir_SSE', 1.0)
79 ('WindGustDir_SSW', 1.0)
80 ('WindGustDir_SW', 1.0)
```

```
81 ('WindGustDir_W', 1.0)
82 ('WindGustDir_WNW', 1.0)
83 ('WindGustDir_WSW', 1.0)
84 ('WindDir9am_E', 1.0)
85 ('WindDir9am_ENE', 1.0)
86 ('WindDir9am_ESE', 1.0)
87 ('WindDir9am_N', 1.0)
88 ('WindDir9am_NE', 1.0)
89 ('WindDir9am_NNE', 1.0)
90 ('WindDir9am_NNW', 1.0)
91 ('WindDir9am_NW', 1.0)
92 ('WindDir9am_S', 1.0)
93 ('WindDir9am_SE', 1.0)
94 ('WindDir9am_SSE', 1.0)
95 ('WindDir9am_SSW', 1.0)
96 ('WindDir9am_SW', 1.0)
97 ('WindDir9am_W', 1.0)
98 ('WindDir9am_WNW', 1.0)
99 ('WindDir9am_WSW', 1.0)
100 ('WindDir3pm_E', 1.0)
101 ('WindDir3pm_ENE', 1.0)
102 ('WindDir3pm_ESE', 1.0)
103 ('WindDir3pm_N', 1.0)
104 ('WindDir3pm_NE', 1.0)
105 ('WindDir3pm_NNE', 1.0)
106 ('WindDir3pm_NNW', 1.0)
107 ('WindDir3pm_NW', 1.0)
108 ('WindDir3pm_S', 1.0)
109 ('WindDir3pm_SE', 1.0)
110 ('WindDir3pm_SSE', 1.0)
111 ('WindDir3pm_SSW', 1.0)
112 ('WindDir3pm_W', 1.0)
113 ('WindDir3pm_WNW', 1.0)
114 ('WindDir3pm_WSW', 1.0)
```

In [108…

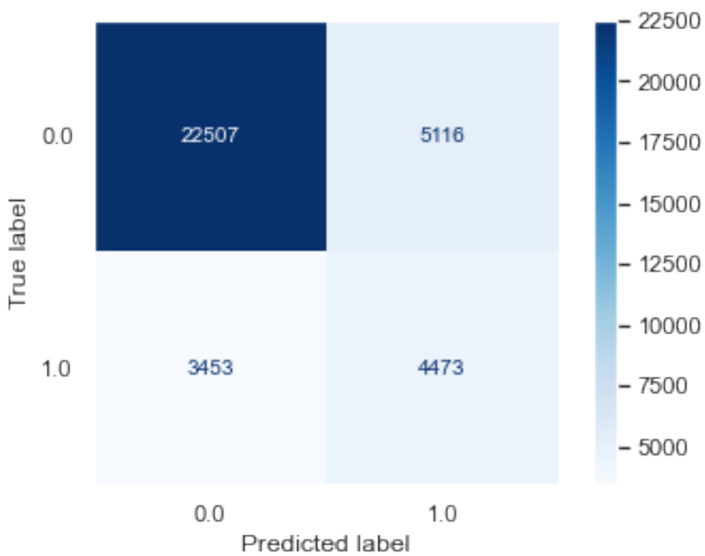
```
# Checking our our train and test data metrics

best_dt_test = dt_gridsearch.predict(X_test)
best_dt_train = dt_gridsearch.predict(X_train_resampled)

print("Training Data Results:\n")
print(classification_report(y_train_resampled, best_dt_train))
print("\nTest Data Results:\n")
print(classification_report(y_test, best_dt_test))
```

```
Training Data Results:

              precision    recall  f1-score   support

         0.0       0.74      0.60      0.66     82693
         1.0       0.66      0.79      0.72     82693

    accuracy                           0.69    165386
   macro avg       0.70      0.69      0.69    165386
weighted avg       0.70      0.69      0.69    165386


Test Data Results:

              precision    recall  f1-score   support

         0.0       0.89      0.57      0.69     27623
         1.0       0.34      0.76      0.47      7926

    accuracy                           0.61     35549
   macro avg       0.61      0.66      0.58     35549
weighted avg       0.77      0.61      0.64     35549
```

This decision tree model prevents Type-II error better than the kNN model. However, it is underfitting severely, and also has the worst weighted-average recall score.

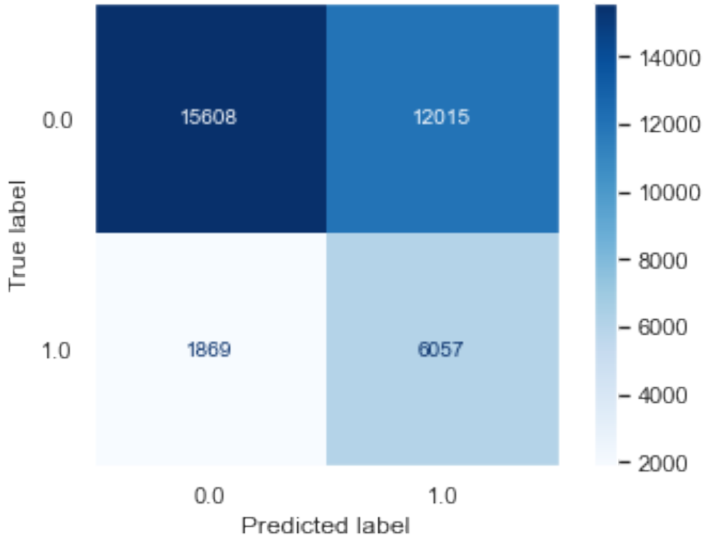
In [109…

```
# Create confusion matrix for baseline

cm = plot_confusion_matrix(dt_base, X_test, y_test,
                           cmap=plt.cm.Blues)
plt.grid(False)
plt.savefig('Visualizations/DT.png', bbox_inches = 'tight')
plt.show()
```



In [110…

```
# Create confusion matrix for best

plot_confusion_matrix(dt_gridsearch, X_test, y_test,
                      cmap=plt.cm.Blues, values_format = '.5g')
plt.grid(False)
plt.savefig('Visualizations/DT.png', bbox_inches = 'tight')
plt.show()
```



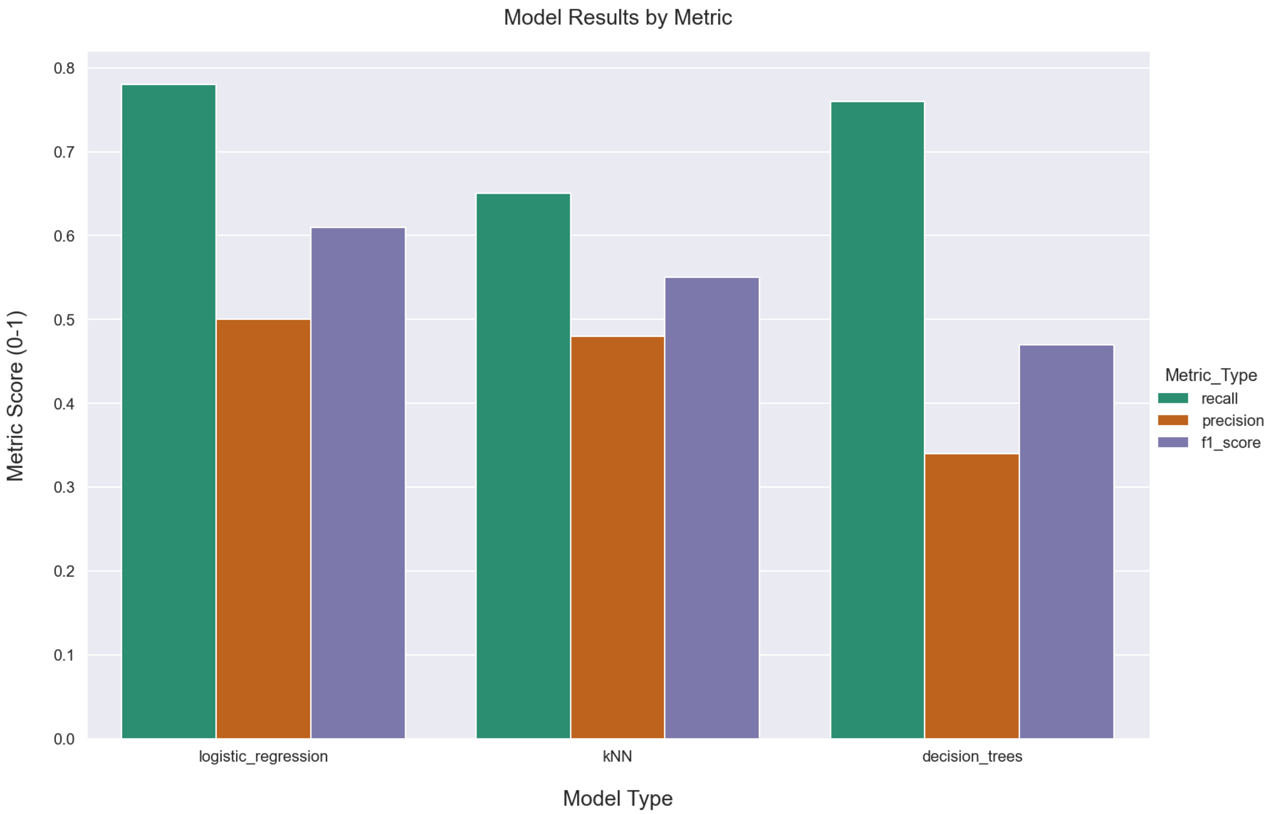
Section 5: Results

Best Model

Let's first compare our three models by metric type, and select the best one based on the information gathered earlier.

In [241…

```
# Put the results from the classification models into a dictionary
results_dictionary = {'recall' : {'logistic_regression':0.78,'kNN':0.65,'de
                      'precision' : {'logistic_regression':0.50,'kNN':0.48,
                      'f1_score': {'logistic_regression': 0.61, 'kNN': 0.55

# Convert the dictionary to a DataFrame, and melt the data into graphable m
model_results_all = pd.DataFrame(results_dictionary)
model_results_all.reset_index(level=0, inplace=True)
model_results_all_sns = pd.melt(model_results_all, id_vars='index', var_nam

# Graph and save the data to an image
results_image = sns.factorplot(x='index',y='Metric_Score',data=model_result
results_image = plt.title("Model Results by Metric\n",size=30)
results_image = plt.xlabel("\nModel Type",size=30)
results_image = plt.ylabel("Metric Score (0-1)\n",size=30)

plt.savefig('Visualizations/ModelResults.png', bbox_inches = 'tight')
```



The logistic regression model appears to be our best model, at a recall score of around .78. Out of the three, it also appears to have the least amount of overfitting/underfitting. Now, I will take a look at the remaining metrics.

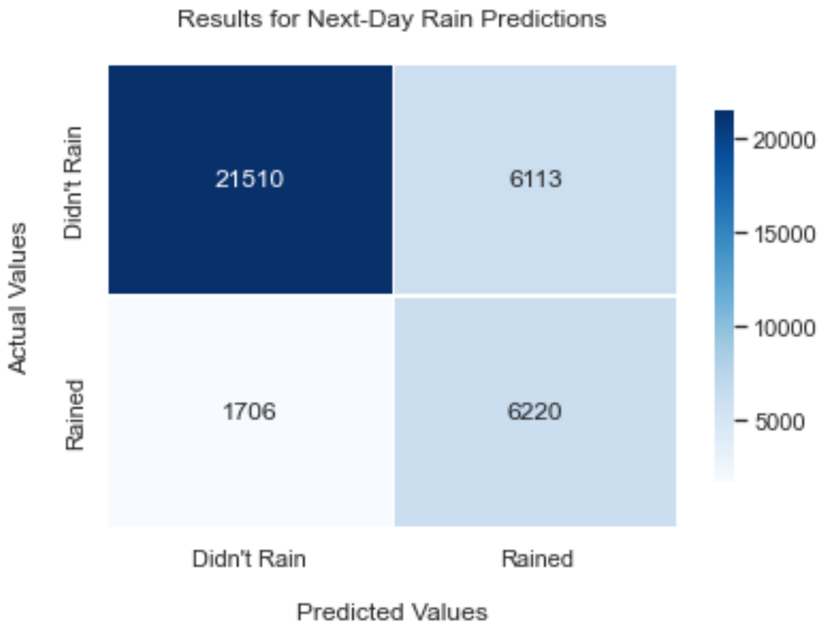
In [111…

```
# Checking our our train and test data metrics

final_test = lr_gridsearch.predict(X_test)
final_train = lr_gridsearch.predict(X_train_resampled)

print("Training Data Results:\n")
print(classification_report(y_train_resampled, final_train))
print("\nTest Data Results:\n")
print(classification_report(y_test, final_test))
```

```
Training Data Results:

              precision    recall  f1-score   support

         0.0       0.78      0.80      0.79     82693
         1.0       0.79      0.78      0.79     82693

    accuracy                           0.79    165386
   macro avg       0.79      0.79      0.79    165386
weighted avg       0.79      0.79      0.79    165386


Test Data Results:

              precision    recall  f1-score   support

         0.0       0.93      0.78      0.85     27623
         1.0       0.50      0.78      0.61      7926

    accuracy                           0.78     35549
   macro avg       0.72      0.78      0.73     35549
weighted avg       0.83      0.78      0.79     35549
```

In [112…

```
# Create confusion matrix for final model
lrcm = confusion_matrix(y_test, final_test)
sns.set_context("talk")
sns.set_theme(style='darkgrid')
ax = sns.heatmap(lrcm, annot=True, cmap='Blues',fmt = 'g',linewidth=0.3, cb

ax.set_title('Results for Next-Day Rain Predictions\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values\n');

## Ticket labels - List must be in alphabetical order
ax.xaxis.set_ticklabels(["Didn't Rain","Rained"])
ax.yaxis.set_ticklabels(["Didn't Rain","Rained"])
plt.savefig('Visualizations/FinalCM.png', bbox_inches = 'tight')
## Display the visualization of the Confusion Matrix.
plt.show()
```



Accuracy: This model has 78% accuracy, meaning that it correctly determines that it will rain the next day 78% of the time. As True Negatives account for 77% of our test data, there is a lot of bias within this metric. Therefore, it is best to ignore it in our analysis.

Precision: If our model says that it will rain tomorrow, there between a 51% chance that it is a true positive and will actually rain the next day. If our model says it wont rain tomorrow, there is a 93% chance that it won't rain tomorrow. The weighted average precision of our model is 83%. Our model is way better at predicting when it won't rain than when it will. In the future, we should take steps to try to raise precision.

Recall: Our recall score is the most important aspect of this model. For instances when it actually rained the next day, our model correctly classified that it would rain 78% of the time. For instances when it did not rain the next day, our model correctly predicted that it would not rain 78% of the time. The weighted average recall of our model is 78%.

Feature Importance

I was able to create 3 different instances of choosing the most important features. One using logistic regression coefficients, one using logit from statsmodels, and one using features importance from decision trees. I will now create a function to calculate the weighted average of the 3 models and interpret the results.

In [113…

```
# Create a function to calculate the weighted average using our odds ratio
# regression models and decision tree model
def meanlogreg(coef):
    result = round(((odds_ratio[coef] + logit_coefs['odds_ratio'][coef] + o
    return result
```

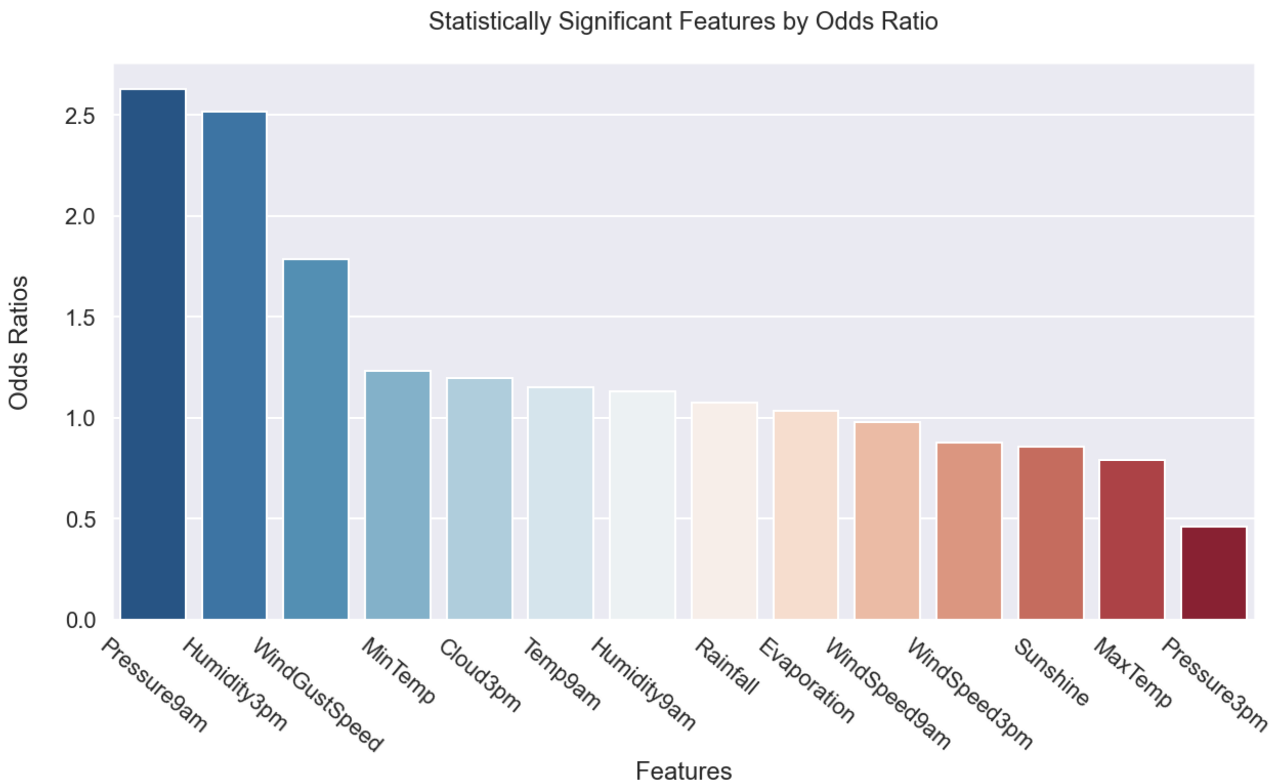
In [114…

```
# Iterate the function for every statistically significant coefficient from
best_feature_values = []
best_features = []
best_features_dict = {}

for coef in logit_coefs.index:
    print(coef, ": Weighted-Average Odds Ratio -",meanlogreg(coef),'\n')
    best_feature_values.append(meanlogreg(coef))

for coef in logit_coefs.index:
    best_features.append(coef)

# Zip the lists to a dictionary
best_features_dict = dict(zip(best_features,best_feature_values))
```

```
Pressure9am : Weighted-Average Odds Ratio - 2.626

Humidity3pm : Weighted-Average Odds Ratio - 2.516

WindGustSpeed : Weighted-Average Odds Ratio - 1.786

MinTemp : Weighted-Average Odds Ratio - 1.232

Cloud3pm : Weighted-Average Odds Ratio - 1.198

Temp9am : Weighted-Average Odds Ratio - 1.154

Humidity9am : Weighted-Average Odds Ratio - 1.129

Rainfall : Weighted-Average Odds Ratio - 1.077

Evaporation : Weighted-Average Odds Ratio - 1.036

WindSpeed9am : Weighted-Average Odds Ratio - 0.98

WindSpeed3pm : Weighted-Average Odds Ratio - 0.879

Sunshine : Weighted-Average Odds Ratio - 0.855

MaxTemp : Weighted-Average Odds Ratio - 0.794

Pressure3pm : Weighted-Average Odds Ratio - 0.464
```

In [211…

```
# Create bar graph depicting the odds ratios
sns.set(rc = {'figure.figsize':(20,10)})
sns.set_context('poster')
fig = sns.barplot(x=best_features, y=best_feature_values,palette='RdBu_r').
fig = plt.xlabel("Features")
fig = plt.ylabel("Odds Ratios\n")
fig = plt.xticks(rotation = 320)
plt.savefig('Visualizations/Features.png', bbox_inches = 'tight')
```



As we can see above, Pressure at 9am and Humidity are the most important **positive** features according to our models, at an odds ratio of 2.63 and 2.52, respectively. Next best is wind gust speed at an odds ratio of roughly 1.78. This means that:

- An increase of 1 hectopascal of pressure at 9am is associated with a 163% *increase in the odds* that it will rain the next day.
- An increase of 1 percentage point in humidity at 3pm is associated with a 152% *increase in the odds* that it will rain the next day.
- An increase of 1 kilometer per hour for the day's strongest wind speed is associated with a 78% *increase in the odds* that it will rain the next day.

After these three variables, our remaining variables that were statistically significant vary from odds ratios from 1.2 down to .5.

Note that the most important **negative** features are Sunshine, Max Temp, and Pressure 3pm at odds ratios, of .86, .79, and .46, respectively, meaning:

- An increase of 1 hectopascal of pressure at 3pm is associated with a 54% *decrease in the odds* that it will rain the next day.
- An increase of 1 degree (C) of the max temperature during a day is associated with a 21% *decrease in the odds* that it will rain the next day.
- An increase of 1 hour of sunshine during a day is associated with a 14% *decrease in the odds* that it will rain the next day.