```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

corona = pd.read_csv("Corona_tested_006.csv")
```

```
C:\Users\satya\AppData\Local\Temp\ipykernel_23412\709195987.py:1:
DtypeWarning: Columns (2,3,4,5,6) have mixed types. Specify dtype
option on import or set low_memory=False.
  corona = pd.read_csv("Corona_tested_006.csv")
```

```
corona.head()
```

```
   Ind_ID   Test_date Cough_symptoms Fever Sore_throat
Shortness_of_breath  \
0       1  11-03-2020              1     0           1
0
1       2  11-03-2020              0     1           0
0
2       3  11-03-2020              0     1           0
0
3       4  11-03-2020              1     0           0
0
4       5  11-03-2020              1     0           0
0

  Headache    Corona Age_60_above  Sex           Known_contact
0        0  Negative          NAN  NAN                  Abroad
1        0  Positive          NAN  NAN                  Abroad
2        0  Positive          NAN  NAN                  Abroad
3        0  Negative          NAN  NAN                  Abroad
4        0  Negative          NAN  NAN  Contact with confirmed
```

```
corona.dtypes
```

```
Ind_ID                 int64
Test_date             object
Cough_symptoms        object
Fever                 object
Sore_throat           object
Shortness_of_breath   object
Headache              object
Corona                object
Age_60_above          object
Sex                   object
Known_contact         object
dtype: object
```

```
corona = corona.replace(r'^\s*$', np.nan, regex=True)
```

```
from datetime import datetime

date_string = "2020-03-11"
date_object = datetime.strptime(date_string, "%Y-%m-%d")

corona.head()
```

```
   Ind_ID   Test_date Cough_symptoms Fever Sore_throat
Shortness_of_breath  \
0       1  11-03-2020              1     0           1
0
1       2  11-03-2020              0     1           0
0
2       3  11-03-2020              0     1           0
0
3       4  11-03-2020              1     0           0
0
4       5  11-03-2020              1     0           0
0

  Headache    Corona Age_60_above  Sex           Known_contact
0        0  Negative          NAN  NAN                  Abroad
1        0  Positive          NAN  NAN                  Abroad
2        0  Positive          NAN  NAN                  Abroad
3        0  Negative          NAN  NAN                  Abroad
4        0  Negative          NAN  NAN  Contact with confirmed
```

```
corona.dtypes
```

```
Ind_ID                  int64
Test_date              object
Cough_symptoms         object
Fever                  object
Sore_throat            object
Shortness_of_breath    object
Headache               object
Corona                 object
Age_60_above           object
Sex                    object
Known_contact          object
dtype: object
```

```
# Check for missing values
print(corona.isnull().sum())
```

```
Ind_ID                  0
Test_date               0
Cough_symptoms          0
Fever                   0
Sore_throat             0
Shortness_of_breath     0
```

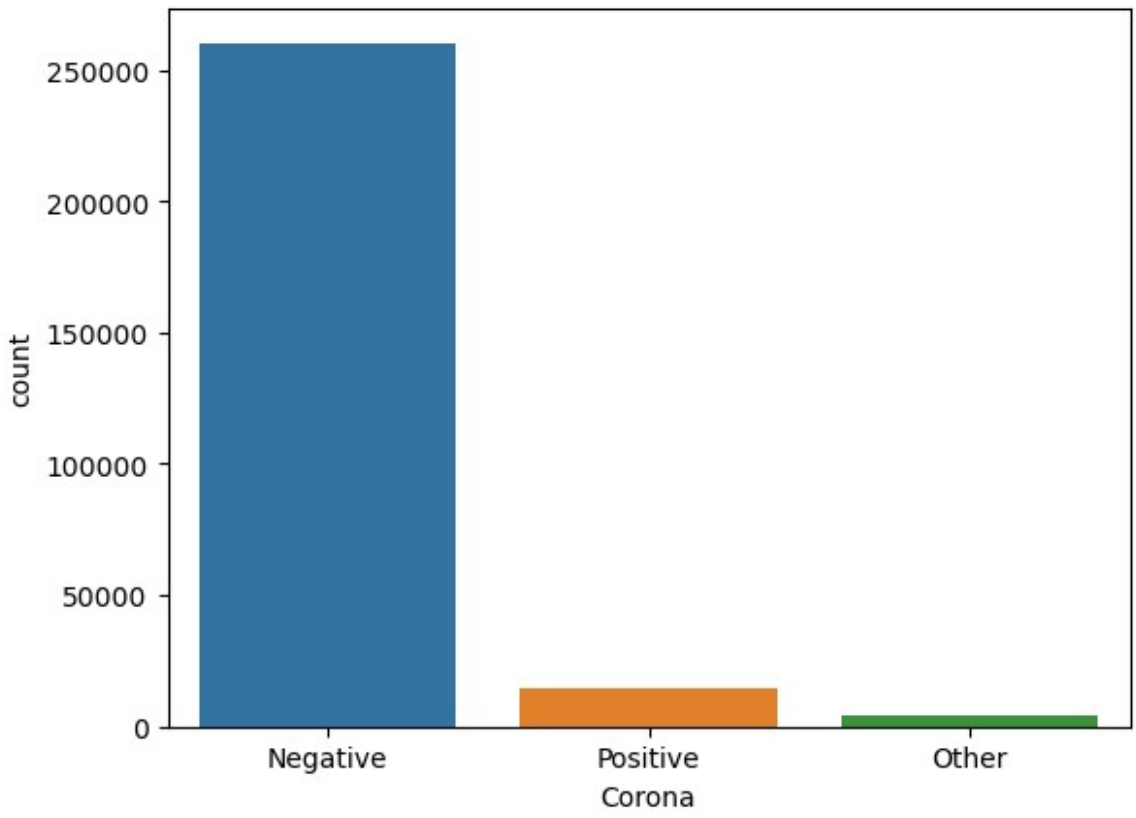
```
Headache                0
Corona                  0
Age_60_above            0
Sex                     0
Known_contact           0
dtype: int64
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Descriptive statistics
print(corona.describe())

# Visualizations
sns.countplot(x='Corona', data=corona)
plt.show()
# Continue with other visualizations as needed
```

```
              Ind_ID
count  278848.000000
mean   139424.500000
std     80496.628269
min         1.000000
25%     69712.750000
50%    139424.500000
75%    209136.250000
max    278848.000000
```




```
# Create dummy variables for categorical features
corona = pd.get_dummies(corona, columns=['Sex', 'Known_contact'],
drop_first=True)

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load your dataset
df = pd.read_csv("corona_tested_006.csv")

# Drop non-numeric columns (for simplicity in this example)
X = df.drop(['Corona','Sex', 'Known_contact'], axis=1)
y = df['Corona']

# Convert categorical variables to numerical (you may need more
advanced preprocessing)
X = pd.get_dummies(X)

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create and train the model
```

```
model_dt = DecisionTreeClassifier()
model_dt.fit(X_train, y_train)

# Make predictions
y_pred = model_dt.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
C:\Users\satya\AppData\Local\Temp\ipykernel_23412\3189365186.py:7:
DtypeWarning: Columns (2,3,4,5,6) have mixed types. Specify dtype
option on import or set low_memory=False.
  df = pd.read_csv("corona_tested_006.csv")

Accuracy: 0.92
Classification Report:
              precision    recall  f1-score   support

    Negative       0.96      0.96      0.96     52041
       Other       0.26      0.24      0.25       791
    Positive       0.46      0.45      0.46      2938

    accuracy                           0.92     55770
   macro avg       0.56      0.55      0.55     55770
weighted avg       0.92      0.92      0.92     55770
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load your dataset
df = pd.read_csv("Corona_tested_006.csv")

# Assuming 'Corona' is the target variable
X = df.drop(['Corona','Sex','Known_contact'], axis=1)
y = df['Corona']

# Convert categorical variables to numerical
X = pd.get_dummies(X)

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

```python
# Create and train the model
model_rf = RandomForestClassifier()
model_rf.fit(X_train, y_train)

# Make predictions
y_pred = model_rf.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")

# Classification report
print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
C:\Users\satya\AppData\Local\Temp\ipykernel_23412\4176993577.py:7:
DtypeWarning: Columns (2,3,4,5,6) have mixed types. Specify dtype
option on import or set low_memory=False.
  df = pd.read_csv("Corona_tested_006.csv")

Accuracy: 0.92
Classification Report:
              precision    recall  f1-score   support

    Negative       0.96      0.96      0.96     52041
       Other       0.25      0.24      0.25       791
    Positive       0.46      0.45      0.46      2938

    accuracy                           0.92     55770
   macro avg       0.56      0.55      0.55     55770
weighted avg       0.92      0.92      0.92     55770
```

```python
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

model_lr = LogisticRegression()
model_lr.fit(X_train, y_train)

y_pred_lr = model_lr.predict(X_test)
accuracy_lr = accuracy_score(y_test, y_pred_lr)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
```

```
Accuracy: 0.92
```

```
import duckdb
conn=duckdb.connect()

conn.register("df",df)
```

```
<duckdb.duckdb.DuckDBPyConnection at 0x28b5ca0e870>
```

```
conn.execute("select * from df").fetchdf()
```

```
        Ind_ID   Test_date Cough_symptoms Fever Sore_throat  \
0            1  11-03-2020              1     0           1
1            2  11-03-2020              0     1           0
2            3  11-03-2020              0     1           0
3            4  11-03-2020              1     0           0
4            5  11-03-2020              1     0           0
...        ...         ...            ...   ...         ...
278843  278844  30-04-2020              0     0           0
278844  278845  30-04-2020              0     0           0
278845  278846  30-04-2020              0     0           0
278846  278847  30-04-2020              0     0           0
278847  278848  30-04-2020              0     0           0

       Shortness_of_breath Headache    Corona Age_60_above     Sex  \
0                        0        0  Negative          NAN     NAN
1                        0        0  Positive          NAN     NAN
2                        0        0  Positive          NAN     NAN
3                        0        0  Negative          NAN     NAN
4                        0        0  Negative          NAN     NAN
...                    ...      ...       ...          ...     ...
278843                   0        0  Positive          NAN    male
278844                   0        0  Negative          NAN  female
278845                   0        0  Negative          NAN    male
278846                   0        0  Negative          NAN    male
278847                   0        0  Negative          NAN  female

                  Known_contact
0                        Abroad
1                        Abroad
2                        Abroad
3                        Abroad
4       Contact with confirmed
...                        ...
278843                   Other
278844                   Other
278845                   Other
278846                   Other
278847                   Other

[278848 rows x 11 columns]
```

The model analysis is made and the data has Accuracy: 0.92. we have applied the the decision tree and random forest and linear regression algothrim to get the details of the data and also the is cleaned to process.

```
#Find the number of corona patients who faced shortness of breath.
# Assuming df is your DataFrame
num_patients_shortness_of_breath = df[((df['Corona'] == 'Negative') |
(df['Corona'] == 'Positive') | (df['Corona'] == 'other')) &
(df['Shortness_of_breath'] == 1)].shape[0]

print("Number of patients with Shortness of Breath for corona:",
num_patients_shortness_of_breath)
```

```
Number of patients with Shortness of Breath for corona: 465
```

```
# Find the number of negative corona patients who have fever and
sore_throat.
# Assuming df is your DataFrame
num_patients_fever_sore_throat= df[(df['Corona'] == 'Negative') &
(df['Sore_throat'] == 1) & (df['Fever'] == 1)].shape[0]

print("Number of patients with fever and sore throat and negative for
corona:", num_patients_fever_sore_throat)
```

```
Number of patients with fever and sore throat and negative for corona:
14
```

```
#Find the female negative corona patients who faced cough and
headache:
# Assuming df is your DataFrame
num_negative_cough_symptoms_headache = df[(df['Corona'] == 'Negative')
& (df['Sex'] == 'female') & (df['Cough_symptoms'] == 1) &
(df['Headache'] == 1)].shape[0]

print("Number of female patients with cough and headache and not
'Positive' for corona:", num_negative_cough_symptoms_headache)
```

```
Number of female patients with cough and headache and not 'Positive'
for corona: 8
```

```
#How many elderly corona patients have faced breathing problems?
# Assuming df is your DataFrame
num_elderly_shortness_of_breath = df[(df['Corona'] != 'Negative') &
(df['Age_60_above'] == 'Yes') & (df['Shortness_of_breath'] ==
1) ].shape[0]
```

```
print("Number of elderly patients with shortness of breath and not
'Negative' for corona:", num_elderly_shortness_of_breath)
```

```
Number of elderly patients with shortness of breath and not 'Negative'
for corona: 95
```

```
# Convert symptom columns to boolean
df[['Fever', 'Cough_symptoms', 'Shortness_of_breath', 'Headache',
'Sore_throat']] = df[['Fever', 'Cough_symptoms',
'Shortness_of_breath', 'Headache', 'Sore_throat']].astype(float)

# Map 'Positive' and 'Negative' to 1 and 0 in the 'Corona' column
df['Corona'] = df['Corona'].map({'Positive': 1, 'Negative': 0})

# Assuming df is your DataFrame
covid_positive_df = df[df['Corona'] == 1]

# Print the counts of each symptom to investigate
print("Counts of each symptom among COVID-positive patients:")
print(covid_positive_df[['Fever', 'Cough_symptoms',
'Shortness_of_breath', 'Headache', 'Sore_throat']].sum())

# Get the top three symptoms
top_three_symptoms = covid_positive_df[['Fever', 'Cough_symptoms',
'Shortness_of_breath', 'Headache', 'Sore_throat']].sum().nlargest(3)

print("Top three symptoms among COVID-positive patients:")
print(top_three_symptoms)
```

```
Counts of each symptom among COVID-positive patients:
Fever                  11399.0
Cough_symptoms         11990.0
Shortness_of_breath     6085.0
Headache                6592.0
Sore_throat             6313.0
dtype: float64
Top three symptoms among COVID-positive patients:
Cough_symptoms    11990.0
Fever             11399.0
Headache           6592.0
dtype: float64
```

```
# Which symptom was less common among COVID negative people?
# Assuming df is your DataFrame
symptoms = ['Fever', 'Cough_symptoms', 'Shortness_of_breath',
'Headache', 'Sore_throat']

# Filter COVID-negative cases
covid_negative_df = df[df['Corona'] == 0]
```

```
# Calculate the frequencies of each symptom
symptom_frequencies = covid_negative_df[symptoms].sum()

# Find the symptom with the lowest frequency
less_common_symptom = symptom_frequencies.idxmin()

print(f"The symptom less common among COVID-negative people is:
{less_common_symptom}")
```

```
The symptom less common among COVID-negative people is:
Shortness_of_breath
```

```
# What are the most common symptoms among COVID positive males whose
known contact was abroad?
# Assuming df is your DataFrame
symptoms = ['Fever', 'Cough_symptoms', 'Shortness_of_breath',
'Headache', 'Sore_throat']

# Filter COVID-positive males with known contact abroad
covid_positive_male_abroad_df = df[(df['Corona'] == 1) & (df['Sex'] ==
'Male') & (df['Known_contact'] == 'Abroad')]

# Calculate the frequencies of each symptom
symptom_frequencies = covid_positive_male_abroad_df[symptoms].sum()

# Find the top three symptoms
top_three_symptoms = symptom_frequencies.nlargest(3)

print("The most common symptoms among COVID-positive males with known
contact abroad:")
print(top_three_symptoms)
```

```
The most common symptoms among COVID-positive males with known contact
abroad:
Fever                 0.0
Cough_symptoms        0.0
Shortness_of_breath   0.0
dtype: float64
```