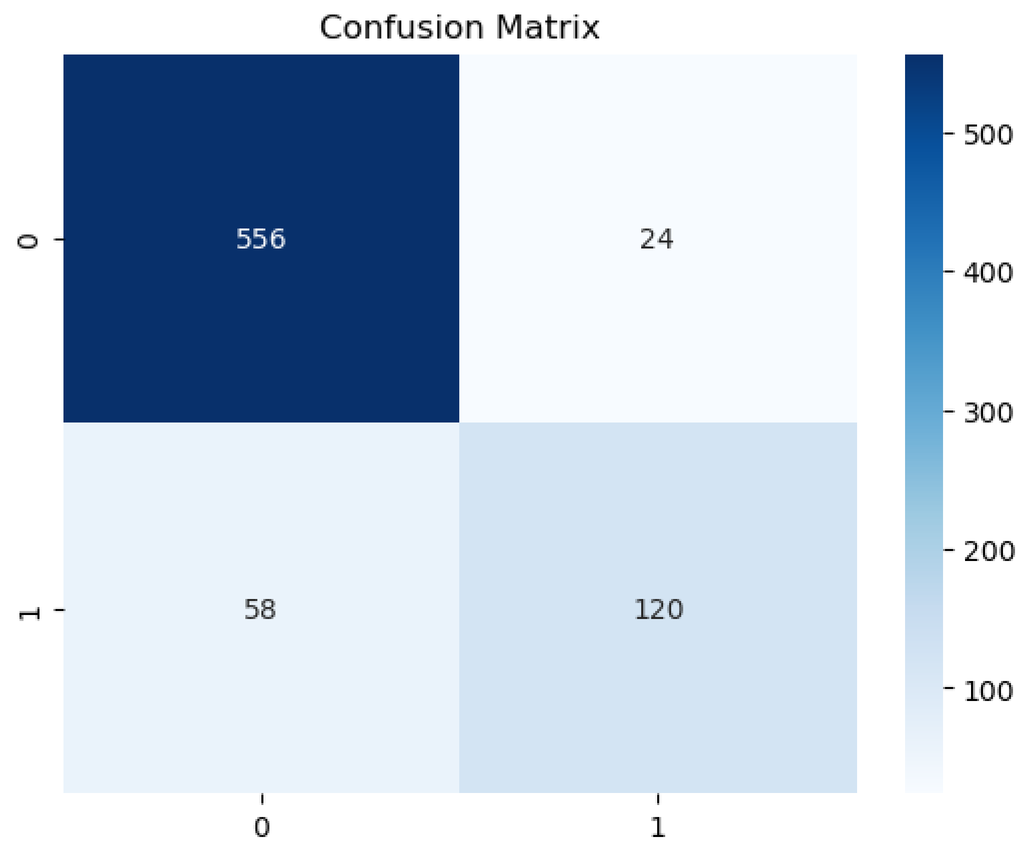
In [6]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.preprocessing import MinMaxScaler, OneHotEncoder
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import StandardScaler, OneHotEncoder

# Step 2: Load Data
link = "https://raw.githubusercontent.com/dsrscientist/dataset3/main/weatherAUS.csv"
ta = pd.read_csv(link)

# Step 3: Exploratory Data Analysis (EDA)
# (No changes here)

# Step 4: Data Preprocessing and Feature Engineering
ta.dropna(inplace=True)
ta.reset_index(drop=True, inplace=True)

# Extract features from the date column
ta['Date'] = pd.to_datetime(ta['Date'])
ta['Year'] = ta['Date'].dt.year
ta['Month'] = ta['Date'].dt.month
ta['Day'] = ta['Date'].dt.day

# Perform feature scaling
scale = MinMaxScaler()
scale_features = ['MinTemp', 'MaxTemp', 'Rainfall', 'Evaporation', 'Sunshine', 'WindGustSpeed',
                    'WindSpeed9am', 'WindSpeed3pm', 'Humidity9am', 'Humidity3pm', 'Pressure9am',
                    'Pressure3pm', 'Cloud9am', 'Cloud3pm', 'Temp9am', 'Temp3pm']
ta[scaled_features] = scaler.fit_transform(data[scaled_features])

# Split the Dataset
feature= ta.drop(['RainTomorrow', 'Date', 'Location'], axis=1)
target= ta['RainTomorrow']

X_train, X_test, y_train, y_test = train_test_split(feature, target, test_size=0.2, random_state=42)

# Step 5: Build Random Forest Model in a pipeline
numeric_features = feature.select_dtypes(include=['int64', 'float64']).columns
categorical_features = feature.select_dtypes(include=['object']).columns

numeric_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', StandardScaler())
])

categorical_transformer = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='constant', fill_value='missing')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)
    ])

model = Pipeline([
    ('preprocessor', preprocessor),
    ('classifier', RandomForestClassifier(random_state=42))
])

# Fit the model
model.fit(X_train, y_train)

# Step 6: Evaluate the Model
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))

# Plot Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.show()
```

```
Accuracy: 0.8918205804749341
Classification Report:
               precision    recall  f1-score   support

          No       0.91      0.96      0.93       580
         Yes       0.83      0.67      0.75       178

    accuracy                           0.89       758
   macro avg       0.87      0.82      0.84       758
weighted avg       0.89      0.89      0.89       758
```



Title: Rain Prediction Model Documentation

1. DATA knowledge:
    - This project provides documentation for the Rain Prediction Model, which forcasts whether it will rain tomorrow.
    - The model is built using a Random Forest Classifier and contain various data preprocessing steps.
2. Import Libraries :
    - Pandas: Data manipulation and analysis.
    - Numpy: Numerical operations.
    - Matplotlib, seaborn: Data visualization.
    - Scikit-learn: Machine learning tools.
3. Source data:
    - The model uses the weatherAUS dataset from the given link.
    - The model uses the weatherAUS dataset from the given link.
4. Exploratory Data Analysis :
    - No changes were made to the EDA section; _it is assumed to have been performed separately.
5. Data Preprocessing and Feature Engineering: a. Handling Missing Values:
    - Rows with missing values were droppe. b. Extracting Features from Date:
    - 'Year', 'Month', and 'Day' feature were extracted from the 'Date' columns. c. Feature Scaling:
    - Min-Max scaling was applied to selecte features
6. Data Splitting:
    - The data was split into training and testing sets.
7. Building the Random Forest Model in a Pipeline: a. Numeric Transformer:
    - Imputed missing values with the mean.
    - Applied Standard Scaling to numeric features. b. Categorical Transformer:
    - Imputed missing values with a constant value.
    - Applied one-hot encoding to categorical features. c. Column Transformer:
    - Applied transformers to numeric and categorical features. d. Model Pipeline:
    - Combined preprocessing and RandomForestClassifier in a pipeline.
8. Model Train:
    - The model was trained using the training dataset.
9. Model perfomance: a. Accuracy Score:
    - Calculated the accuracy of the model on the test set. b. Classification Report:
    - Provided a detailed classification report.
10. Confusion Matrix:
    - Plotted a confusion matrix to visualize model performance.
11. Conclusion:
    - The Rain Prediction Model has been inflawlessely built and evaluated.