```
import pandas as pd
import os

df = pd.read_csv(r"C:\Users\scott\Downloads\archive\
bot_detection_data.csv")
df
```

```
       User ID         Username  \
0       132131            flong
1       289683   hinesstephanie
2       779715       roberttran
3       696168            pmason
4       704441            noah87
...        ...               ...
49995   491196            uberg
49996   739297     jessicamunoz
49997   674475   lynncunningham
49998   167081  richardthompson
49999   311204         daniel29

                                                   Tweet  Retweet
Count  \
0      Station activity person against natural majori...
85
1      Authority research natural life material staff...
55
2      Manage whose quickly especially foot none to g...
6
3      Just cover eight opportunity strong policy which.
54
4                          Animal sign six data good or.
26
...                                                  ...              ..
.
49995  Want but put card direction know miss former h...
64
49996  Provide whole maybe agree church respond most ...
18
49997  Bring different everyone international capital...
43
49998  Than about single generation itself seek sell ...
45
49999  Here morning class various room human true bec...
91

       Mention Count  Follower Count  Verified  Bot Label
Location  \
0                  1            2353     False          1
Adkinston
1                  5            9617      True          0
```

```
Sanderston
2                  2            4363      True           0
Harrisonfurt
3                  5            2242      True           1
Martinezberg
4                  3            8438     False           1
Camachoville
...              ...             ...       ...         ...
...
49995              0            9911      True           1  Lake
Kimberlyburgh
49996              5            9900     False           1
Greenbury
49997              3            6313      True           1
Deborahfort
49998              1            6343     False           0
Stephenside
49999              4            4006     False           0
Novakberg

                 Created At                           Hashtags
0      2020-05-11 15:29:50                                NaN
1      2022-11-26 05:18:10                          both live
2      2022-08-08 03:16:54                        phone ahead
3      2021-08-14 22:27:05                ever quickly new I
4      2020-04-13 21:24:21                   foreign mention
...                    ...                                ...
49995  2023-04-20 11:06:26  teach quality ten education any
49996  2022-10-18 03:57:35            add walk among believe
49997  2020-07-08 03:54:08          onto admit artist first
49998  2022-03-22 12:13:44                             star
49999  2022-12-03 06:11:07                             home

[50000 rows x 11 columns]
```

```python
import pandas as pd
import ast

# Function to convert string representation of lists to actual lists
def convert_hashtags_to_list(hashtags):
    if pd.isna(hashtags):
        return []

    # If already a list, return as is
    if isinstance(hashtags, list):
        return hashtags

    # If empty string, return empty list
    if hashtags == '':
        return []
```

```
    try:
        # Try to evaluate the string as a literal (if it's formatted
like a Python list)
        return ast.literal_eval(hashtags)
    except (ValueError, SyntaxError):
        # If it's a comma-separated string
        if ',' in hashtags:
            return [tag.strip() for tag in hashtags.split(',')]
        # If it's a space-separated string
        else:
            return hashtags.split()

# Apply the function to convert the hashtags column to lists
df['Hashtags'] = df['Hashtags'].apply(convert_hashtags_to_list)

# Display a sample to verify the conversion
print(df['Hashtags'].head())
```

```
0                           []
,1                [both, live]
,2              [phone, ahead]
,3    [ever, quickly, new, I]
,4          [foreign, mention]
,Name: Hashtags, dtype: object
```

```
df
```

```
       User ID         Username  \
0       132131            flong
1       289683   hinesstephanie
2       779715       roberttran
3       696168           pmason
4       704441           noah87
...        ...              ...
49995   491196            uberg
49996   739297     jessicamunoz
49997   674475   lynncunningham
49998   167081  richardthompson
49999   311204         daniel29

                                                   Tweet  Retweet
Count  \
0      Station activity person against natural majori...
85
1      Authority research natural life material staff...
55
2      Manage whose quickly especially foot none to g...
6
3      Just cover eight opportunity strong policy which.
```

```
54
4                          Animal sign six data good or.
26
...                                                  ...             ..
.
49995  Want but put card direction know miss former h...
64
49996  Provide whole maybe agree church respond most ...
18
49997  Bring different everyone international capital...
43
49998  Than about single generation itself seek sell ...
45
49999  Here morning class various room human true bec...
91

       Mention Count  Follower Count  Verified  Bot Label
Location  \
0                  1            2353     False          1
Adkinston
1                  5            9617      True          0
Sanderston
2                  2            4363      True          0
Harrisonfurt
3                  5            2242      True          1
Martinezberg
4                  3            8438     False          1
Camachoville
...              ...             ...       ...        ...
...
49995              0            9911      True          1  Lake
Kimberlyburgh
49996              5            9900     False          1
Greenbury
49997              3            6313      True          1
Deborahfort
49998              1            6343     False          0
Stephenside
49999              4            4006     False          0
Novakberg

                Created At                                  Hashtags
0     2020-05-11 15:29:50                                        []
1     2022-11-26 05:18:10                          [both, live]
2     2022-08-08 03:16:54                        [phone, ahead]
3     2021-08-14 22:27:05                [ever, quickly, new, I]
4     2020-04-13 21:24:21                     [foreign, mention]
...                   ...                                    ...
49995 2023-04-20 11:06:26  [teach, quality, ten, education, any]
```

```
49996 2022-10-18 03:57:35            [add, walk, among, believe]
49997 2020-07-08 03:54:08          [onto, admit, artist, first]
49998 2022-03-22 12:13:44                                [star]
49999 2022-12-03 06:11:07                                [home]

[50000 rows x 11 columns]
```

```python
import pandas as pd
import numpy as np

# Check for missing values in each column
print("Missing values in each column:")
missing_values = df.isna().sum()
print(missing_values)
print("\nPercentage of missing values:")
print((missing_values / len(df)) * 100)

# Decide on strategy for each column based on importance
# For critical columns: User ID, Tweet, Bot Label - drop rows if
missing
critical_columns = ['User ID', 'Tweet', 'Bot Label']
rows_before = len(df)
df = df.dropna(subset=critical_columns)
rows_after = len(df)
print(f"\nDropped {rows_before - rows_after} rows with missing values
in critical columns")

# For location - fill with "Unknown"
if missing_values['Location'] > 0:
    df['Location'] = df['Location'].fillna("Unknown")
    print(f"Filled {missing_values['Location']} missing Location
values with 'Unknown'")

# For Username - fill with "anonymous_user"
if missing_values['Username'] > 0:
    df['Username'] = df['Username'].fillna("anonymous_user")
    print(f"Filled {missing_values['Username']} missing Username
values with 'anonymous_user'")

# For numeric columns - fill with median values
numeric_columns = ['Retweet Count', 'Mention Count', 'Follower Count']
for col in numeric_columns:
    if missing_values[col] > 0:
        median_value = df[col].median()
        df[col] = df[col].fillna(median_value)
        print(f"Filled {missing_values[col]} missing {col} values with
median: {median_value}")
```

```python
# For Hashtags - fill with empty list
if missing_values['Hashtags'] > 0:
    df['Hashtags'] = df['Hashtags'].fillna('[]')
    print(f"Filled {missing_values['Hashtags']} missing Hashtags
values with empty list")

# For Verified - fill with False
if missing_values['Verified'] > 0:
    df['Verified'] = df['Verified'].fillna(False)
    print(f"Filled {missing_values['Verified']} missing Verified
values with False")

# For Created At - fill with median date
if missing_values['Created At'] > 0:
    median_date = df['Created At'].median()
    df['Created At'] = df['Created At'].fillna(median_date)
    print(f"Filled {missing_values['Created At']} missing Created At
values with median date: {median_date}")

# Check if we've handled all missing values
print("\nRemaining missing values:")
print(df.isna().sum())
```

```
Missing values in each column:
,User ID           0
,Username          0
,Tweet             0
,Retweet Count     0
,Mention Count     0
,Follower Count    0
,Verified          0
,Bot Label         0
,Location          0
,Created At        0
,Hashtags          0
,dtype: int64
,
,Percentage of missing values:
,User ID           0.0
,Username          0.0
,Tweet             0.0
,Retweet Count     0.0
,Mention Count     0.0
,Follower Count    0.0
,Verified          0.0
,Bot Label         0.0
,Location          0.0
,Created At        0.0
,Hashtags          0.0
,dtype: float64
```

```
,
,Dropped 0 rows with missing values in critical columns
,
,Remaining missing values:
,User ID           0
,Username          0
,Tweet             0
,Retweet Count     0
,Mention Count     0
,Follower Count    0
,Verified          0
,Bot Label         0
,Location          0
,Created At        0
,Hashtags          0
,dtype: int64
```

This code:

1. First checks and displays the count and percentage of missing values in each column
2. Drops rows with missing values in critical columns (User ID, Tweet, Bot Label)
3. Applies appropriate fill strategies for other columns:
   - Text columns like Location and Username get meaningful text replacements
   - Numeric columns get filled with median values
   - Hashtags get filled with empty lists
   - Boolean 'Verified' column gets filled with False
   - Datetime column gets filled with the median date

This approach balances data preservation with data quality by only dropping rows when critical information is missing and using appropriate fill strategies for other columns.

```
df

       User ID         Username  \
0       132131            flong
1       289683   hinesstephanie
2       779715       roberttran
3       696168           pmason
4       704441           noah87
...        ...              ...
49995   491196            uberg
49996   739297     jessicamunoz
49997   674475   lynncunningham
49998   167081  richardthompson
49999   311204         daniel29

                                                    Tweet  Retweet
Count  \
0      Station activity person against natural majori...
```

```
85
1      Authority research natural life material staff...
55
2      Manage whose quickly especially foot none to g...
6
3      Just cover eight opportunity strong policy which.
54
4                          Animal sign six data good or.
26
...                                                  ...             ..
.
49995  Want but put card direction know miss former h...
64
49996  Provide whole maybe agree church respond most ...
18
49997  Bring different everyone international capital...
43
49998  Than about single generation itself seek sell ...
45
49999  Here morning class various room human true bec...
91

       Mention Count  Follower Count  Verified  Bot Label
Location  \
0                  1            2353     False          1
Adkinston
1                  5            9617      True          0
Sanderston
2                  2            4363      True          0
Harrisonfurt
3                  5            2242      True          1
Martinezberg
4                  3            8438     False          1
Camachoville
...              ...             ...       ...        ...
...
49995              0            9911      True          1  Lake
Kimberlyburgh
49996              5            9900     False          1
Greenbury
49997              3            6313      True          1
Deborahfort
49998              1            6343     False          0
Stephenside
49999              4            4006     False          0
Novakberg

                Created At                                    Hashtags
0     2020-05-11 15:29:50                                          []
```

```
1     2022-11-26 05:18:10                              [both, live]
2     2022-08-08 03:16:54                            [phone, ahead]
3     2021-08-14 22:27:05                  [ever, quickly, new, I]
4     2020-04-13 21:24:21                       [foreign, mention]
...                   ...                                      ...
49995 2023-04-20 11:06:26  [teach, quality, ten, education, any]
49996 2022-10-18 03:57:35             [add, walk, among, believe]
49997 2020-07-08 03:54:08           [onto, admit, artist, first]
49998 2022-03-22 12:13:44                                 [star]
49999 2022-12-03 06:11:07                                 [home]

[50000 rows x 11 columns]
```

```
import pandas as pd
import os

# 1. Export to CSV (most common)
df.to_csv('cleaned_data.csv', index=False)


# Print the current working directory so you know where files are
saved
print(f"Files saved to: {os.getcwd()}")

# Optional: Create a download link for use in Jupyter Notebook
from IPython.display import HTML, display

def create_download_link(df, filename, text):
    csv = df.to_csv(index=False)
    b64 = base64.b64encode(csv.encode())
    payload = b64.decode()
    html = f'<a download="{filename}" href="data:text/csv;base64,
{payload}" target="_blank">{text}</a>'
    return HTML(html)


# display(create_download_link(cleaned_data, 'cleaned_data.csv',
'Download CSV'))
```

```
Files saved to: C:\Users\scott
```

```
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.preprocessing import MinMaxScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, f1_score,
classification_report, confusion_matrix

# the cleaned dataset
df = pd.read_csv("cleaned_data.csv")

# previewing data
print("Dataset shape:", df.shape)
print("First few rows:\n", df.head())

# we drop any identifier columns that won't help prediction
df = df.drop(columns=['id', 'user_handle', 'tweet_text'],
errors='ignore')



#non_numeric to drop columns that arent numeric before training
non_numeric = ['Username', 'Tweet', 'Location', 'Created At',
'Hashtags']
X = df.drop(columns=non_numeric + ['Bot Label'], errors='ignore')
y = df['Bot Label']

# Normalize numerical features
scaler = MinMaxScaler()
X_scaled = pd.DataFrame(scaler.fit_transform(X), columns=X.columns)

# Train-test split (80/20)
X_train, X_test, y_train, y_test = train_test_split(
    X_scaled, y, test_size=0.2, random_state=42, stratify=y
)

# Initialize and train logistic regression model
log_model = LogisticRegression(max_iter=1000, random_state=42)
log_model.fit(X_train, y_train)

# Predict on test set
y_pred = log_model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Logistic Regression Accuracy: {:.2f}%".format(accuracy * 100))
print("F1 Score: {:.2f}".format(f1))
print("\nClassification Report:\n", classification_report(y_test,
y_pred))
```

```
# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```