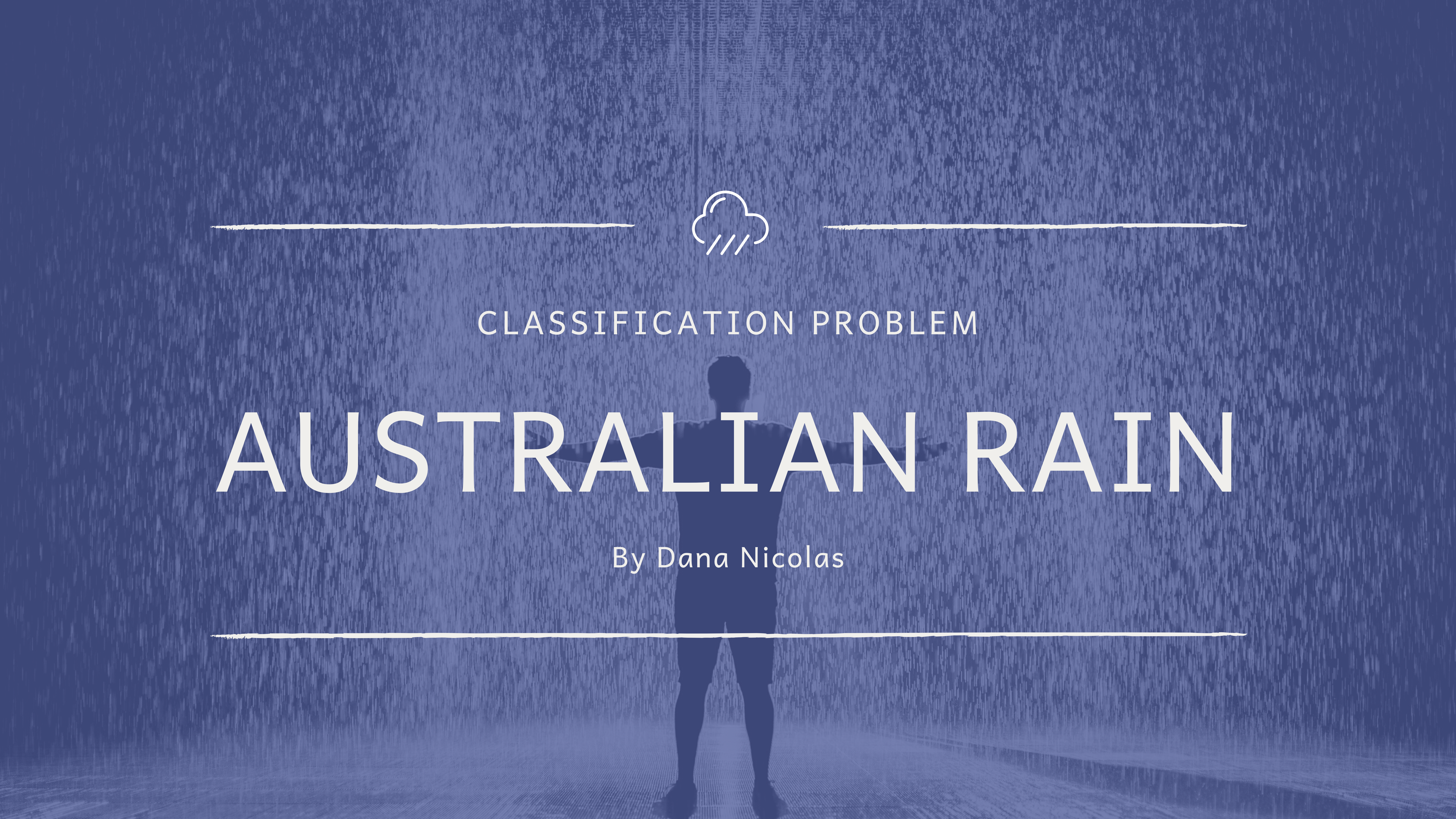
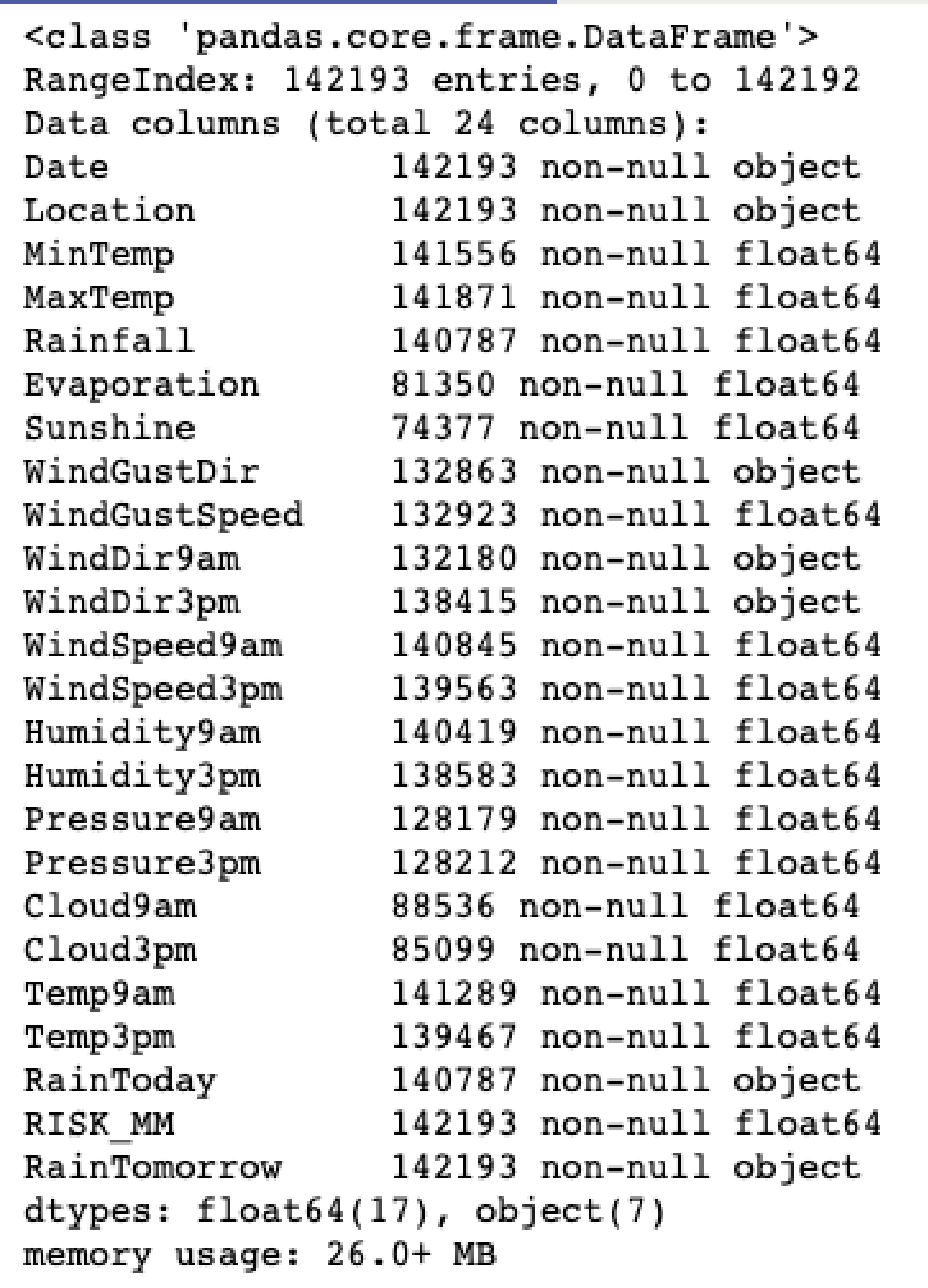


CLASSIFICATION PROBLEM

AUSTRALIAN RAIN

By Dana Nicolas

Data Set



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142193 entries, 0 to 142192
Data columns (total 24 columns):
Date                 142193 non-null object
Location             142193 non-null object
MinTemp              141556 non-null float64
MaxTemp              141871 non-null float64
Rainfall             140787 non-null float64
Evaporation          81350 non-null float64
Sunshine             74377 non-null float64
WindGustDir          132863 non-null object
WindGustSpeed        132923 non-null float64
WindDir9am           132180 non-null object
WindDir3pm           138415 non-null object
WindSpeed9am         140845 non-null float64
WindSpeed3pm         139563 non-null float64
Humidity9am          140419 non-null float64
Humidity3pm          138583 non-null float64
Pressure9am          128179 non-null float64
Pressure3pm          128212 non-null float64
Cloud9am             88536 non-null float64
Cloud3pm             85099 non-null float64
Temp9am              141289 non-null float64
Temp3pm              139467 non-null float64
RainToday            140787 non-null object
RISK_MM              142193 non-null float64
RainTomorrow         142193 non-null object
dtypes: float64(17), object(7)
memory usage: 26.0+ MB
```

5 Categorical

14 Numerical

2 Boolean

Dropped columns

Sunshine (47.7% missing values)

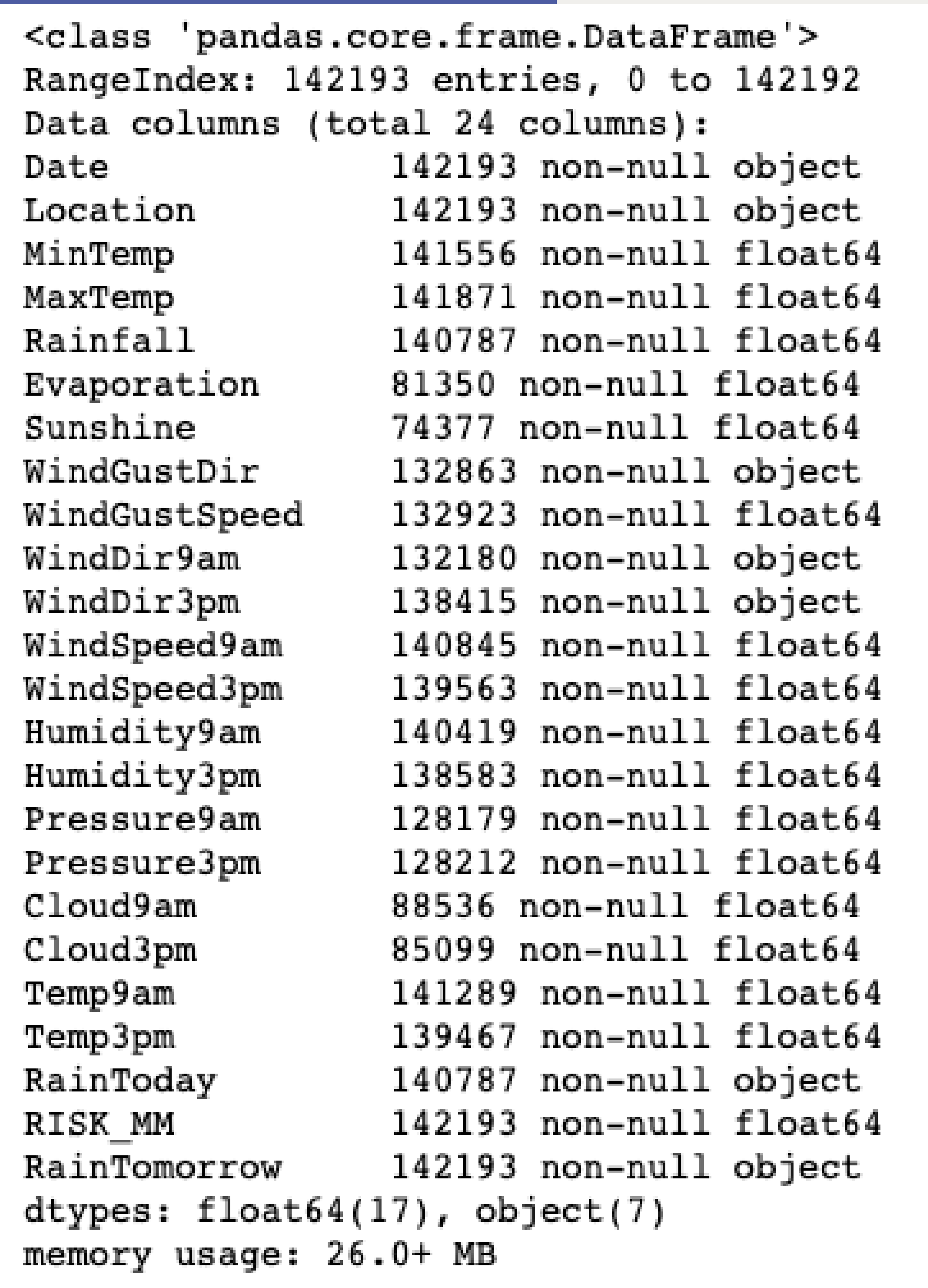
Evaporation (42.8% missing values)

Notable

Rainfall (63.5% missing values)

Date string was converted to timestamp

Data Set



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142193 entries, 0 to 142192
Data columns (total 24 columns):
Date                 142193 non-null object
Location             142193 non-null object
MinTemp              141556 non-null float64
MaxTemp              141871 non-null float64
Rainfall             140787 non-null float64
Evaporation          81350 non-null float64
Sunshine             74377 non-null float64
WindGustDir          132863 non-null object
WindGustSpeed        132923 non-null float64
WindDir9am           132180 non-null object
WindDir3pm           138415 non-null object
WindSpeed9am         140845 non-null float64
WindSpeed3pm         139563 non-null float64
Humidity9am          140419 non-null float64
Humidity3pm          138583 non-null float64
Pressure9am          128179 non-null float64
Pressure3pm          128212 non-null float64
Cloud9am             88536 non-null float64
Cloud3pm             85099 non-null float64
Temp9am              141289 non-null float64
Temp3pm              139467 non-null float64
RainToday            140787 non-null object
RISK_MM              142193 non-null float64
RainTomorrow         142193 non-null object
dtypes: float64(17), object(7)
memory usage: 26.0+ MB
```

Used np.mean() to fill na values

MinTemp

MaxTemp

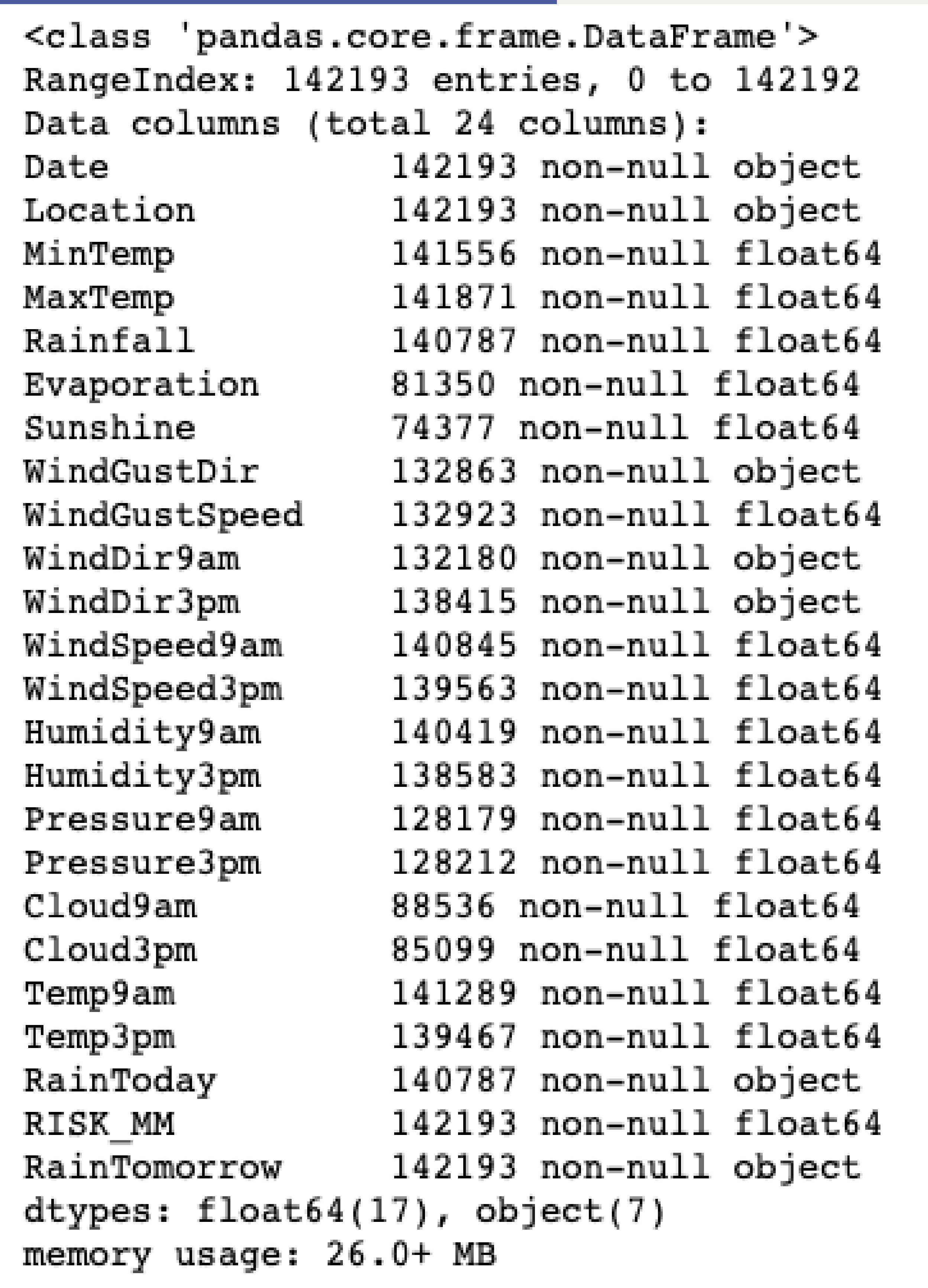
Temp9am

Temp3pm

WindGustSpeed

WindSpeed9am

WindSpeed3pm



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142193 entries, 0 to 142192
Data columns (total 24 columns):
Date                 142193 non-null object
Location             142193 non-null object
MinTemp              141556 non-null float64
MaxTemp              141871 non-null float64
Rainfall             140787 non-null float64
Evaporation          81350 non-null float64
Sunshine             74377 non-null float64
WindGustDir          132863 non-null object
WindGustSpeed        132923 non-null float64
WindDir9am           132180 non-null object
WindDir3pm           138415 non-null object
WindSpeed9am         140845 non-null float64
WindSpeed3pm         139563 non-null float64
Humidity9am          140419 non-null float64
Humidity3pm          138583 non-null float64
Pressure9am          128179 non-null float64
Pressure3pm          128212 non-null float64
Cloud9am             88536 non-null float64
Cloud3pm             85099 non-null float64
Temp9am              141289 non-null float64
Temp3pm              139467 non-null float64
RainToday            140787 non-null object
RISK_MM              142193 non-null float64
RainTomorrow         142193 non-null object
dtypes: float64(17), object(7)
memory usage: 26.0+ MB
```

Data Set

Used mode to fill na values

WindGustDir

WindDir9am

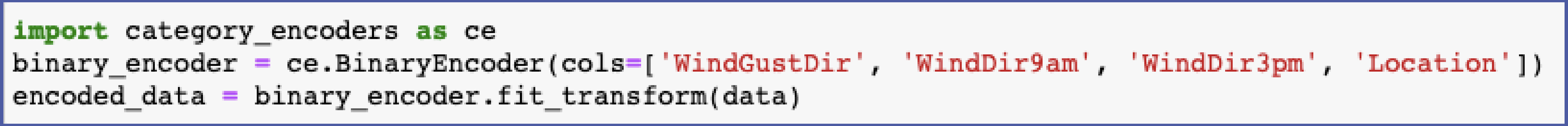
WindDir3pm

Changed to boolean values

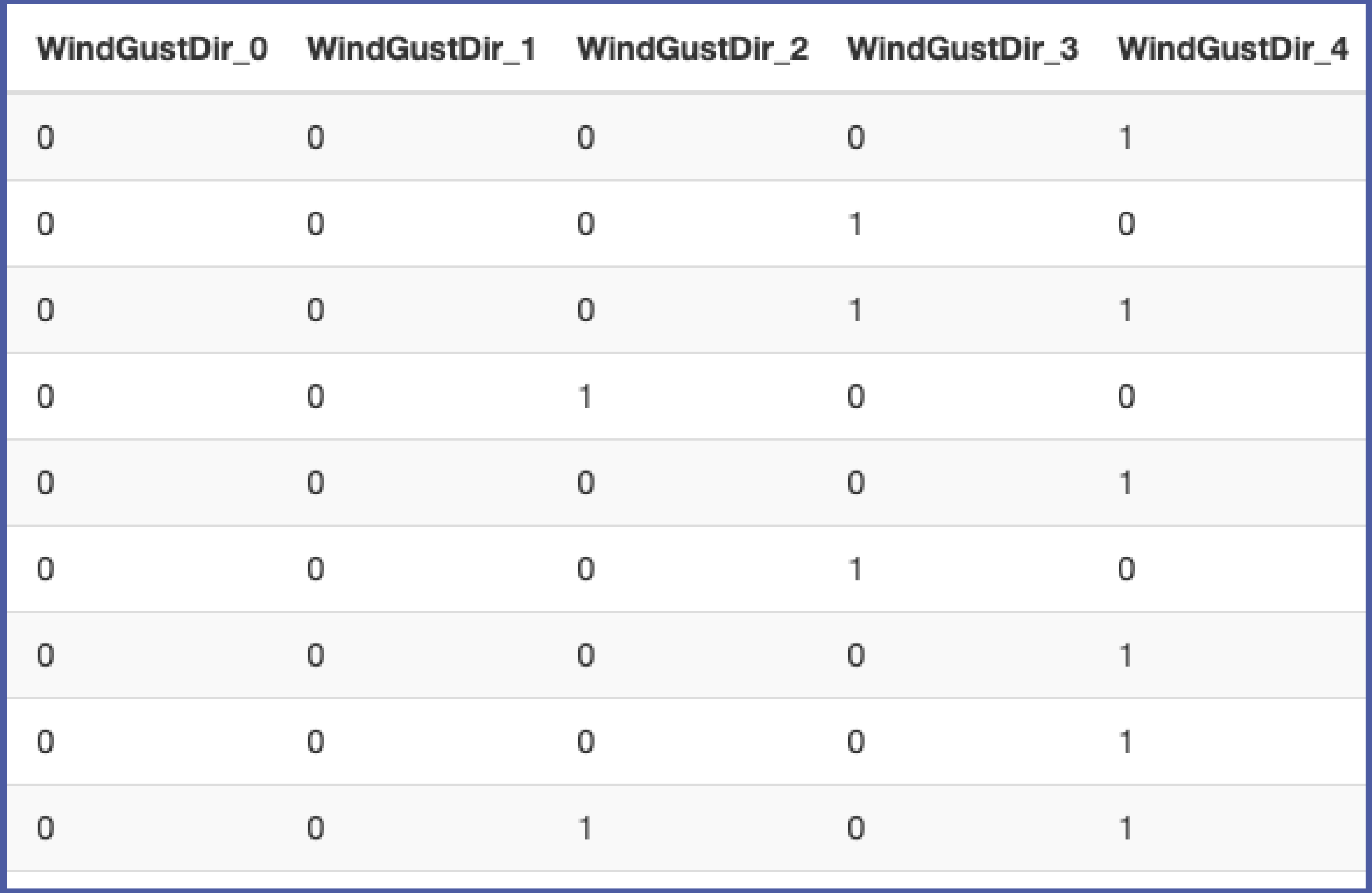
RainToday

RainTomorrow

ENCODING CATEGORICAL DATA



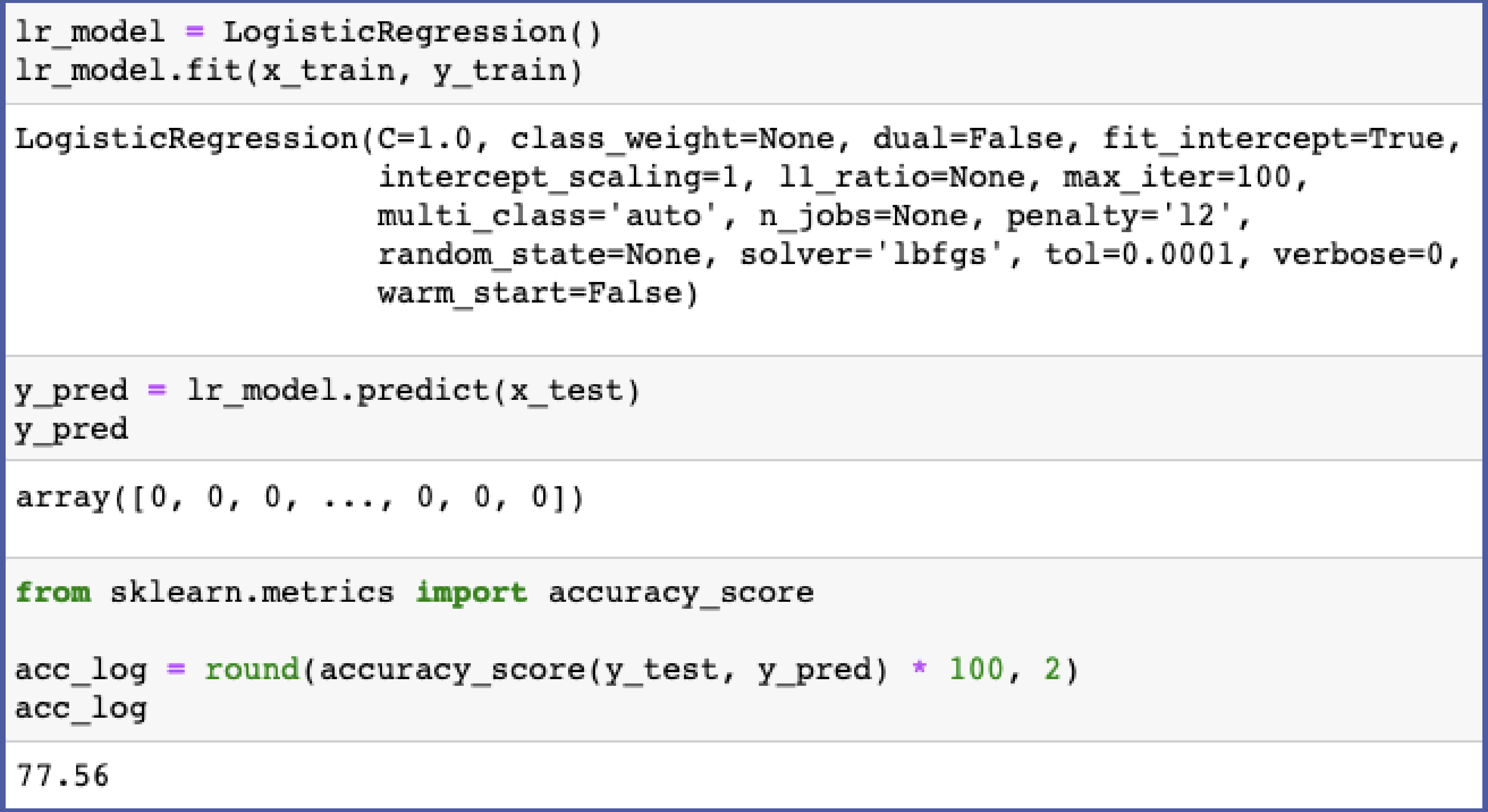
```
import category_encoders as ce
binary_encoder = ce.BinaryEncoder(cols=['WindGustDir', 'WindDir9am', 'WindDir3pm', 'Location'])
encoded_data = binary_encoder.fit_transform(data)
```



WindGustDir_0	WindGustDir_1	WindGustDir_2	WindGustDir_3	WindGustDir_4
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	0	1
0	0	0	0	1
0	0	1	0	1

LOGISTIC REGRESSION

BASIC LOGISTIC REGRESSION



```
lr_model = LogisticRegression()
lr_model.fit(x_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
y_pred = lr_model.predict(x_test)
y_pred
```

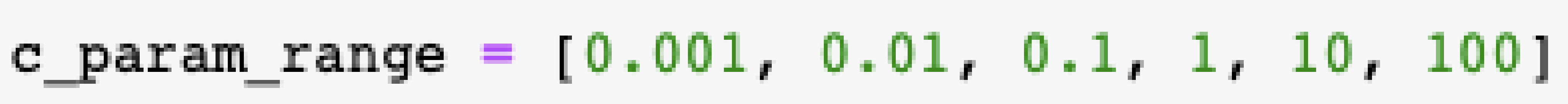
```
array([0, 0, 0, ..., 0, 0, 0])
```

```
from sklearn.metrics import accuracy_score

acc_log = round(accuracy_score(y_test, y_pred) * 100, 2)
acc_log
```

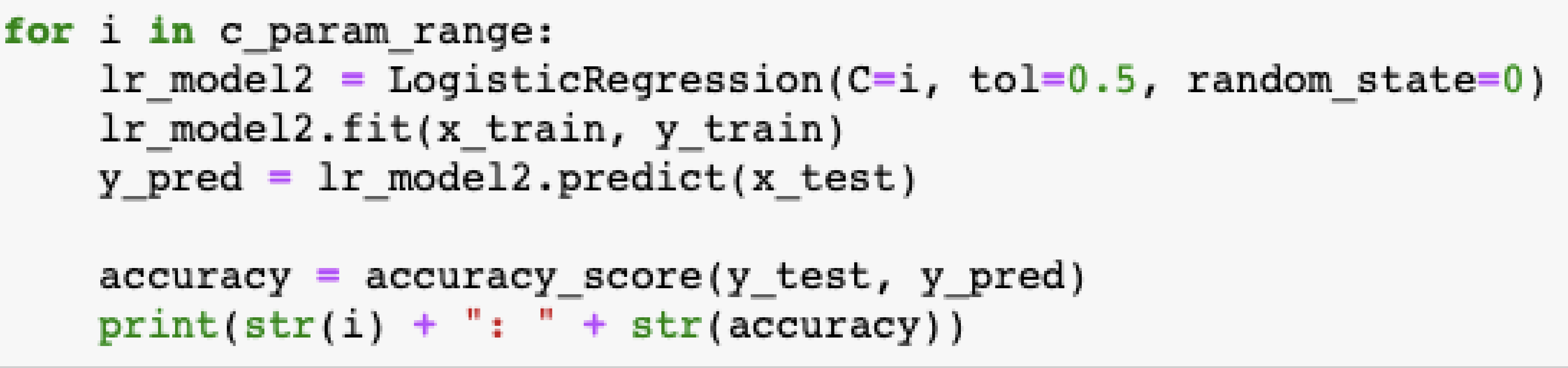
```
77.56
```

LOGISTIC REGRESSION

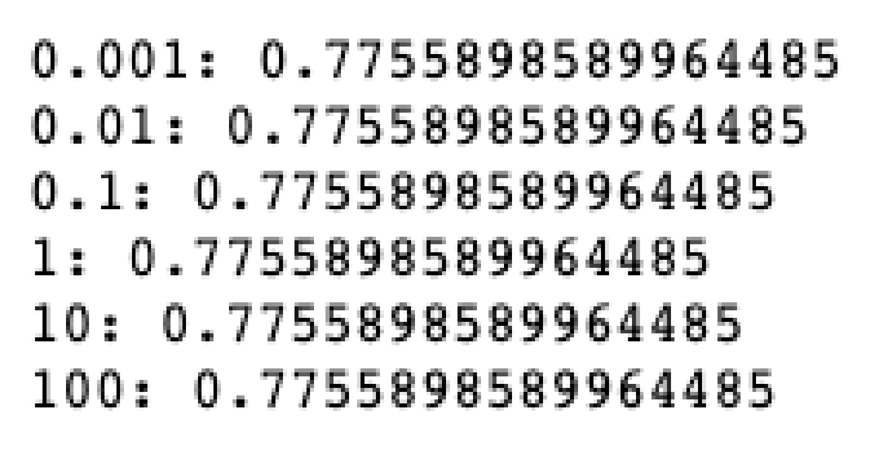
TUNING WITH C, TOL, RANDOM STATE



```
c_param_range = [0.001, 0.01, 0.1, 1, 10, 100]
```

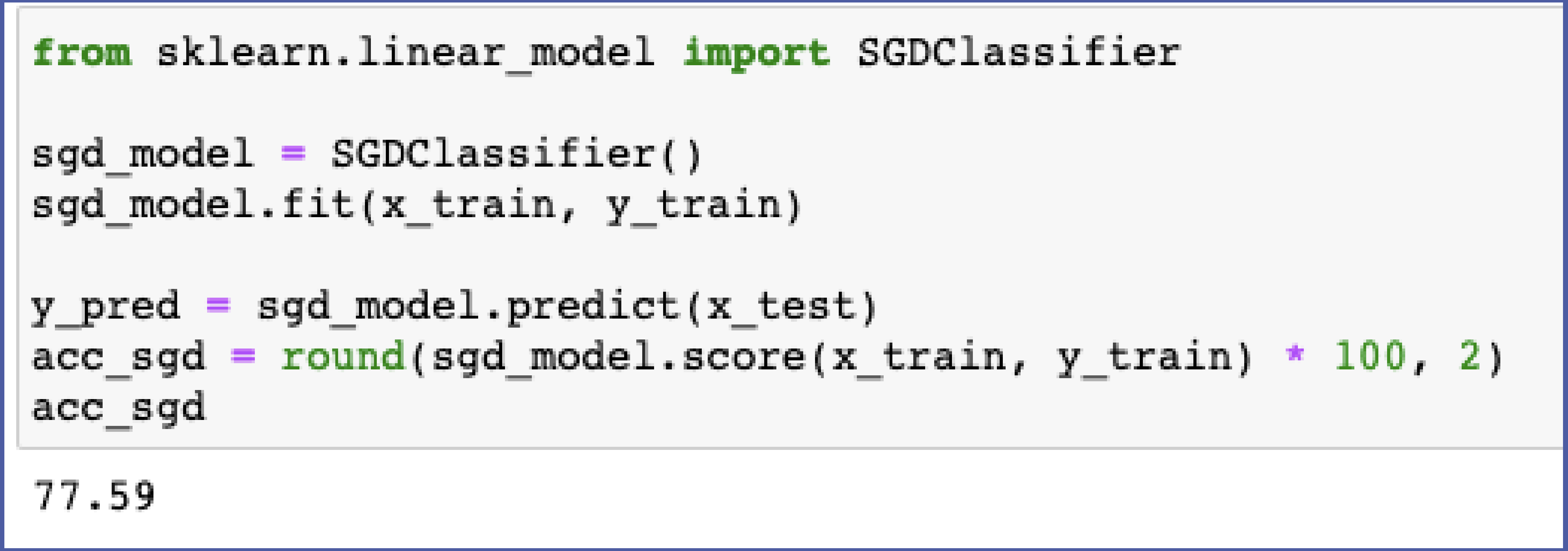


```
for i in c_param_range:
    lr_model2 = LogisticRegression(C=i, tol=0.5, random_state=0)
    lr_model2.fit(x_train, y_train)
    y_pred = lr_model2.predict(x_test)

    accuracy = accuracy_score(y_test, y_pred)
    print(str(i) + ": " + str(accuracy))
```



```
0.001: 0.7755898589964485
0.01: 0.7755898589964485
0.1: 0.7755898589964485
1: 0.7755898589964485
10: 0.7755898589964485
100: 0.7755898589964485
```

STOCHASTIC GRADIENT DESCENT

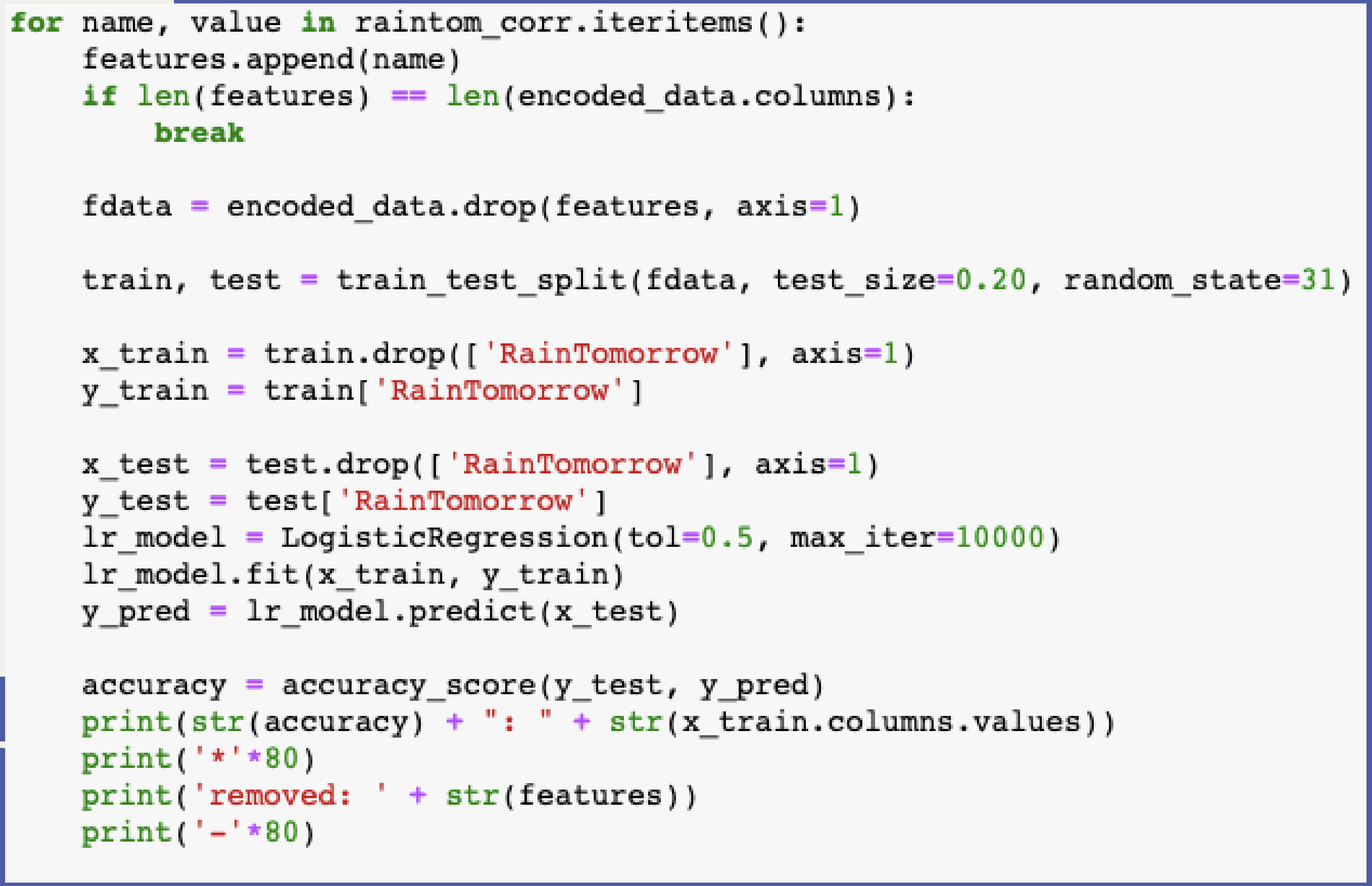


```
from sklearn.linear_model import SGDClassifier

sgd_model = SGDClassifier()
sgd_model.fit(x_train, y_train)

y_pred = sgd_model.predict(x_test)
acc_sgd = round(sgd_model.score(x_train, y_train) * 100, 2)
acc_sgd
```

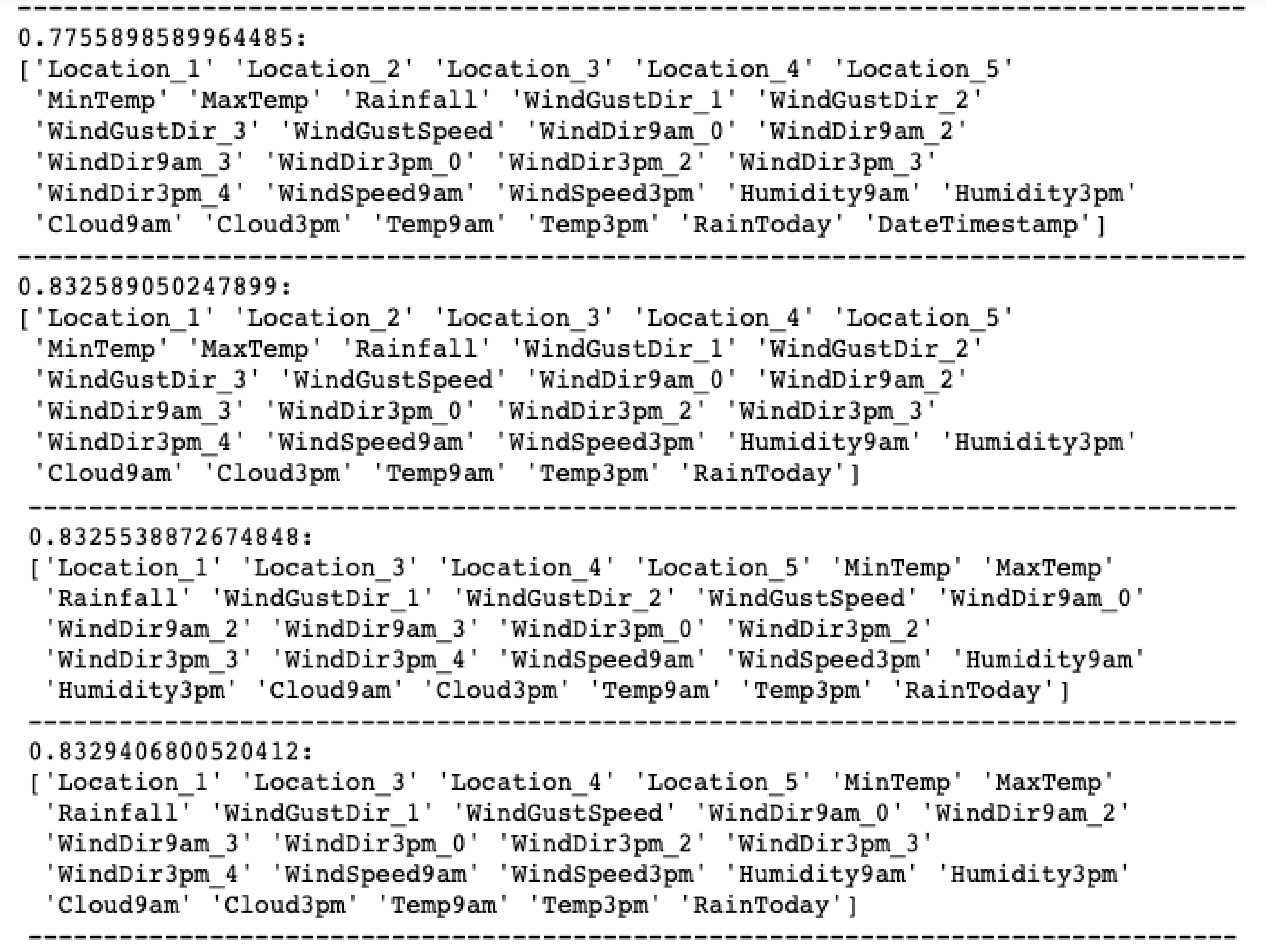
```
77.59
```

Feature Selection

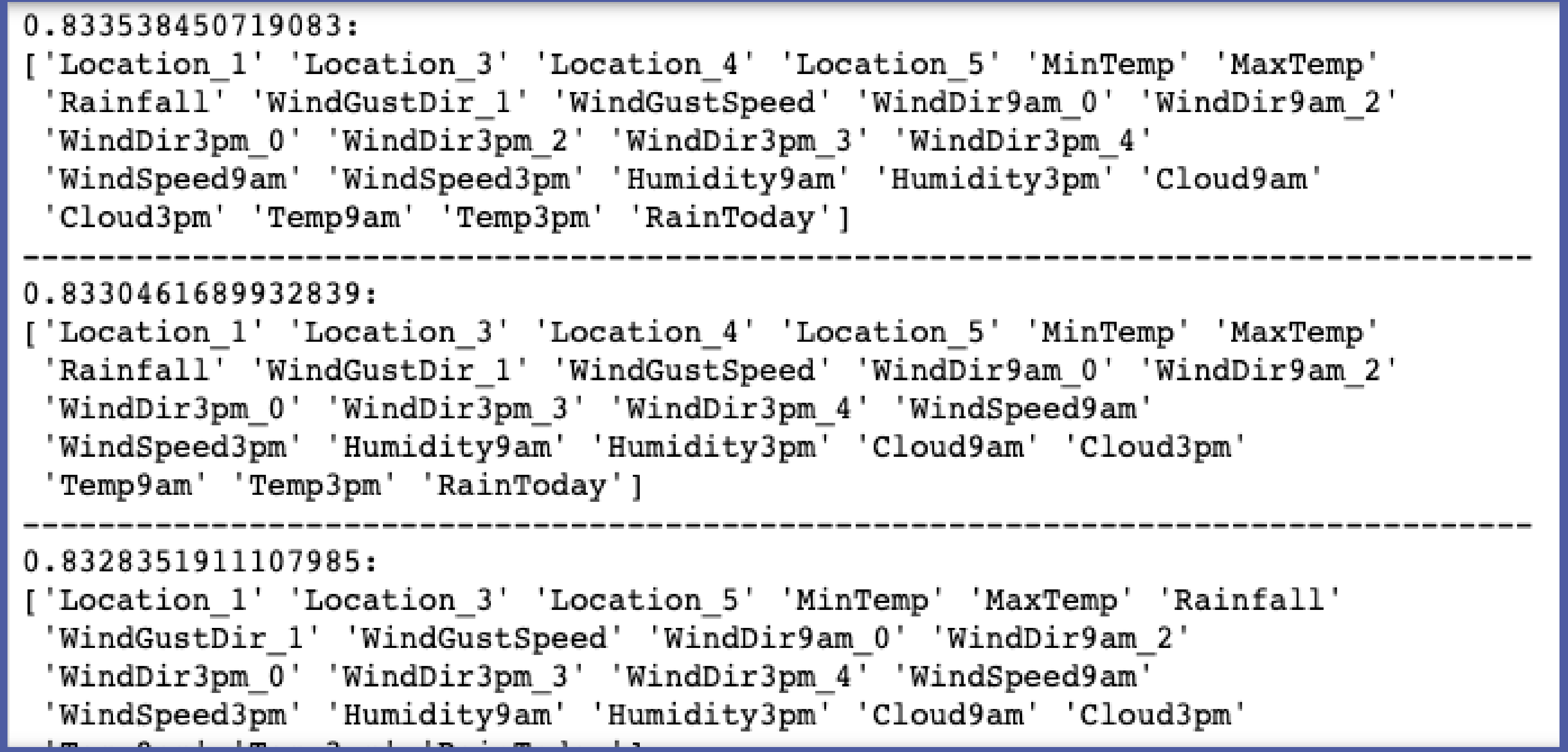
BACKWARD SELECTION



```
for name, value in raintom_corr.iteritems():
    features.append(name)
    if len(features) == len(encoded_data.columns):
        break

    fdata = encoded_data.drop(features, axis=1)

    train, test = train_test_split(fdata, test_size=0.20, random_state=31)

    x_train = train.drop(['RainTomorrow'], axis=1)
    y_train = train['RainTomorrow']

    x_test = test.drop(['RainTomorrow'], axis=1)
    y_test = test['RainTomorrow']
    lr_model = LogisticRegression(tol=0.5, max_iter=10000)
    lr_model.fit(x_train, y_train)
    y_pred = lr_model.predict(x_test)

    accuracy = accuracy_score(y_test, y_pred)
    print(str(accuracy) + ": " + str(x_train.columns.values))
    print('*'*80)
    print('removed: ' + str(features))
    print('-'*80)
```



```
--------------------------------------------------------------------------------
0.7755898589964485:
['Location_1' 'Location_2' 'Location_3' 'Location_4' 'Location_5'
 'MinTemp' 'MaxTemp' 'Rainfall' 'WindGustDir_1' 'WindGustDir_2'
 'WindGustDir_3' 'WindGustSpeed' 'WindDir9am_0' 'WindDir9am_2'
 'WindDir9am_3' 'WindDir3pm_0' 'WindDir3pm_2' 'WindDir3pm_3'
 'WindDir3pm_4' 'WindSpeed9am' 'WindSpeed3pm' 'Humidity9am' 'Humidity3pm'
 'Cloud9am' 'Cloud3pm' 'Temp9am' 'Temp3pm' 'RainToday' 'DateTimestamp']
--------------------------------------------------------------------------------
0.832589050247899:
['Location_1' 'Location_2' 'Location_3' 'Location_4' 'Location_5'
 'MinTemp' 'MaxTemp' 'Rainfall' 'WindGustDir_1' 'WindGustDir_2'
 'WindGustDir_3' 'WindGustSpeed' 'WindDir9am_0' 'WindDir9am_2'
 'WindDir9am_3' 'WindDir3pm_0' 'WindDir3pm_2' 'WindDir3pm_3'
 'WindDir3pm_4' 'WindSpeed9am' 'WindSpeed3pm' 'Humidity9am' 'Humidity3pm'
 'Cloud9am' 'Cloud3pm' 'Temp9am' 'Temp3pm' 'RainToday']
--------------------------------------------------------------------------------
0.8325538872674848:
['Location_1' 'Location_3' 'Location_4' 'Location_5' 'MinTemp' 'MaxTemp'
 'Rainfall' 'WindGustDir_1' 'WindGustDir_2' 'WindGustSpeed' 'WindDir9am_0'
 'WindDir9am_2' 'WindDir9am_3' 'WindDir3pm_0' 'WindDir3pm_2'
 'WindDir3pm_3' 'WindDir3pm_4' 'WindSpeed9am' 'WindSpeed3pm' 'Humidity9am'
 'Humidity3pm' 'Cloud9am' 'Cloud3pm' 'Temp9am' 'Temp3pm' 'RainToday']
--------------------------------------------------------------------------------
0.8329406800520412:
['Location_1' 'Location_3' 'Location_4' 'Location_5' 'MinTemp' 'MaxTemp'
 'Rainfall' 'WindGustDir_1' 'WindGustSpeed' 'WindDir9am_0' 'WindDir9am_2'
 'WindDir9am_3' 'WindDir3pm_0' 'WindDir3pm_2' 'WindDir3pm_3'
 'WindDir3pm_4' 'WindSpeed9am' 'WindSpeed3pm' 'Humidity9am' 'Humidity3pm'
 'Cloud9am' 'Cloud3pm' 'Temp9am' 'Temp3pm' 'RainToday']
--------------------------------------------------------------------------------
```



```
0.833538450719083:
['Location_1' 'Location_3' 'Location_4' 'Location_5' 'MinTemp' 'MaxTemp'
 'Rainfall' 'WindGustDir_1' 'WindGustSpeed' 'WindDir9am_0' 'WindDir9am_2'
 'WindDir3pm_0' 'WindDir3pm_2' 'WindDir3pm_3' 'WindDir3pm_4'
 'WindSpeed9am' 'WindSpeed3pm' 'Humidity9am' 'Humidity3pm' 'Cloud9am'
 'Cloud3pm' 'Temp9am' 'Temp3pm' 'RainToday']
--------------------------------------------------------------------------------
0.8330461689932839:
['Location_1' 'Location_3' 'Location_4' 'Location_5' 'MinTemp' 'MaxTemp'
 'Rainfall' 'WindGustDir_1' 'WindGustSpeed' 'WindDir9am_0' 'WindDir9am_2'
 'WindDir3pm_0' 'WindDir3pm_3' 'WindDir3pm_4' 'WindSpeed9am'
 'WindSpeed3pm' 'Humidity9am' 'Humidity3pm' 'Cloud9am' 'Cloud3pm'
 'Temp9am' 'Temp3pm' 'RainToday']
--------------------------------------------------------------------------------
0.8328351911107985:
['Location_1' 'Location_3' 'Location_5' 'MinTemp' 'MaxTemp' 'Rainfall'
 'WindGustDir_1' 'WindGustSpeed' 'WindDir9am_0' 'WindDir9am_2'
 'WindDir3pm_0' 'WindDir3pm_3' 'WindDir3pm_4' 'WindSpeed9am'
 'WindSpeed3pm' 'Humidity9am' 'Humidity3pm' 'Cloud9am' 'Cloud3pm'
 'Temp9am' 'Temp3pm' 'RainToday']
```

Summary

- ## Binary Encoding

  category_encoders package in Python.

- ## Don't blindly fillna with zeros

  Zero temperature doesn't make sense in Australia does it? or does it?

- ## Binary vs Dummy variables

  Depending on your data one might make more sense than the other.

- ## Feature Selection

  Greatly improves accuracy. (77.59 -> 83.35)

References

http://www.bom.gov.au/climate/data

Copyright Commonwealth of Australia 2010, Bureau of Meteorology.

https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

Rain in Australia: Predict rain tomorrow in Australia

Thank you

Questions?