Machine Faliure Prediction

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
data = pd.read_csv('data.csv')
print(data.shape)
```

```
(944, 10)
```

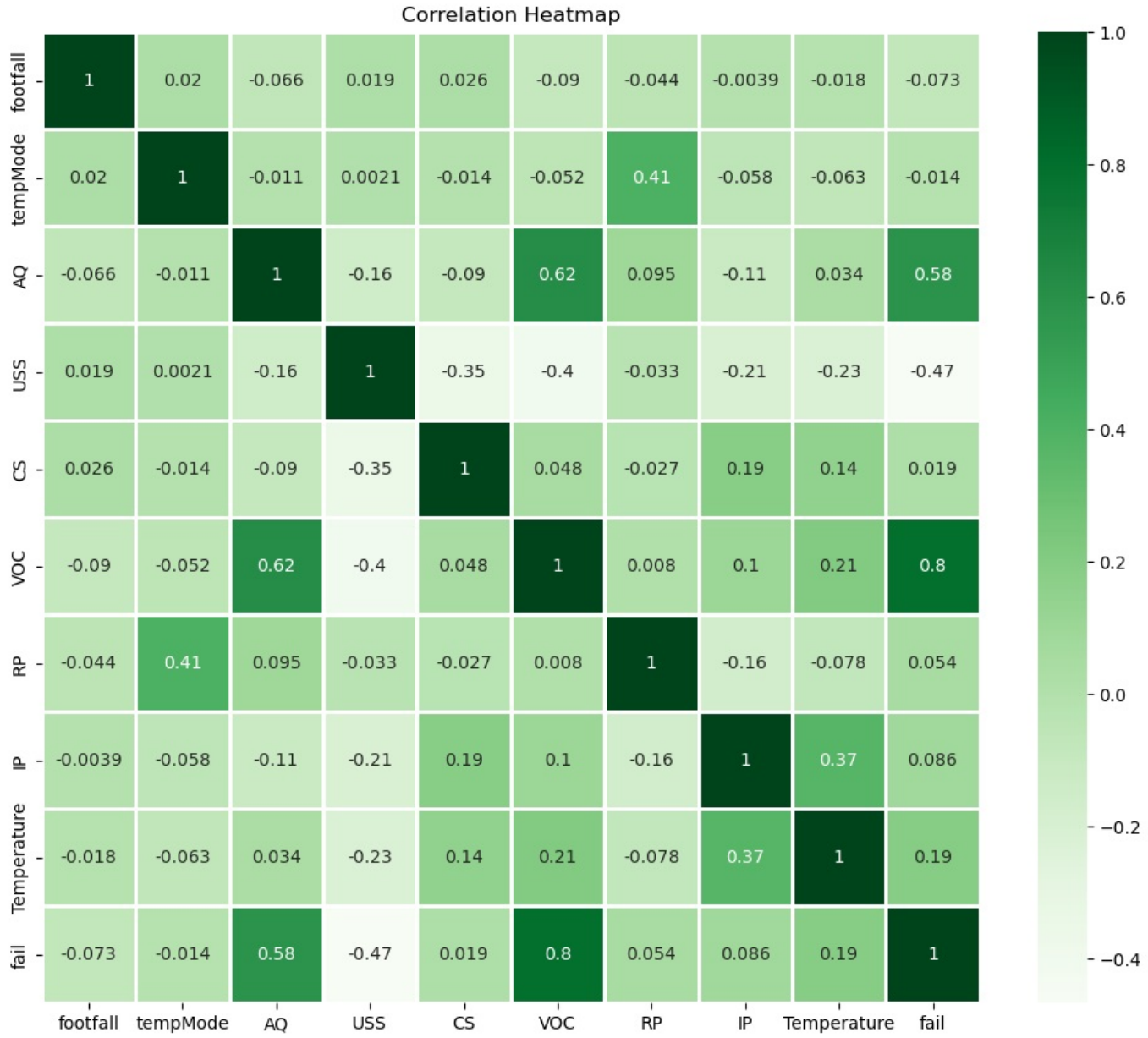
In [4]:
```
print(data.describe)
print(data.info)
```

```
<bound method NDFrame.describe of      footfall  tempMode  AQ  USS  CS  VOC  RP  IP  Temperature  fail
0           0         7   7    1   6    6  36   3            1     1
1         190         1   3    3   5    1  20   4            1     0
2          31         7   2    2   6    1  24   6            1     0
3          83         4   3    4   5    1  28   6            1     0
4         640         7   5    6   4    0  68   6            1     0
..        ...       ...  ..  ...  ..  ...  ..  ..          ...   ...
939         0         7   7    1   6    4  73   6           24     1
940         0         7   5    2   6    6  50   6           24     1
941         0         3   6    2   7    5  43   6           24     1
942         0         6   6    2   5    6  46   7           24     1
943        18         7   4    2   6    3  61   7           24     1

[944 rows x 10 columns]>
<bound method DataFrame.info of      footfall  tempMode  AQ  USS  CS  VOC  RP  IP  Temperature  fail
0           0         7   7    1   6    6  36   3            1     1
1         190         1   3    3   5    1  20   4            1     0
2          31         7   2    2   6    1  24   6            1     0
3          83         4   3    4   5    1  28   6            1     0
4         640         7   5    6   4    0  68   6            1     0
..        ...       ...  ..  ...  ..  ...  ..  ..          ...   ...
939         0         7   7    1   6    4  73   6           24     1
940         0         7   5    2   6    6  50   6           24     1
941         0         3   6    2   7    5  43   6           24     1
942         0         6   6    2   5    6  46   7           24     1
943        18         7   4    2   6    3  61   7           24     1

[944 rows x 10 columns]>
```

In [5]:
```
# Data cleaning

missing_value = data.isnull().sum()
missing_value.info
```

Out[5]:
```
<bound method Series.info of footfall       0
tempMode       0
AQ             0
USS            0
CS             0
VOC            0
RP             0
IP             0
Temperature    0
fail           0
dtype: int64>
```

In [6]:
```
# Data Vizualization using Correlation heatmap

corr_matrix = data.corr()

plt.figure(figsize=(12, 10))
sns.heatmap(corr_matrix, annot=True, cmap='Greens', linewidths=0.8)
plt.title('Correlation Heatmap')
plt.show()
```



In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = data.drop(columns=['fail'])
y = data['fail']

# Normalize the continuous variable (footfall)
scaler = StandardScaler()
X['footfall'] = scaler.fit_transform(X[['footfall']])

# Split the dataset into training and testing
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

X_train.shape, X_test.shape
```

Out[7]: ((755, 9), (189, 9))

In [8]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix, roc_auc_score, classification_report

logistic_reg = LogisticRegression()

# Train the model
logistic_reg.fit(X_train, y_train)

# Making predictions
y_prediction = logistic_reg.predict(X_test)
y_prediction_proba = logistic_reg.predict_proba(X_test)[:, 1]
```

In [9]:
```
# Evaluation of model
accuracy = accuracy_score(y_test, y_prediction)
conf_matrix = confusion_matrix(y_test, y_prediction)
roc_auc = roc_auc_score(y_test, y_prediction_proba)
classification_rep = classification_report(y_test, y_prediction)
```

```
print(f'Accuracy: {accuracy}\n')
print(f'Confusion Matrix:\n{conf_matrix}\n')
print(f'ROC-AUC Score: {roc_auc}\n')
print(f'Classification Report:\n{classification_rep}')
```

```
Accuracy: 0.8783068783068783

Confusion Matrix:
[[95 17]
 [ 6 71]]

ROC-AUC Score: 0.9464285714285714

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.85      0.89       112
           1       0.81      0.92      0.86        77

    accuracy                           0.88       189
   macro avg       0.87      0.89      0.88       189
weighted avg       0.89      0.88      0.88       189
```