

weather-prediction-using-cnn

March 26, 2024

```
[1]: import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

C:\Users\happy\anaconda3\lib\site-packages\scipy__init__.py:155: UserWarning: A NumPy version >=1.18.5 and <1.25.0 is required for this version of SciPy (detected version 1.26.4
warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}")

```
[2]: df = pd.read_csv('weather_prediction_dataset.csv')
df_bbq = pd.read_csv('weather_prediction_bbq_labels.csv')
```

```
[3]: df.head(6)
```

```
[3]:      DATE  MONTH  BASEL_cloud_cover  BASEL_humidity  BASEL_pressure  \
0  20000101      1              8           0.89         1.0286
1  20000102      1              8           0.87         1.0318
2  20000103      1              5           0.81         1.0314
3  20000104      1              7           0.79         1.0262
4  20000105      1              5           0.90         1.0246
5  20000106      1              3           0.85         1.0244
```

```
      BASEL_global_radiation  BASEL_precipitation  BASEL_sunshine  \
0              0.20              0.03              0.0
1              0.25              0.00              0.0
2              0.50              0.00              3.7
3              0.63              0.35              6.9
4              0.51              0.07              3.7
5              0.56              0.00              5.7
```

```
      BASEL_temp_mean  BASEL_temp_min  ...  STOCKHOLM_temp_min  \
0              2.9              1.6  ...              -9.3
1              3.6              2.7  ...              0.5
2              2.2              0.1  ...             -1.0
3              3.9              0.5  ...              2.5
4              6.0              3.8  ...             -1.8
```

```

5          4.2          1.9 ...          -0.6

STOCKHOLM_temp_max TOURS_wind_speed TOURS_humidity TOURS_pressure \
0          0.7          1.6          0.97          1.0275
1          2.0          2.0          0.99          1.0293
2          2.8          3.4          0.91          1.0267
3          4.6          4.9          0.95          1.0222
4          2.9          3.6          0.95          1.0209
5          4.0          3.4          0.92          1.0209

TOURS_global_radiation TOURS_precipitation TOURS_temp_mean \
0          0.25          0.04          8.5
1          0.17          0.16          7.9
2          0.27          0.00          8.1
3          0.11          0.44          8.6
4          0.39          0.04          8.0
5          0.55          0.20          7.1

TOURS_temp_min TOURS_temp_max
0          7.2          9.8
1          6.6          9.2
2          6.6          9.6
3          6.4          10.8
4          6.4          9.5
5          3.5          10.7

[6 rows x 165 columns]

```

```
[4]: df_bbq.head(6)
```

```

[4]:      DATE  BASEL_BBQ_weather  BUDAPEST_BBQ_weather  DE_BBQ_weather \
0  20000101          False          False          False
1  20000102          False          False          False
2  20000103          False          False          False
3  20000104          False          False          False
4  20000105          False          False          False
5  20000106          False          False          False

DRESDEN_BBQ_weather  DUSSELDORF_BBQ_weather  HEATHROW_BBQ_weather \
0          False          False          False
1          False          False          False
2          False          False          False
3          False          False          False
4          False          False          False
5          False          False          False

KASSEL_BBQ_weather  LJUBLJANA_BBQ_weather  MAASTRICHT_BBQ_weather \

```

0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False

	MALMO_BBQ_weather	MONTELMAR_BBQ_weather	MUENCHEN_BBQ_weather	\
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	
5	False	False	False	

	OSLO_BBQ_weather	PERPIGNAN_BBQ_weather	SONNBLICK_BBQ_weather	\
0	False	False	False	
1	False	False	False	
2	False	False	False	
3	False	False	False	
4	False	False	False	
5	False	False	False	

	STOCKHOLM_BBQ_weather	TOURS_BBQ_weather
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False

```
[5]: len(df_bbq.columns), df_bbq.columns
```

```
[5]: (18,
      Index(['DATE', 'BASEL_BBQ_weather', 'BUDAPEST_BBQ_weather', 'DE_BBQ_weather',
            'DRESDEN_BBQ_weather', 'DUSSELDORF_BBQ_weather', 'HEATHROW_BBQ_weather',
            'KASSEL_BBQ_weather', 'LJUBLJANA_BBQ_weather', 'MAASTRICHT_BBQ_weather',
            'MALMO_BBQ_weather', 'MONTELMAR_BBQ_weather', 'MUENCHEN_BBQ_weather',
            'OSLO_BBQ_weather', 'PERPIGNAN_BBQ_weather', 'SONNBLICK_BBQ_weather',
            'STOCKHOLM_BBQ_weather', 'TOURS_BBQ_weather'],
            dtype='object'))
```

```
[6]: df.columns
```

```
[6]: Index(['DATE', 'MONTH', 'BASEL_cloud_cover', 'BASEL_humidity',
            'BASEL_pressure', 'BASEL_global_radiation', 'BASEL_precipitation',
            'BASEL_sunshine', 'BASEL_temp_mean', 'BASEL_temp_min',
```

```
...
'STOCKHOLM_temp_min', 'STOCKHOLM_temp_max', 'TOURS_wind_speed',
'TOURS_humidity', 'TOURS_pressure', 'TOURS_global_radiation',
'TOURS_precipitation', 'TOURS_temp_mean', 'TOURS_temp_min',
'TOURS_temp_max'],
dtype='object', length=165)
```

1 Dresden Weather

```
[7]: df.columns[30:41]
```

```
[7]: Index(['DRESDEN_cloud_cover', 'DRESDEN_wind_speed', 'DRESDEN_wind_gust',
'DRESDEN_humidity', 'DRESDEN_global_radiation', 'DRESDEN_precipitation',
'DRESDEN_sunshine', 'DRESDEN_temp_mean', 'DRESDEN_temp_min',
'DRESDEN_temp_max', 'DUSSELDORF_cloud_cover'],
dtype='object')
```

```
[8]: df_Dresden = df[['DATE', 'DRESDEN_cloud_cover', 'DRESDEN_wind_speed',
↪ 'DRESDEN_wind_gust',
'DRESDEN_humidity', 'DRESDEN_global_radiation', 'DRESDEN_precipitation',
'DRESDEN_sunshine', 'DRESDEN_temp_mean', 'DRESDEN_temp_min',
'DRESDEN_temp_max']]
```

```
[9]: df_Dresden['BBQ'] = df_bbq['DRESDEN_BBQ_weather'];
```

C:\Users\happy\AppData\Local\Temp\ipykernel_6012\2405519419.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_Dresden['BBQ'] = df_bbq['DRESDEN_BBQ_weather'];
```

```
[10]: df_Dresden.head()
```

```
[10]:
```

	DATE	DRESDEN_cloud_cover	DRESDEN_wind_speed	DRESDEN_wind_gust	\
0	20000101	8	3.2	7.2	
1	20000102	7	4.0	8.8	
2	20000103	7	5.4	12.1	
3	20000104	8	6.0	14.4	
4	20000105	2	5.6	15.8	

	DRESDEN_humidity	DRESDEN_global_radiation	DRESDEN_precipitation	\
0	0.89	0.09	0.32	
1	0.89	0.23	0.00	

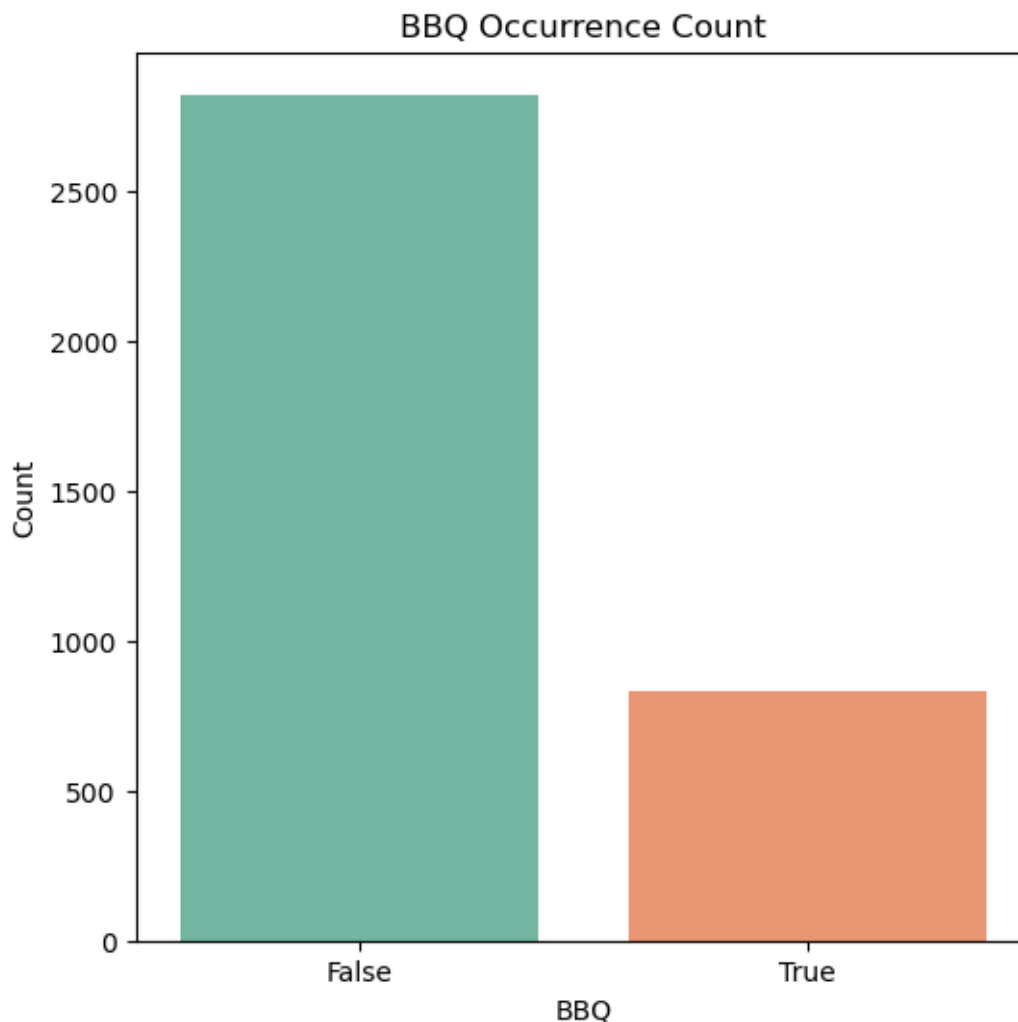
2	0.79		0.18	0.00
3	0.88		0.11	0.22
4	0.76		0.49	0.00

	DRESDEN_sunshine	DRESDEN_temp_mean	DRESDEN_temp_min	DRESDEN_temp_max	\
0	0.0	1.0	-1.8	2.0	
1	0.4	2.5	1.4	4.0	
2	0.0	4.2	1.3	5.1	
3	0.0	4.4	3.4	5.2	
4	5.7	1.8	-0.5	6.9	

	BBQ
0	False
1	False
2	False
3	False
4	False

2 Data Visualization

```
[11]: plt.figure(figsize=(6, 6))
sns.countplot(data=df_Dresden, x='BBQ', palette='Set2')
plt.title('BBQ Occurrence Count')
plt.xlabel('BBQ')
plt.ylabel('Count')
plt.show()
```



3 Data Preperation

```
[12]: df_Dresden['BBQ'] = df_Dresden['BBQ'].map({True:1, False:0});
```

C:\Users\happy\AppData\Local\Temp\ipykernel_6012\2977729730.py:1:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using `.loc[row_indexer,col_indexer] = value` instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
df_Dresden['BBQ'] = df_Dresden['BBQ'].map({True:1, False:0});
```

```
[13]: import tensorflow as tf
from tensorflow.keras.layers import Dense, Dropout, Conv1D
from tensorflow.keras.models import Sequential
from tensorflow.keras.optimizers import Adam

from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, \
    accuracy_score

[14]: X = df_Dresden.drop(['DATE', 'BBQ'], axis = 1)
y = df_Dresden['BBQ']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, \
    random_state = 28)
len(X_train), len(X_test)
```

```
[14]: (2923, 731)
```

4 CNN Model Building

```
[15]: from tensorflow.keras.layers import Conv1D, Flatten, Reshape
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

X_train.shape, X_test.shape

model1 = Sequential([
    Reshape((1, X_train.shape[1], 1)),
    Conv1D(filters=10, kernel_size=2, activation='relu', input_shape=(1, \
    X_train.shape[1], 1)),
    Flatten(),
    Dense(1, activation='sigmoid')
])
```

C:\Users\happy\anaconda3\lib\site-packages\keras\src\layers\convolutional\base_conv.py:99: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.

```
super().__init__(
```

```
[16]: model1.compile(loss='binary_crossentropy', optimizer='adam',
    metrics=['accuracy'])
```

```
[17]: from tensorflow.keras.layers import Conv1D, Flatten, Reshape
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

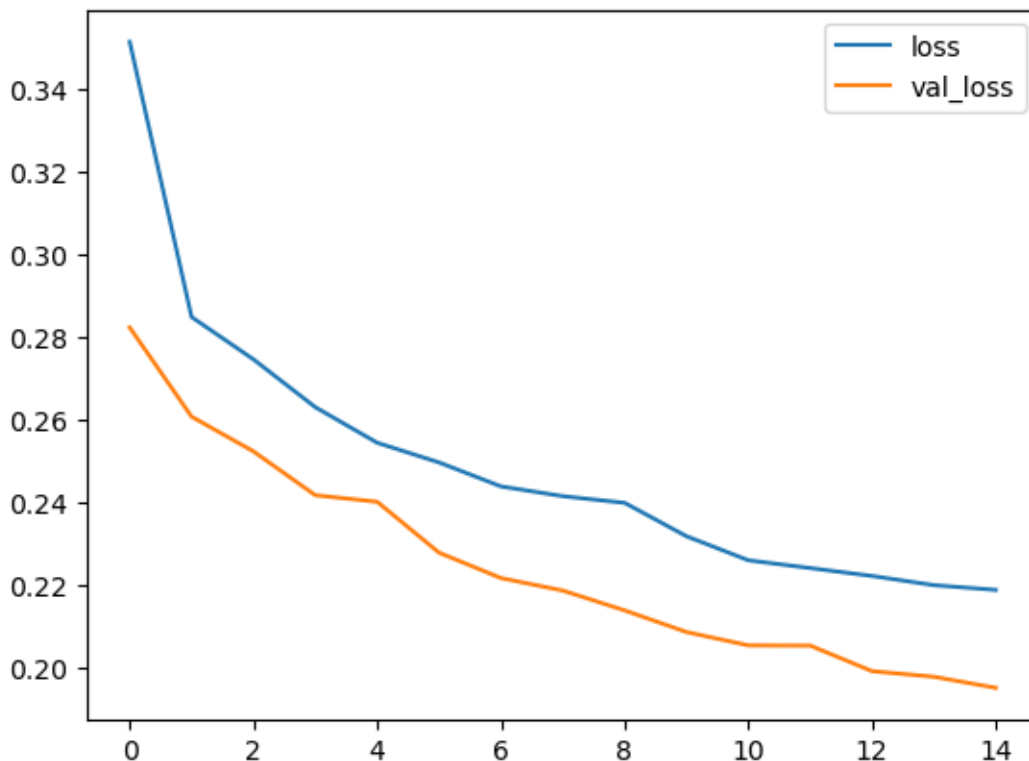
```
model1 = Sequential([
    Reshape((X_train.shape[1], 1)), # Adjust the reshape layer
    Conv1D(filters=10, kernel_size=2, activation='relu'),
    Flatten(),
    Dense(1, activation='sigmoid')
])
```

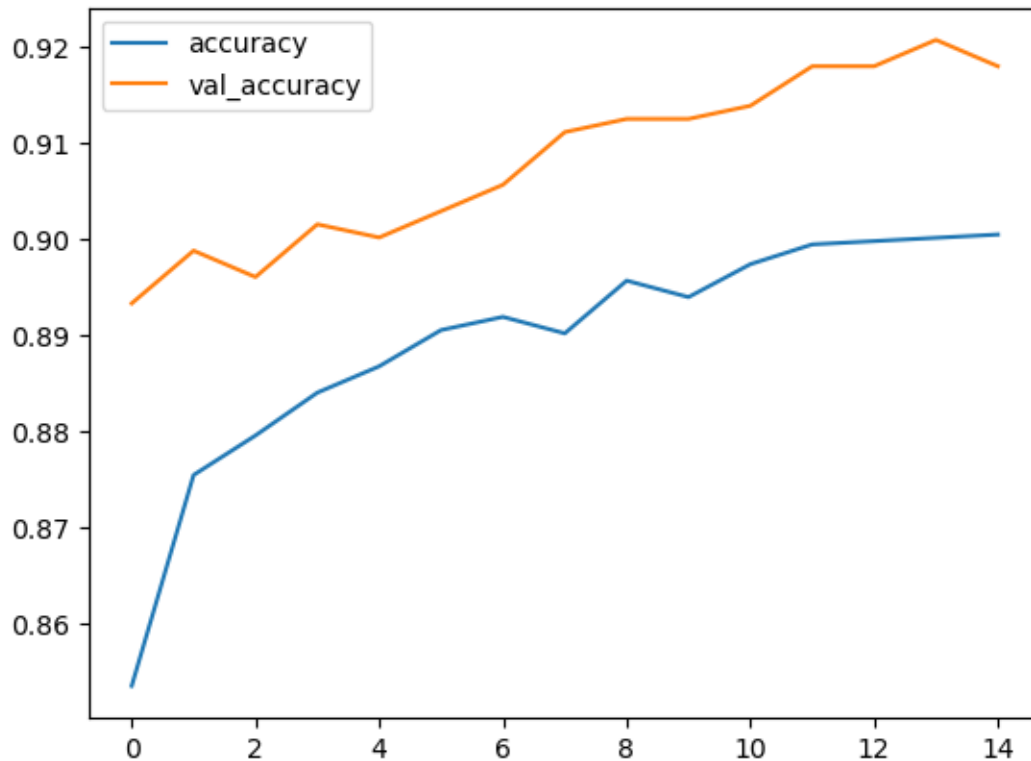
```
[18]: model1.compile(optimizer='adam', loss='binary_crossentropy',
    ↪ metrics=['accuracy'])
model1.fit(x=X_train, y=y_train, epochs=15, validation_data=(X_test, y_test),
    ↪ verbose=0)
```

[18]: <keras.src.callbacks.history.History at 0x2420027bbe0>

```
[19]: model1_loss_acc = pd.DataFrame(model1.history.history)
model1_loss_acc[['loss', 'val_loss']].plot()
model1_loss_acc[['accuracy', 'val_accuracy']].plot()
```

[19]: <AxesSubplot:>





```
[20]: pred_1 = np.round(model1.predict(X_test))
      print(classification_report(y_test, pred_1))
```

```
23/23          0s 6ms/step
          precision    recall  f1-score   support

           0       0.95      0.95      0.95         574
           1       0.81      0.80      0.81         157

   accuracy                   0.92         731
  macro avg       0.88      0.88      0.88         731
 weighted avg       0.92      0.92      0.92         731
```

```
[21]: cnn_acc = accuracy_score(y_test, pred_1)
      cnn_acc
```

```
[21]: 0.9179206566347469
```

```
[22]: plt.figure(figsize = (6,6))
      sns.heatmap(confusion_matrix(y_test, pred_1), annot=True)
      plt.title('Confusion matrix for predicting barbecue weather with CNN');
```

Confusion matrix for predicting barbecue weather with CNN

