```
import os
import pandas as pd

# Specify the directory containing the txt files
directory = 'DATASETS/Stock_datasets/Data/Stocks'  # Replace with the
path to your folder

# Initialize an empty list to hold individual DataFrames
dataframes = []

# Initialize a counter for files processed
file_count = 0

# Iterate over each file in the directory
for filename in os.listdir(directory):
    if file_count >= 10:  # Stop after processing 10 files
        break

    if filename.endswith('.txt'):
        # Create the full file path
        file_path = os.path.join(directory, filename)

        # Extract the base name and country code (e.g., 'aaap' and
'us' from 'aaap.us.txt')
        base_name, country = filename.split('.')[0],
filename.split('.')[1]

        try:
            # Read the txt file as a CSV into a DataFrame
            df = pd.read_csv(file_path, delimiter=',')

            # Check if the DataFrame has data
            if df.empty:
                print(f"Skipping empty file: {filename}")
                continue

            # Add new columns with stock name and country
            df['Stock_Name'] = base_name
            df['Country'] = country

            # Append the DataFrame to the list
            dataframes.append(df)
            print(f"Successfully loaded file: {filename}")
            file_count += 1  # Increment the counter

        except pd.errors.EmptyDataError:
            print(f"Skipping empty or malformed file: {filename}")
        except pd.errors.ParserError:
            print(f"Skipping file with parsing error: {filename}")
    else:
```

```
        print(f"Skipping non-txt file: {filename}")

# Concatenate all DataFrames in the list into a single DataFrame, if
there are any
if dataframes:
    all_data = pd.concat(dataframes, ignore_index=True)
    print("Combined DataFrame with 'Stock_Name' and 'Country'
columns:")
    print(all_data)
else:
    print("No valid data files found in the directory.")
```

```
Successfully loaded file: a.us.txt
Successfully loaded file: aa.us.txt
Successfully loaded file: aaap.us.txt
Successfully loaded file: aaba.us.txt
Successfully loaded file: aac.us.txt
Successfully loaded file: aal.us.txt
Successfully loaded file: aamc.us.txt
Successfully loaded file: aame.us.txt
Successfully loaded file: aan.us.txt
Successfully loaded file: aaoi.us.txt
Combined DataFrame with 'Stock_Name' and 'Country' columns:
             Date    Open     High      Low   Close    Volume  OpenInt
\
0      1999-11-18  30.713  33.7540  27.0020  29.702  66277506        0

1      1999-11-19  28.986  29.0270  26.8720  27.257  16142920        0

2      1999-11-22  27.886  29.7020  27.0440  29.702   6970266        0

3      1999-11-23  28.688  29.4460  27.0020  27.002   6332082        0

4      1999-11-24  27.083  28.3090  27.0020  27.717   5132147        0

...           ...     ...      ...      ...     ...       ...      ...

32682  2017-11-06  37.580  38.7200  36.7000  37.800   2388819        0

32683  2017-11-07  37.980  38.0750  36.9700  37.890   3408164        0

32684  2017-11-08  40.470  44.6432  39.1000  43.640   7669971        0

32685  2017-11-09  43.000  44.2400  41.3441  43.200   2867311        0

32686  2017-11-10  42.670  46.0717  42.5500  45.300   2967517        0


      Stock_Name Country
0              a      us
1              a      us
```

```
2              a      us
3              a      us
4              a      us
...          ...     ...
32682       aaoi      us
32683       aaoi      us
32684       aaoi      us
32685       aaoi      us
32686       aaoi      us

[32687 rows x 9 columns]
```

```
# all_data.groupby('Country').head()
df = all_data.copy()
df.head()
```

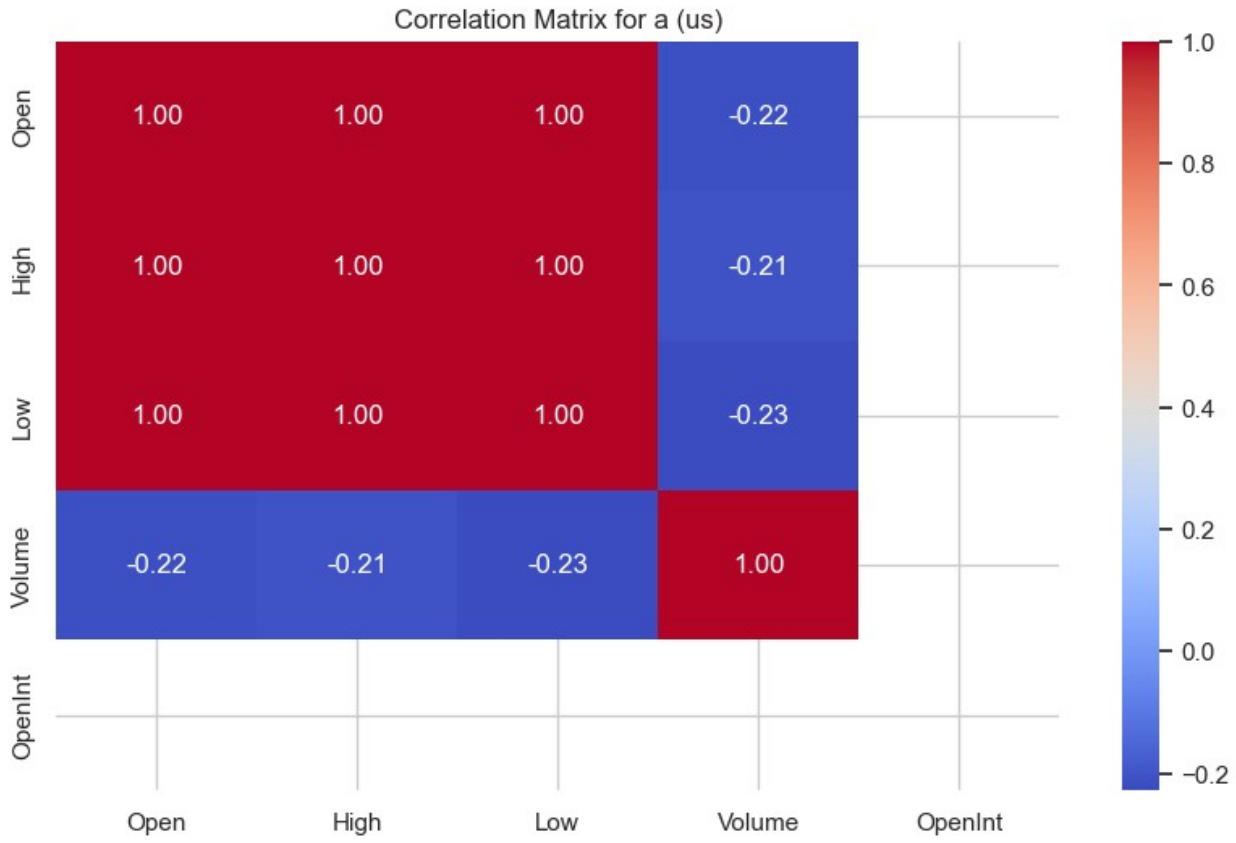
```
         Date    Open    High     Low   Close    Volume  OpenInt
Stock_Name  \
0  1999-11-18  30.713  33.754  27.002  29.702  66277506        0
a
1  1999-11-19  28.986  29.027  26.872  27.257  16142920        0
a
2  1999-11-22  27.886  29.702  27.044  29.702   6970266        0
a
3  1999-11-23  28.688  29.446  27.002  27.002   6332082        0
a
4  1999-11-24  27.083  28.309  27.002  27.717   5132147        0
a

  Country
0      us
1      us
2      us
3      us
4      us
```

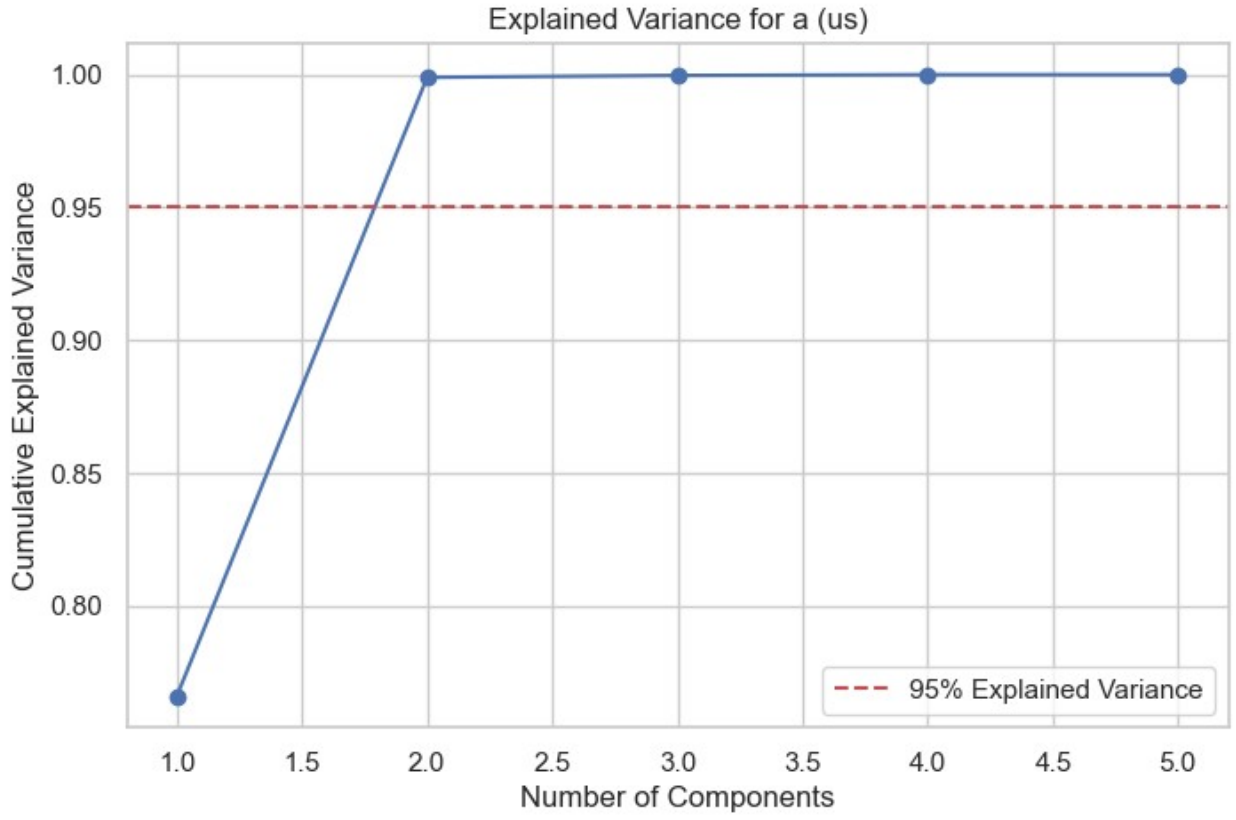
```
import os
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.decomposition import PCA
from sklearn.feature_selection import VarianceThreshold
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
```

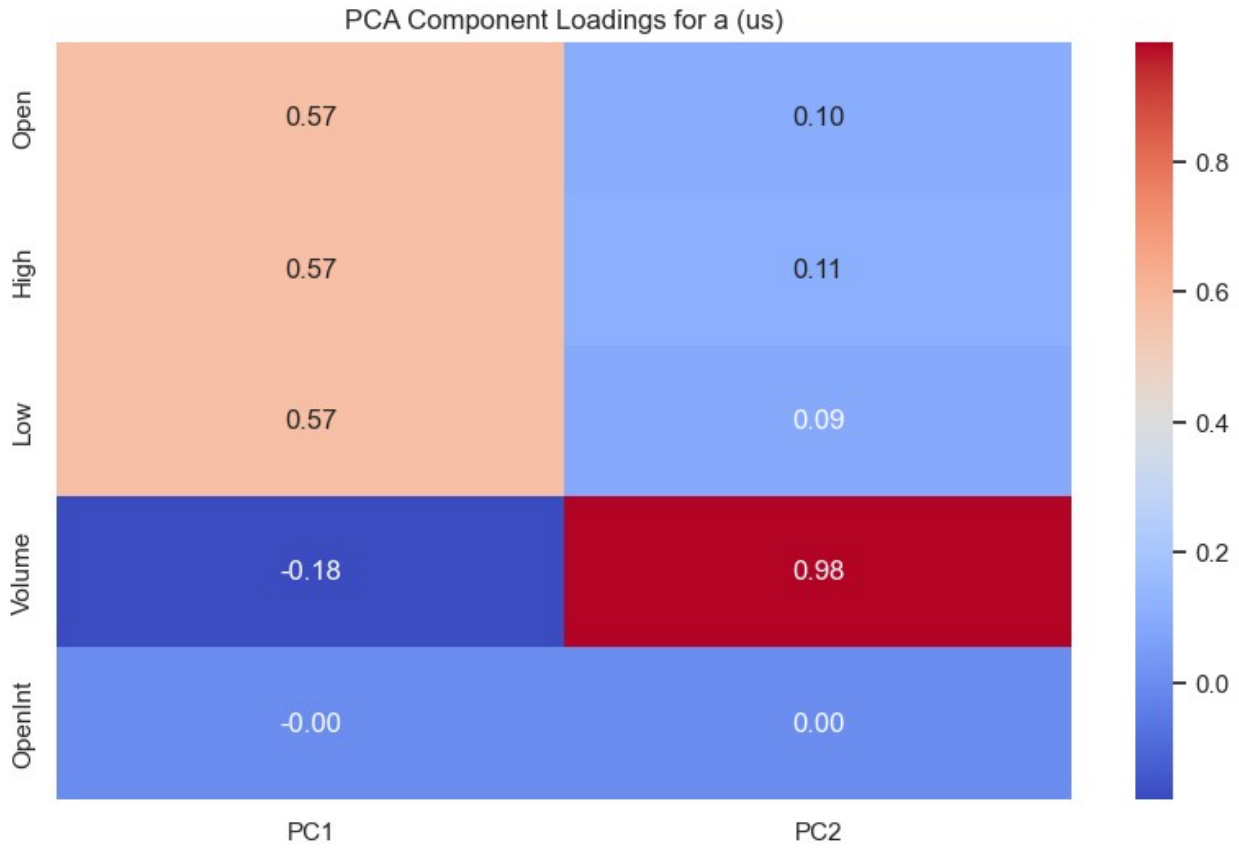
```
# Drop irrelevant columns
df = df.drop(columns=['Date'])

# Dictionary to store results
results = {}

# Set up Seaborn styling
sns.set(style="whitegrid")

# Iterate through each stock-country group
for (stock, country), group in df.groupby(['Stock_Name', 'Country']):
    print(f"\nProcessing Stock: {stock}, Country: {country}")

    # Separate features and target
    X = group.drop(columns=['Close', 'Stock_Name', 'Country'])  #
Predicting 'Close' price
    y = group['Close']

    # Data Preparation: Standardize features
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Visualize correlation matrix with heatmap
    plt.figure(figsize=(10, 6))
    sns.heatmap(pd.DataFrame(X, columns=X.columns).corr(), annot=True,
cmap='coolwarm', fmt=".2f")
    plt.title(f"Correlation Matrix for {stock} ({country})")
    plt.show()

    # Feature Selection Technique 1: Low Variance Filter
    var_thresh = VarianceThreshold(threshold=0.1)
    X_low_variance = var_thresh.fit_transform(X_scaled)

    # Feature Selection Technique 2: PCA
    pca = PCA()
    pca.fit(X_scaled)

    # Calculate cumulative explained variance
    explained_variance = np.cumsum(pca.explained_variance_ratio_)
    n_components = np.argmax(explained_variance >= 0.95) + 1  # 95%
variance

    # Apply PCA with selected components
    pca = PCA(n_components=n_components)
    X_pca = pca.fit_transform(X_scaled)

    # Plot cumulative explained variance
    plt.figure(figsize=(8, 5))
    plt.plot(range(1, len(explained_variance) + 1),
explained_variance, marker='o')
```

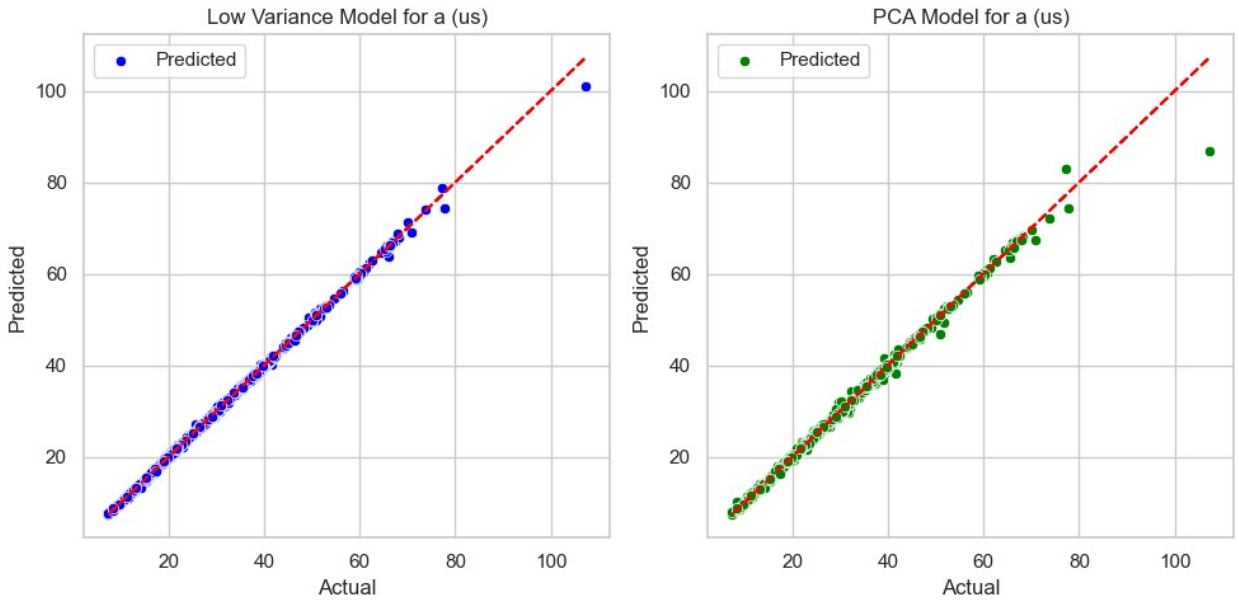
```
    plt.xlabel('Number of Components')
    plt.ylabel('Cumulative Explained Variance')
    plt.title(f'Explained Variance for {stock} ({country})')
    plt.axhline(y=0.95, color='r', linestyle='--', label='95%
Explained Variance')
    plt.legend()
    plt.show()

    # PCA component loadings (contribution of each feature to each
component)
    loading_matrix = pd.DataFrame(pca.components_.T, index=X.columns,
columns=[f'PC{i+1}' for i in range(n_components)])
    plt.figure(figsize=(10, 6))
    sns.heatmap(loading_matrix, annot=True, cmap="coolwarm",
fmt=".2f")
    plt.title(f"PCA Component Loadings for {stock} ({country})")
    plt.show()

    # Model Training and Evaluation for each technique
    # Split data for each feature set
    X_train_lv, X_test_lv, y_train, y_test =
train_test_split(X_low_variance, y, test_size=0.2, random_state=42)
    X_train_pca, X_test_pca, _, _ = train_test_split(X_pca, y,
test_size=0.2, random_state=42)

    # Train models
    model_lv = LinearRegression().fit(X_train_lv, y_train)
    model_pca = LinearRegression().fit(X_train_pca, y_train)

    # Predictions
    y_pred_lv = model_lv.predict(X_test_lv)
    y_pred_pca = model_pca.predict(X_test_pca)

    # Calculate Mean Squared Error
    mse_lv = mean_squared_error(y_test, y_pred_lv)
    mse_pca = mean_squared_error(y_test, y_pred_pca)

    # Scatter plot of predicted vs actual for both models
    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    sns.scatterplot(x=y_test, y=y_pred_lv, color='blue',
label='Predicted')
    plt.plot(y_test, y_test, color='red', linestyle='--')
    plt.title(f"Low Variance Model for {stock} ({country})")
    plt.xlabel("Actual")
    plt.ylabel("Predicted")

    plt.subplot(1, 2, 2)
    sns.scatterplot(x=y_test, y=y_pred_pca, color='green',
```

```
label='Predicted')
    plt.plot(y_test, y_test, color='red', linestyle='--')
    plt.title(f"PCA Model for {stock} ({country})")
    plt.xlabel("Actual")
    plt.ylabel("Predicted")
    plt.tight_layout()
    plt.show()

    # Store results
    results[(stock, country)] = {
        'Low Variance Filter MSE': mse_lv,
        'PCA MSE': mse_pca,
        'PCA Components': n_components,
    }
    print(f"Results for {stock} ({country}):")
    print(f"  Low Variance Filter MSE: {mse_lv}")
    print(f"  PCA MSE: {mse_pca}")

# Display summarized results
print("\nSummary of results for each Stock-Country combination:")
for (stock, country), metrics in results.items():
    print(f"{stock} ({country}) - LV MSE: {metrics['Low Variance
Filter MSE']}, PCA MSE: {metrics['PCA MSE']}, PCA Components:
{metrics['PCA Components']}")


Processing Stock: a, Country: us
```

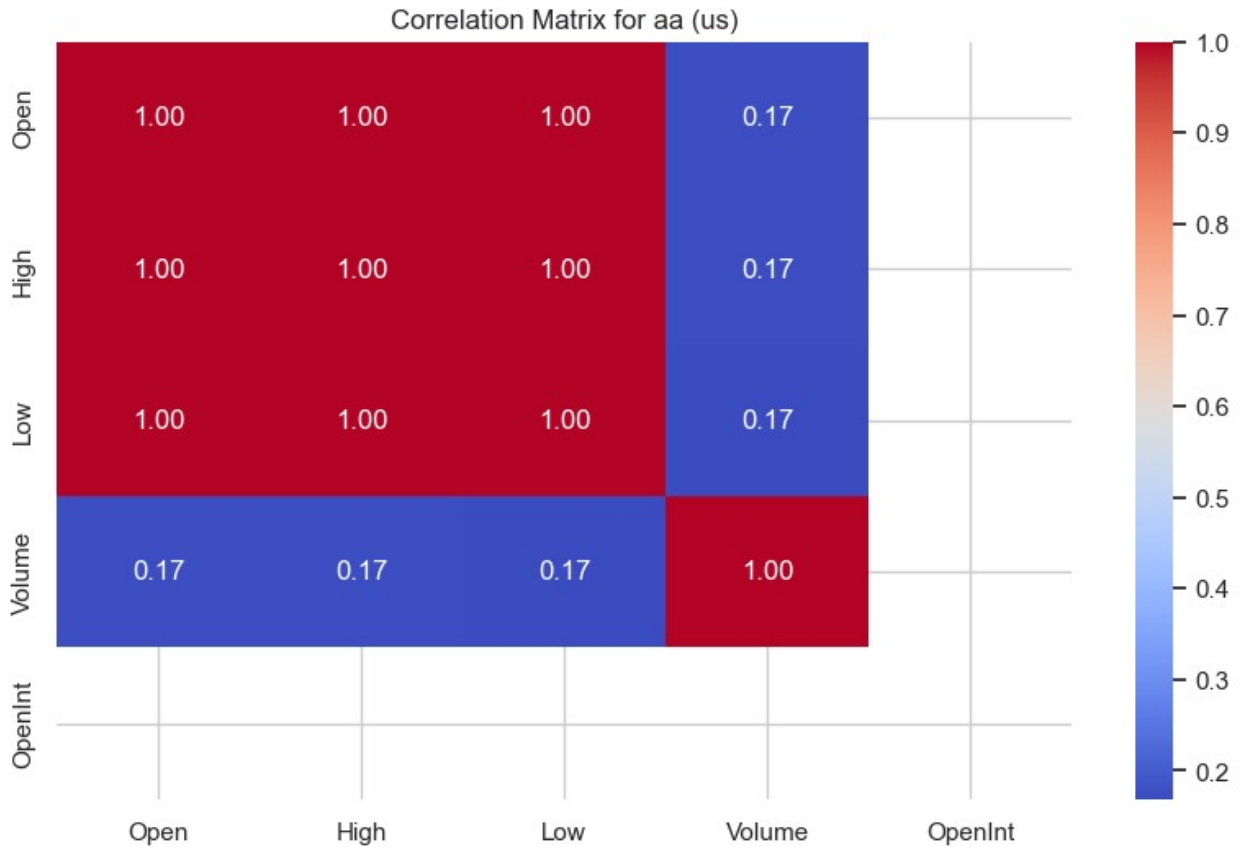

Correlation Matrix for a (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
-0.22
1.00
1.00
1.00
-0.21
1.00
1.00
1.00
-0.23
-0.22
-0.21
-0.23
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

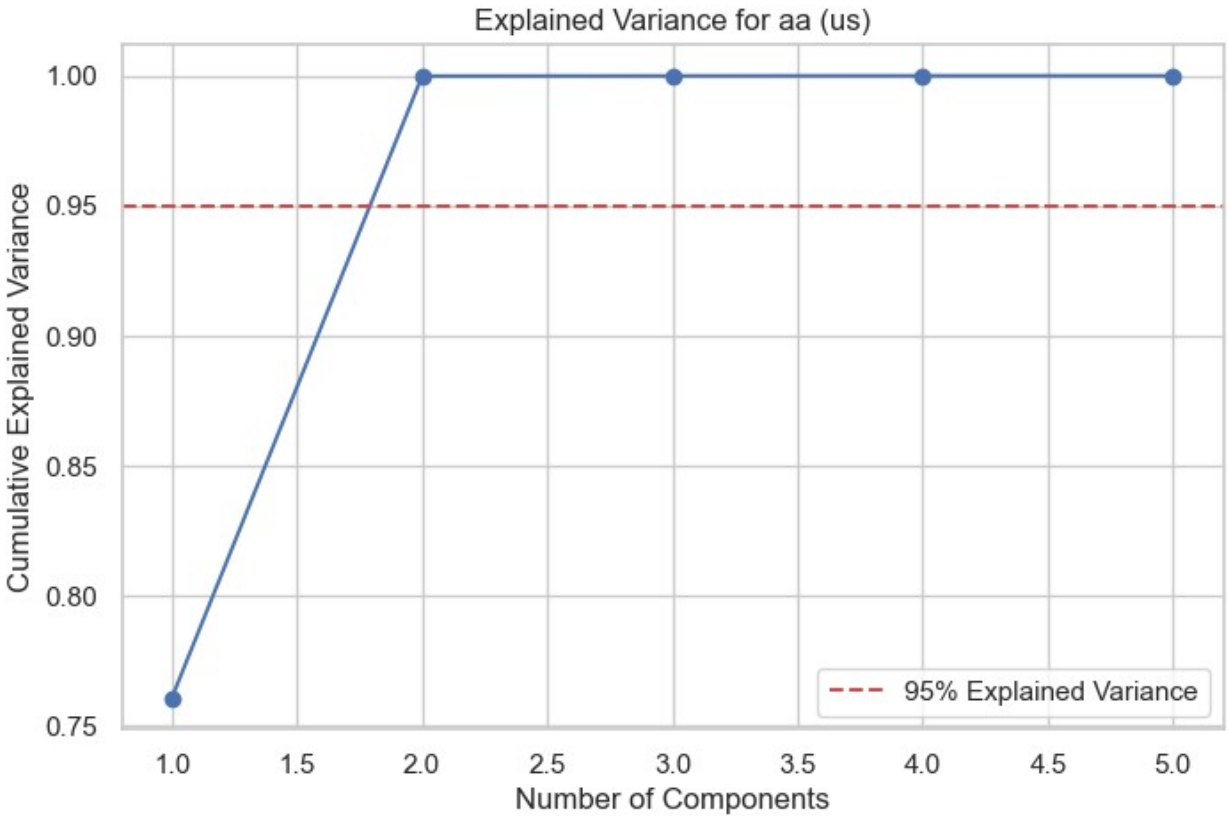

Explained Variance for a (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

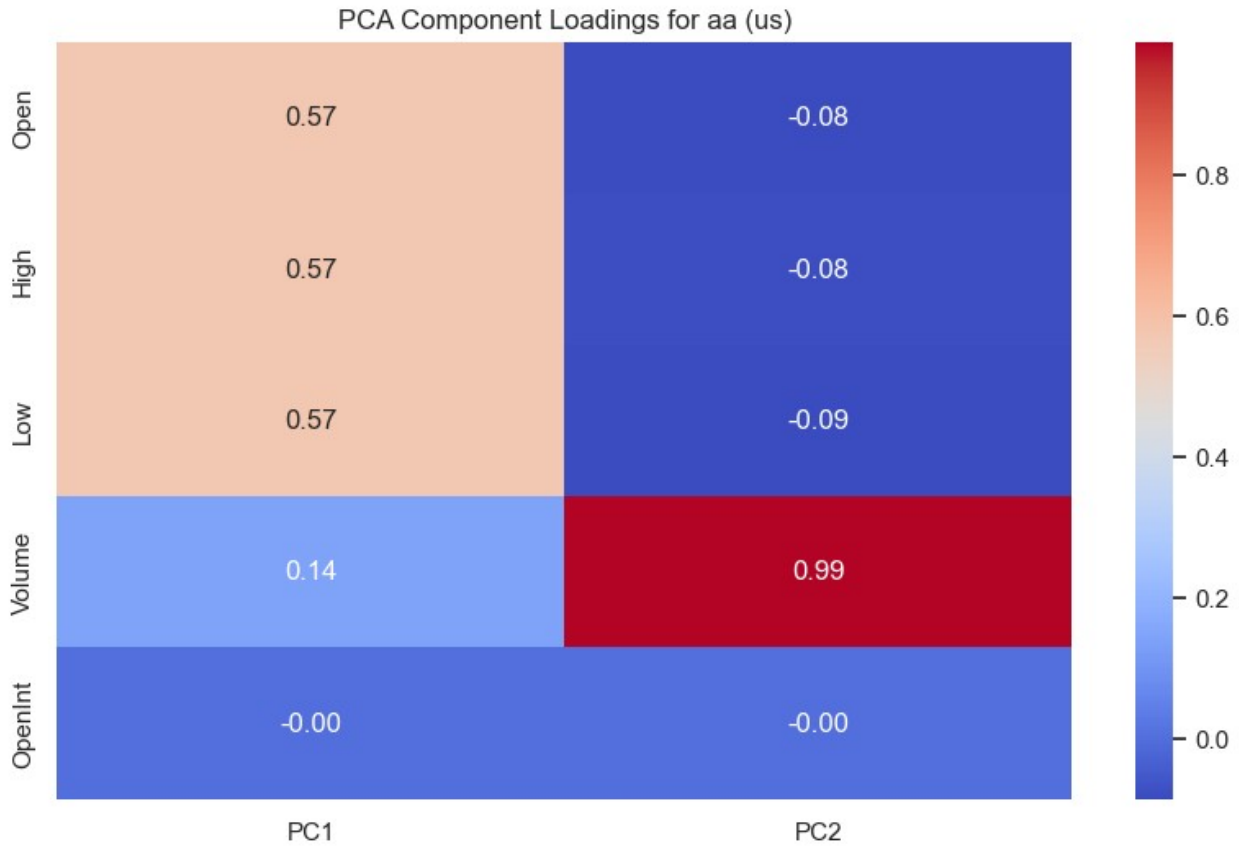


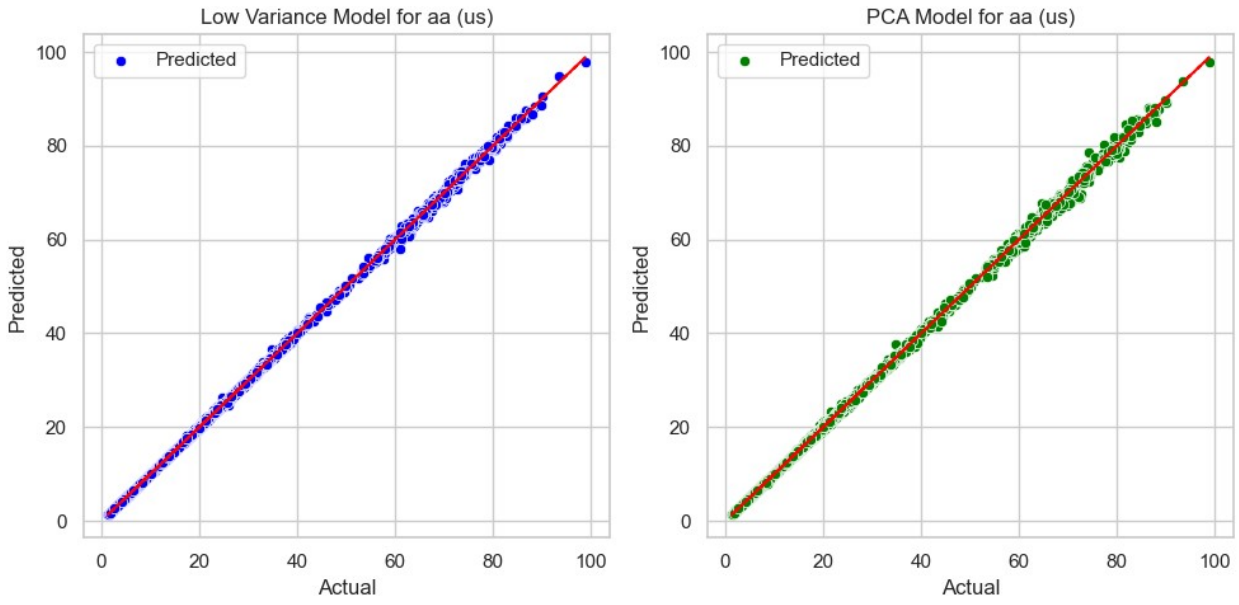



```
Results for a (us):
  Low Variance Filter MSE: 0.1286081944181578
  PCA MSE: 0.7218900236595417

Processing Stock: aa, Country: us
```

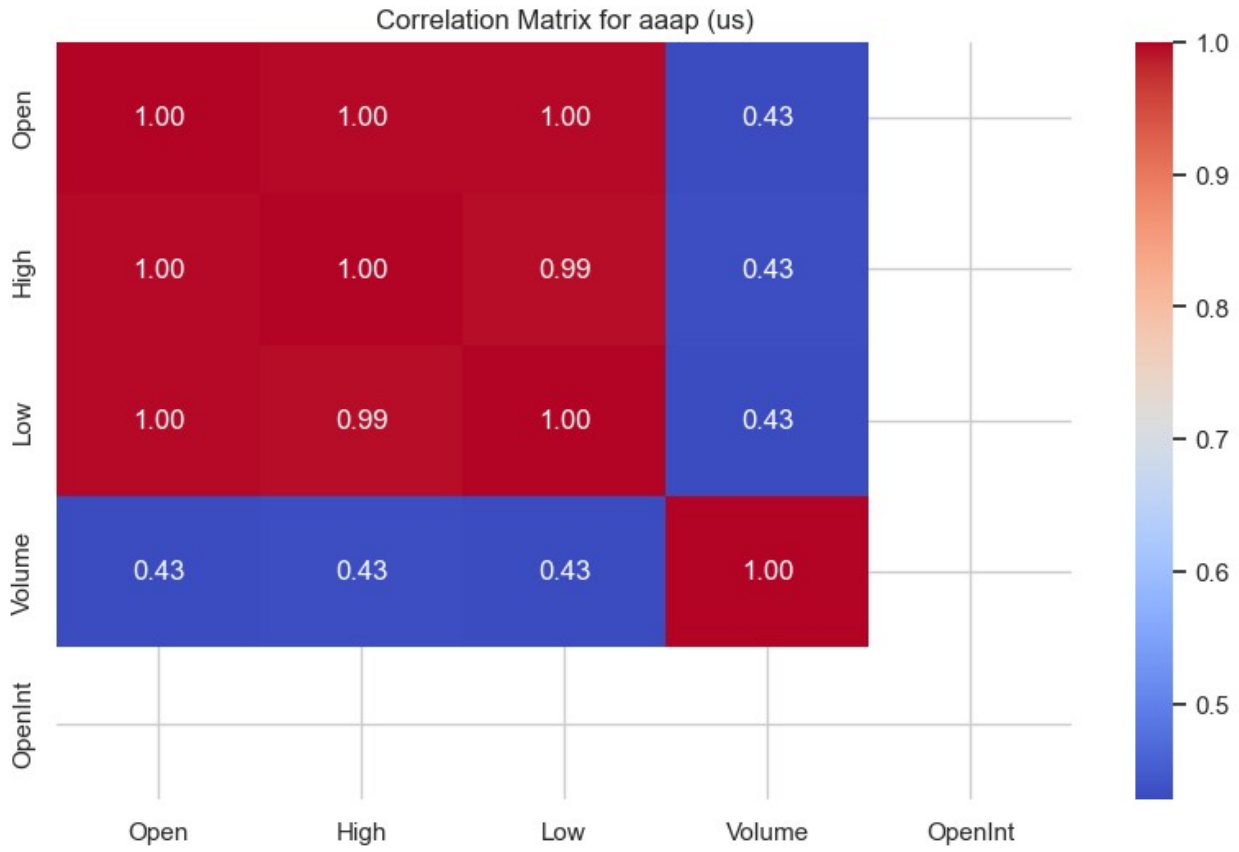

Correlation Matrix for aa (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
0.17
1.00
1.00
1.00
0.17
1.00
1.00
1.00
0.17
0.17
0.17
0.17
1.00
Open
High
Low
Volume
OpenInt
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2

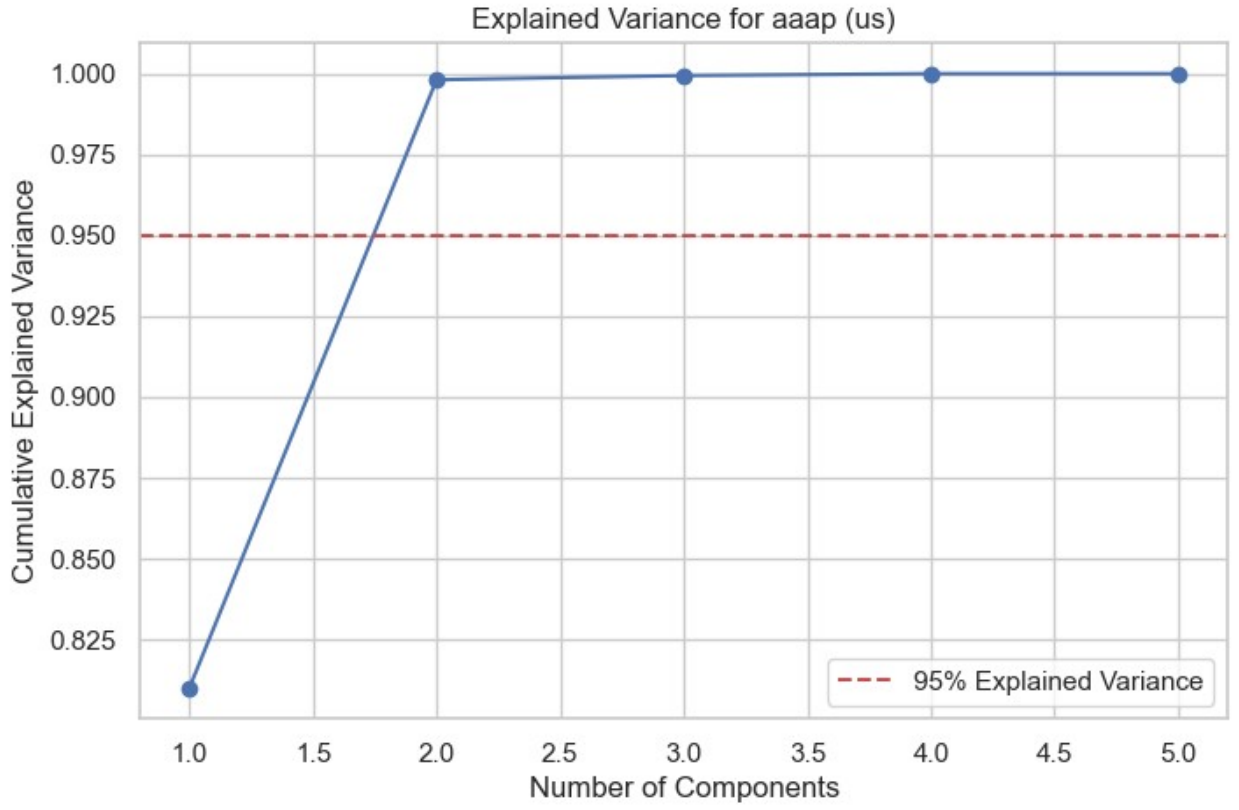

Explained Variance for aa (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components
95% Explained Variance

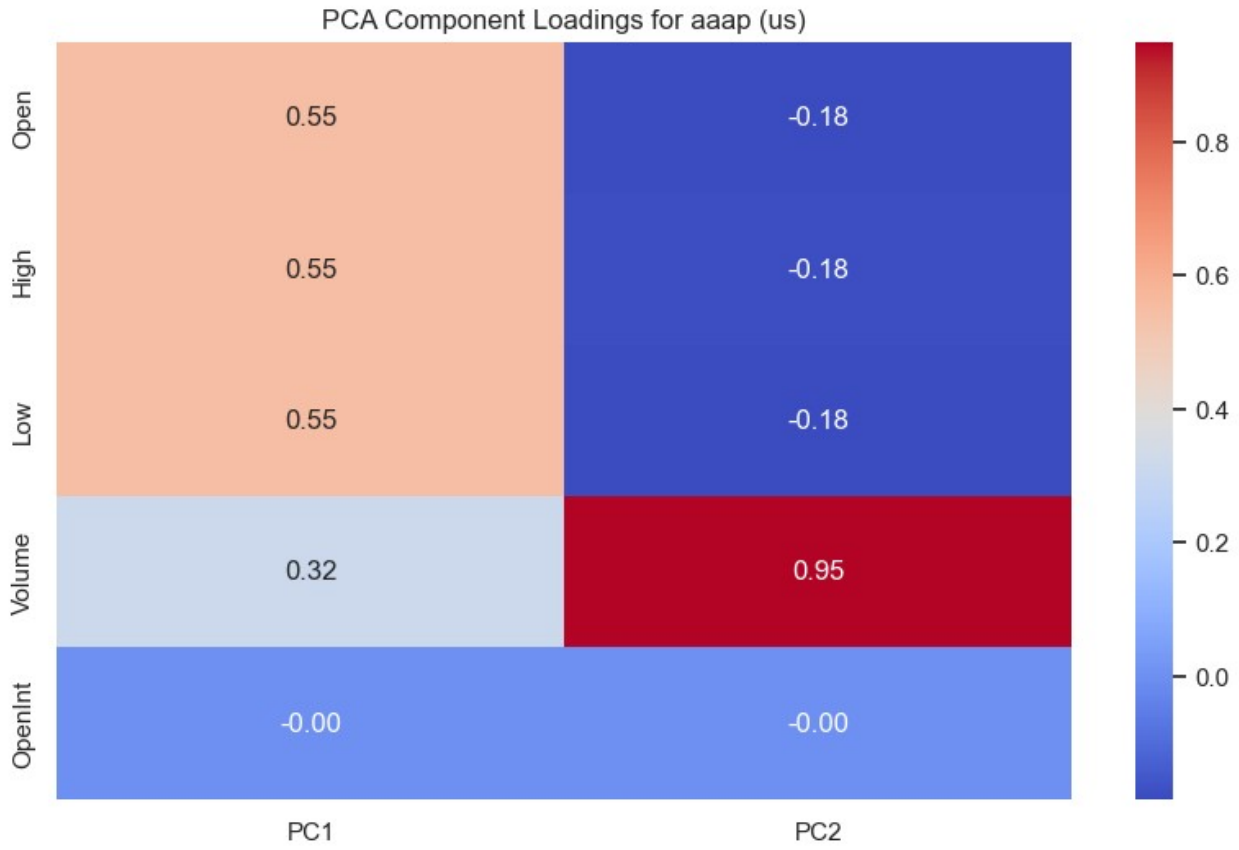


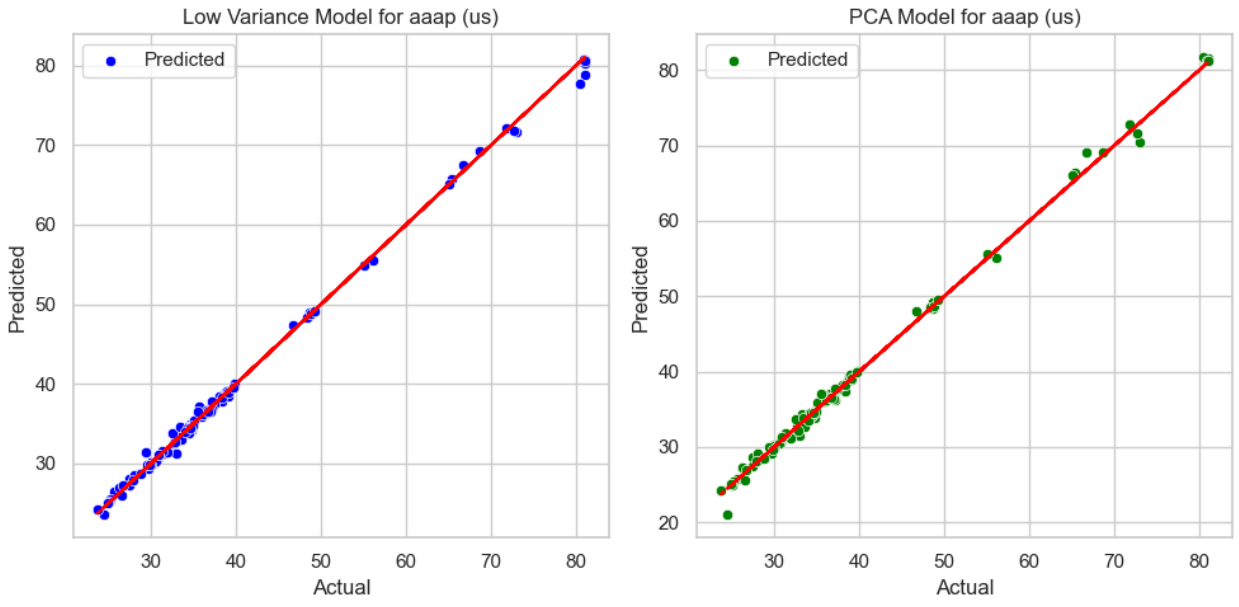



```
Results for aa (us):
  Low Variance Filter MSE: 0.09160272094390905
  PCA MSE: 0.23049572282259112

Processing Stock: aaap, Country: us
```

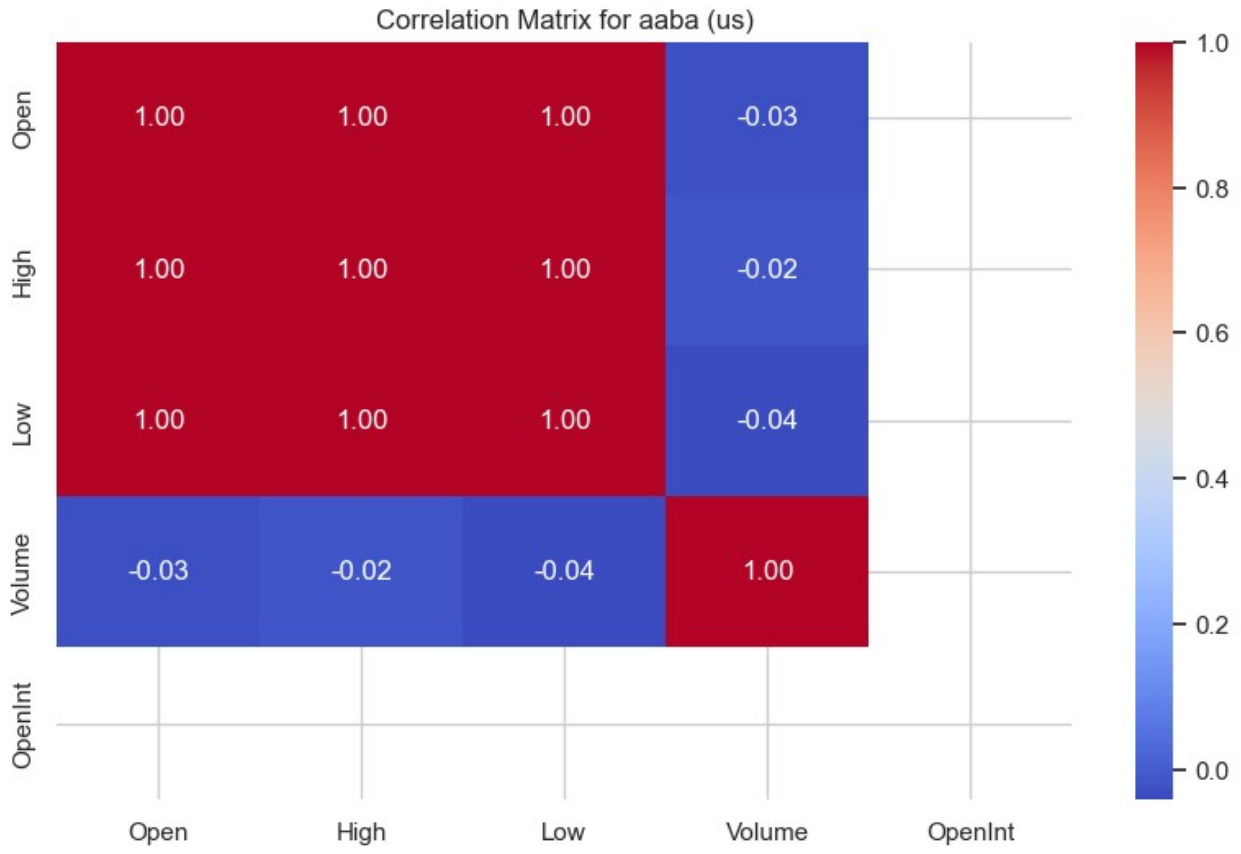

Correlation Matrix for aaap (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
0.43
1.00
1.00
0.99
0.43
1.00
0.99
1.00
0.43
0.43
0.43
0.43
1.00
Open
High
Low
Volume
OpenInt
1.0
0.9
0.8
0.7
0.6
0.5

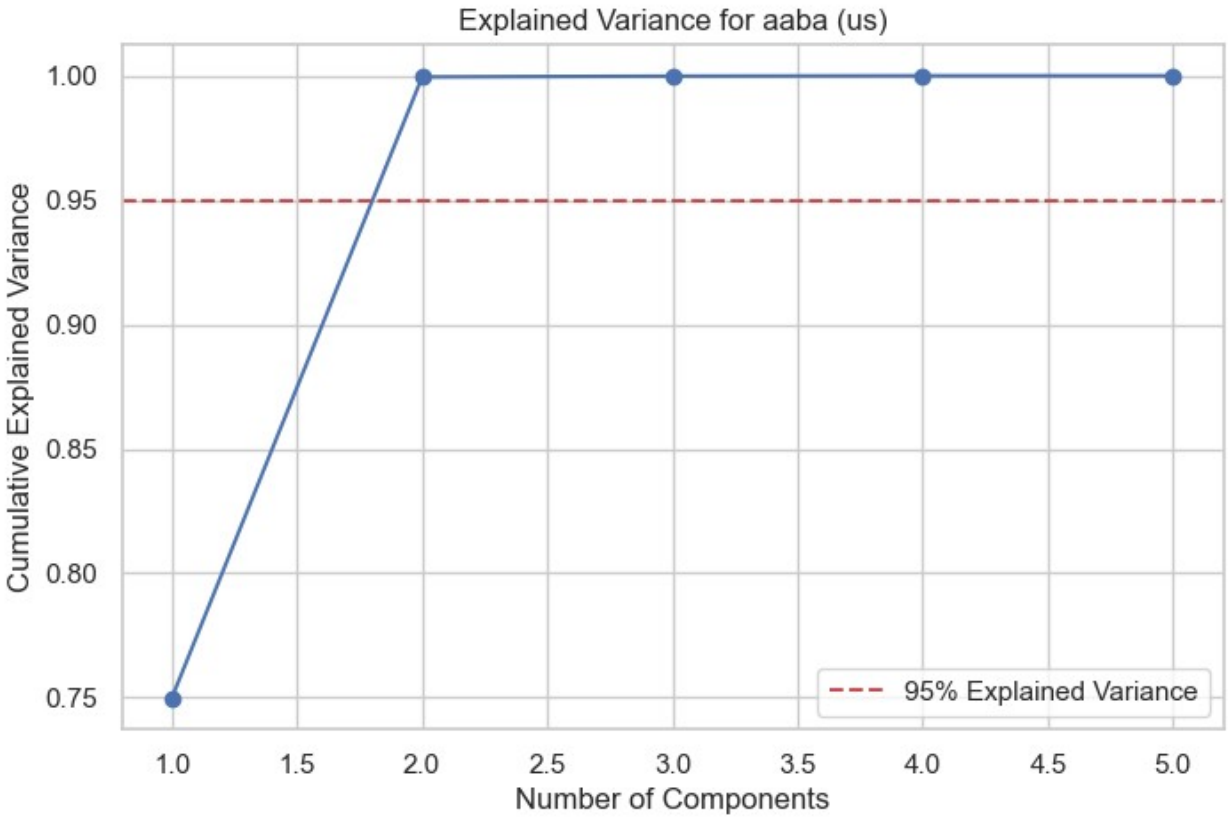

Explained Variance for aaap (us)
Cumulative Explained Variance
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

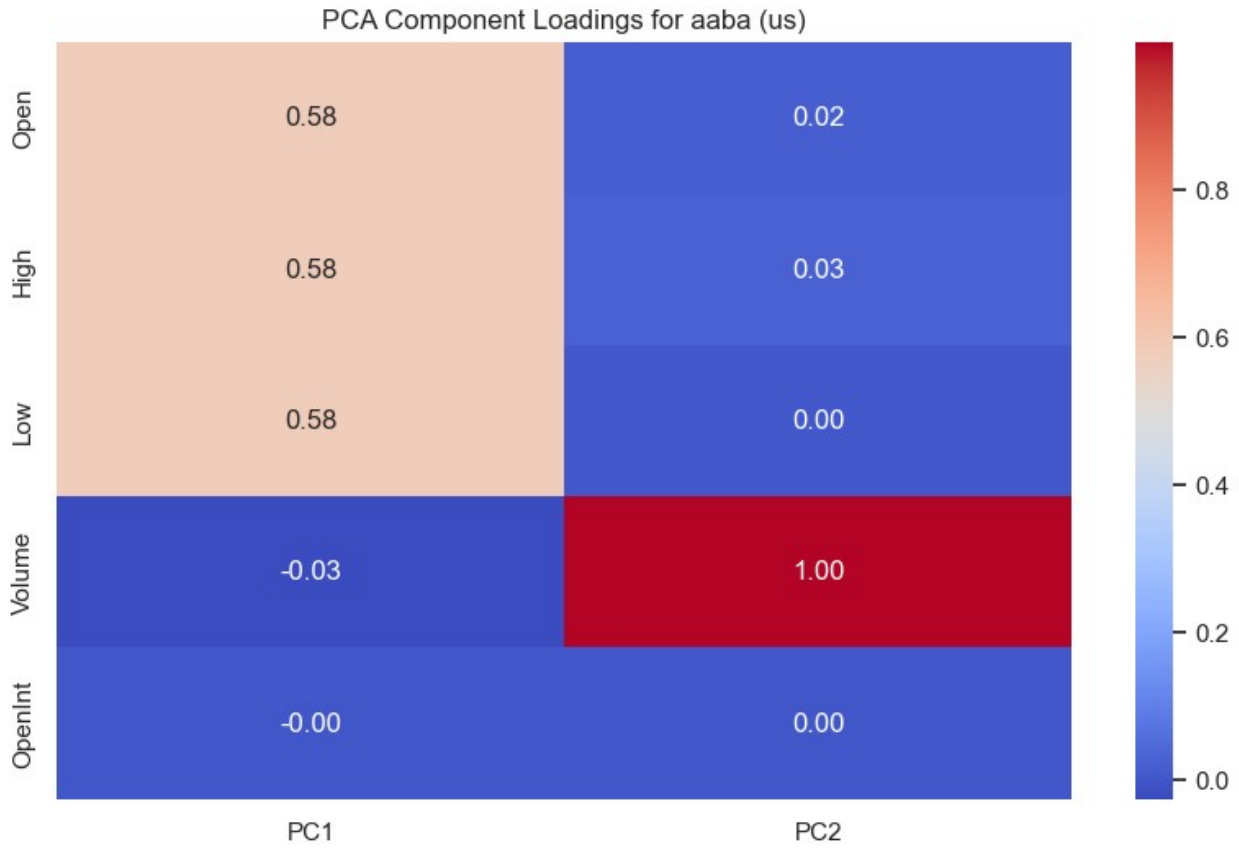


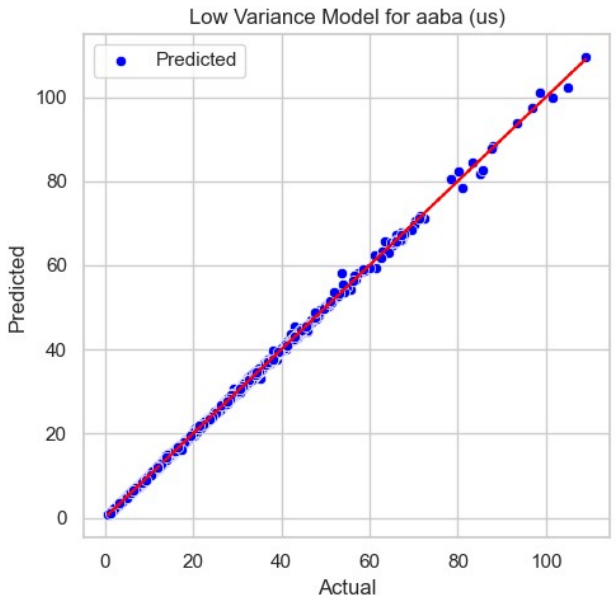


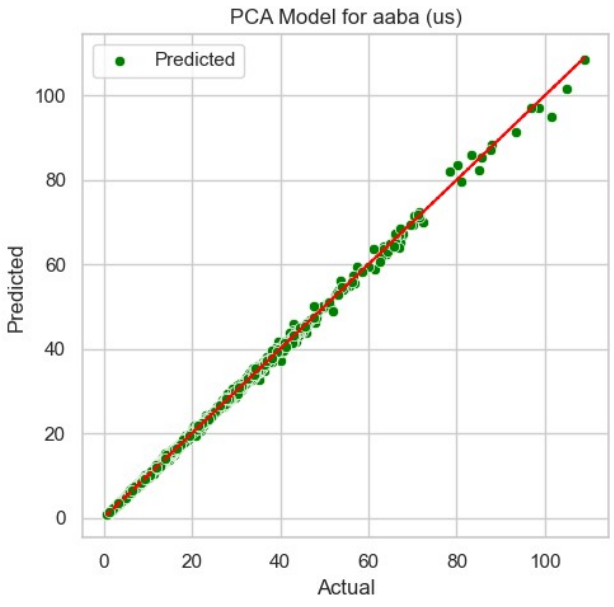

```
Results for aaap (us):
  Low Variance Filter MSE: 0.4084437577248566
  PCA MSE: 0.5807660716257087

Processing Stock: aaba, Country: us
```


Correlation Matrix for aaba (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
-0.03
1.00
1.00
1.00
-0.02
1.00
1.00
1.00
-0.04
-0.03
-0.02
-0.04
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2
0.0

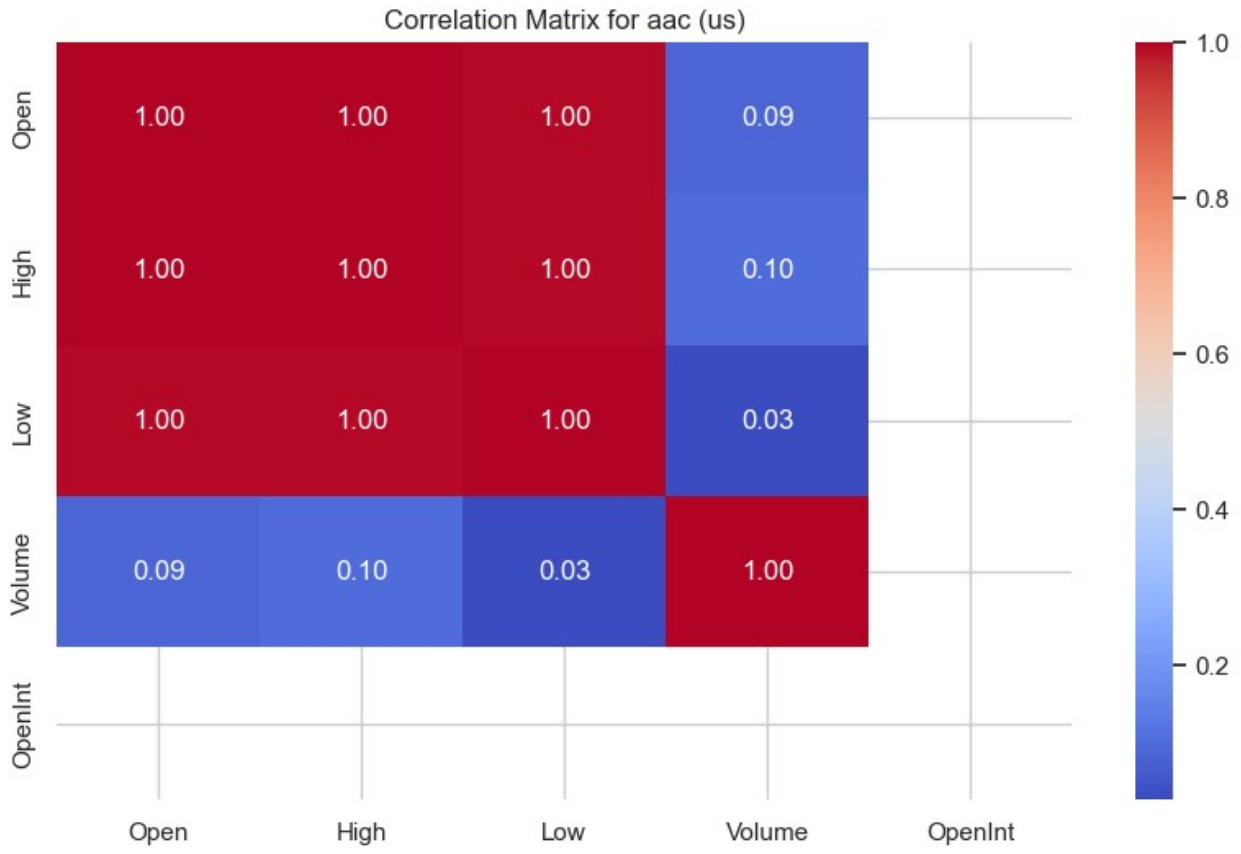

Explained Variance for aaba (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

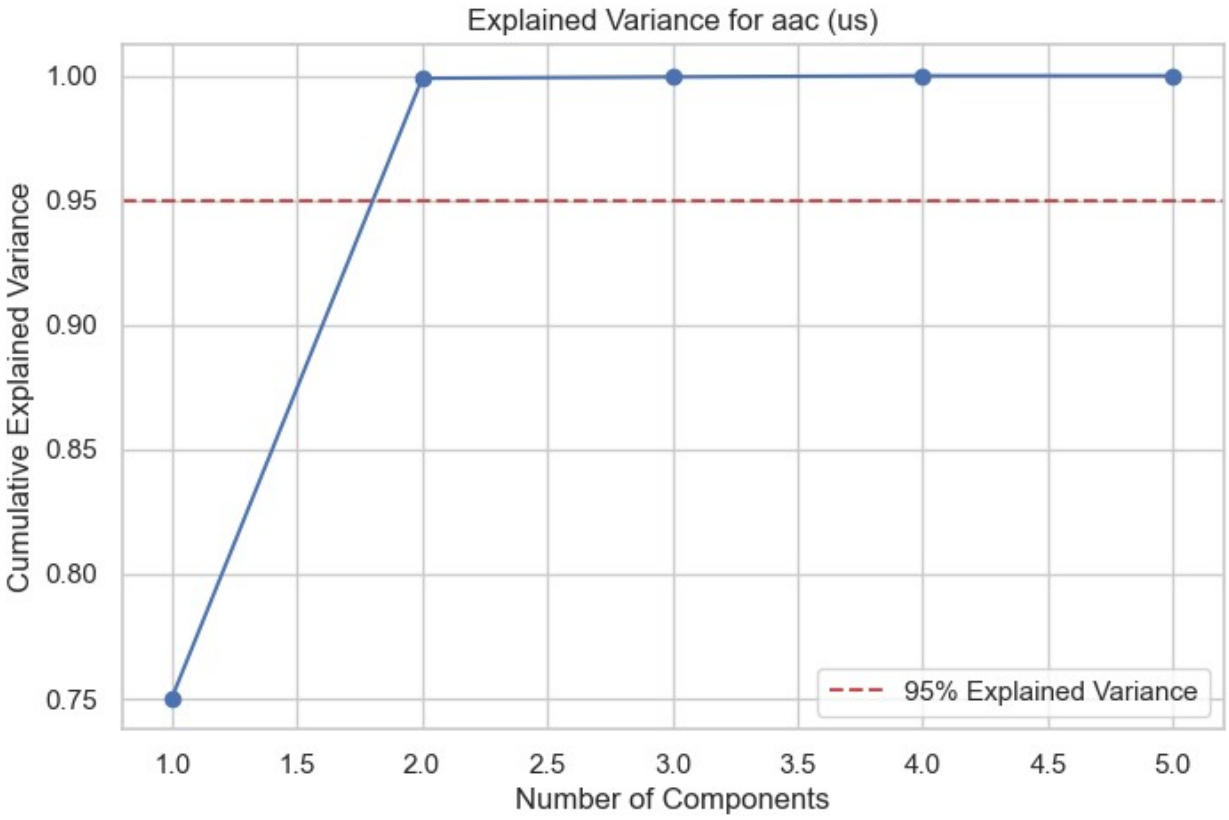


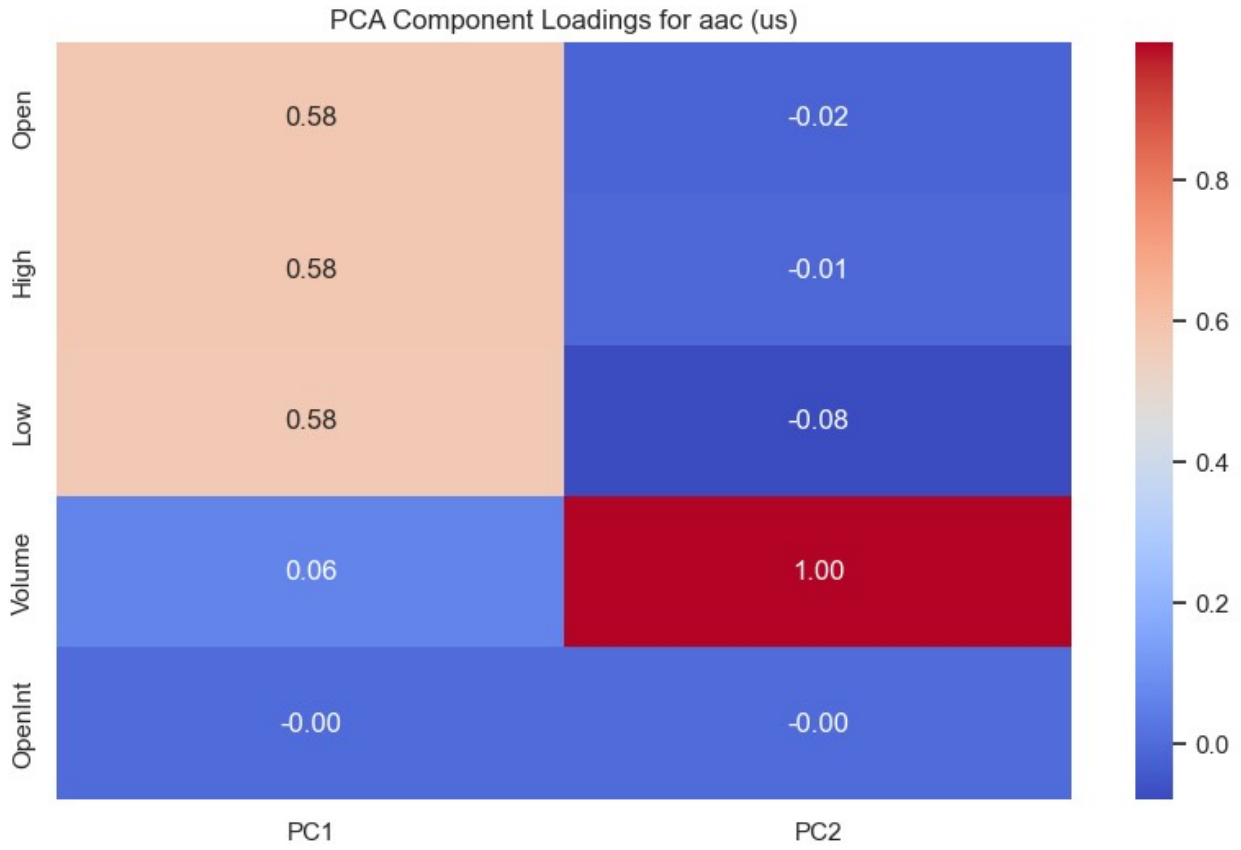


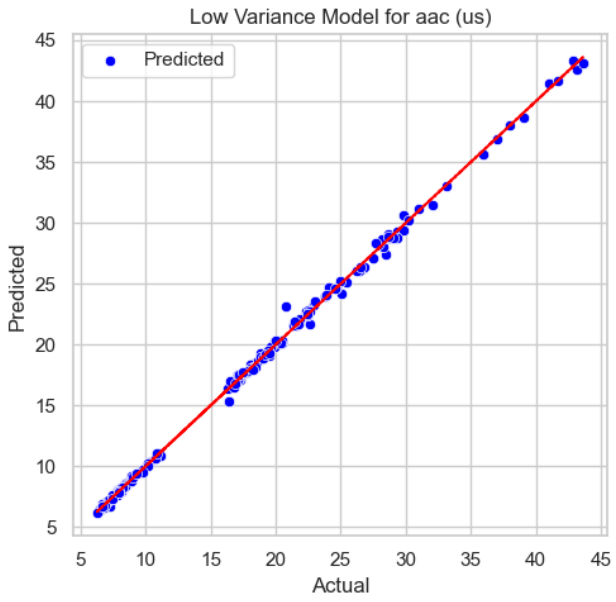


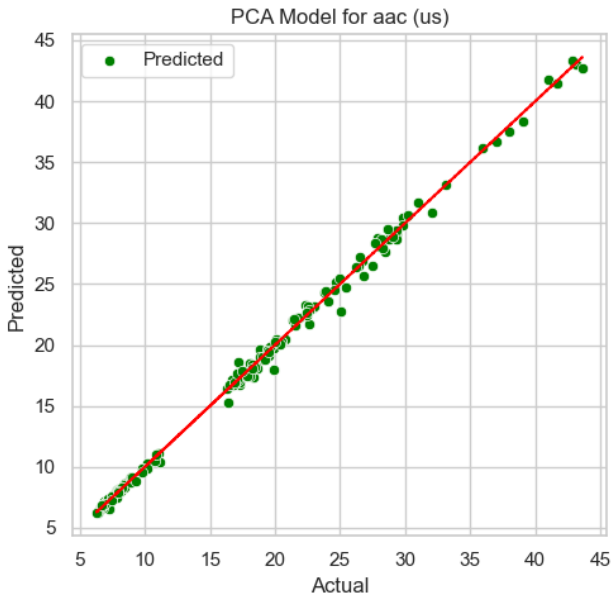

```
Results for aaba (us):
  Low Variance Filter MSE: 0.1682450082300325
  PCA MSE: 0.32529051666181724

Processing Stock: aac, Country: us
```


Correlation Matrix for aac (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
0.09
1.00
1.00
1.00
0.10
1.00
1.00
1.00
0.03
0.09
0.10
0.03
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2

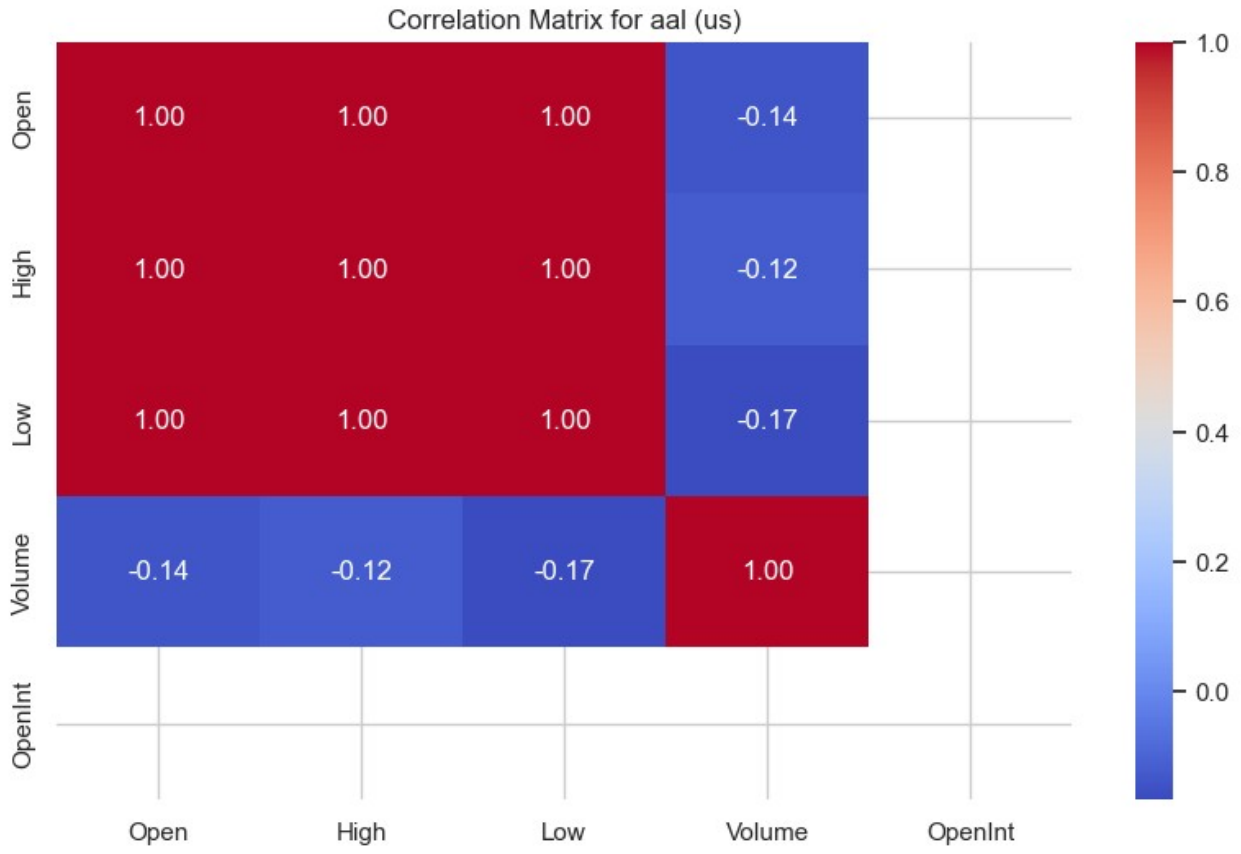

Explained Variance for aac (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

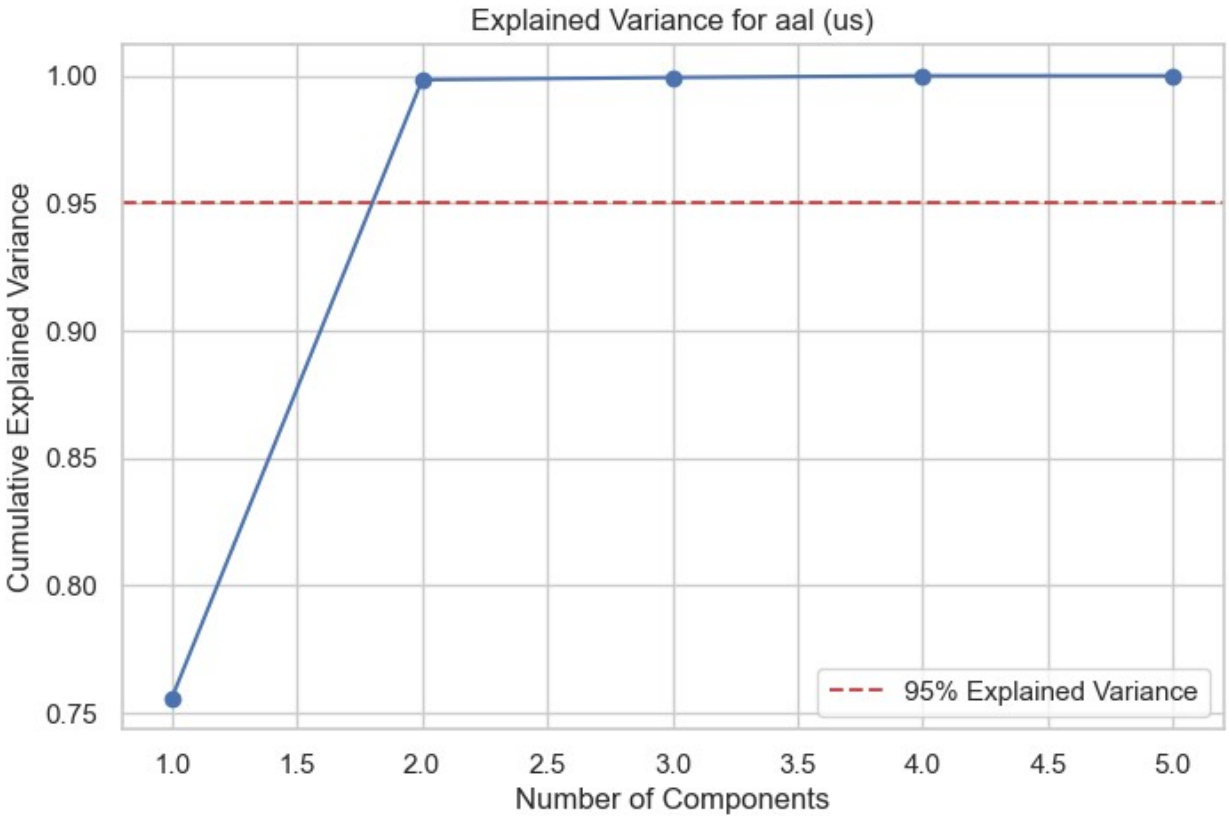


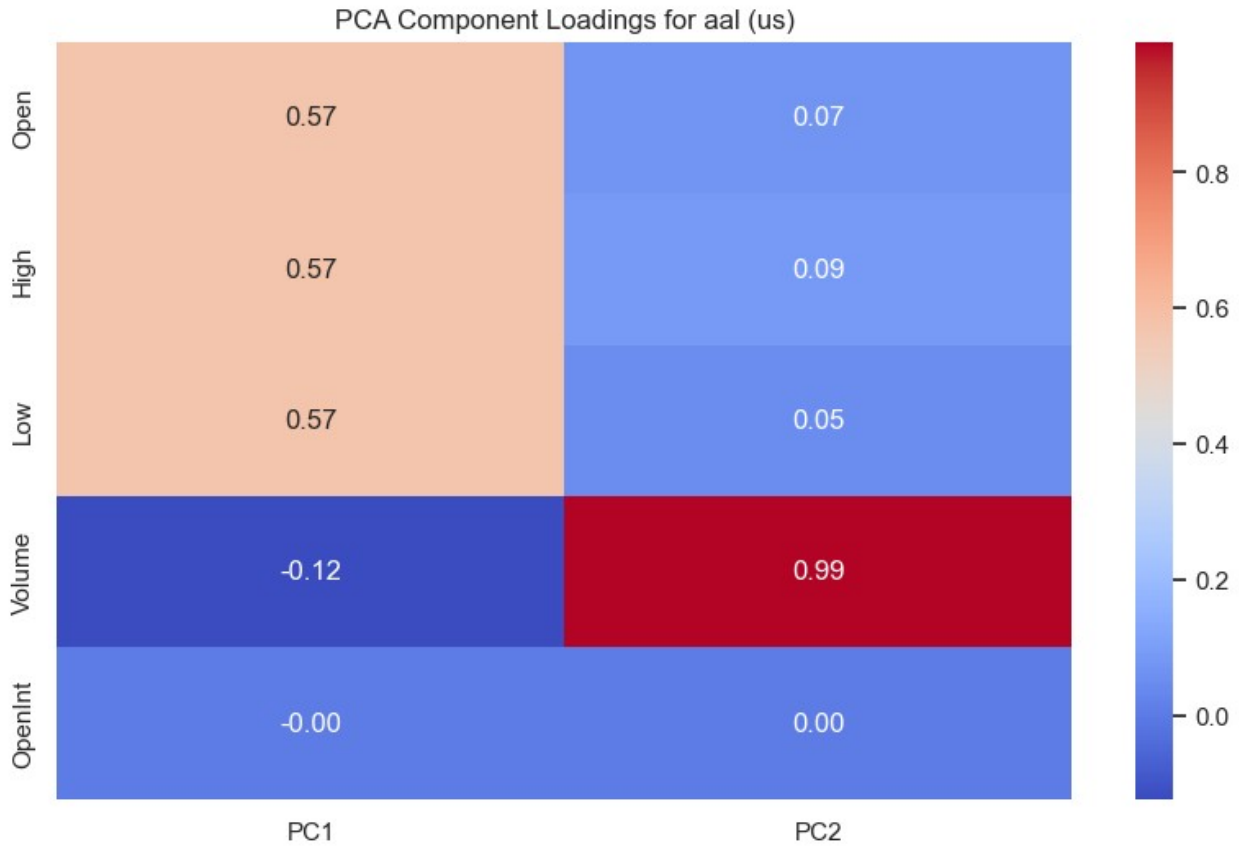


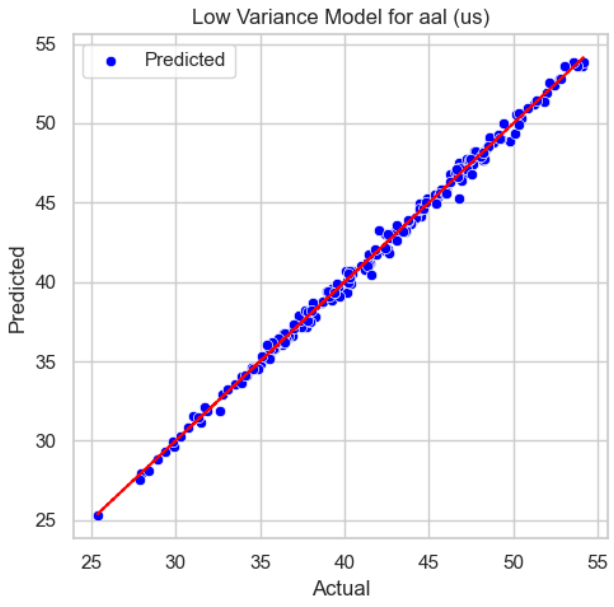


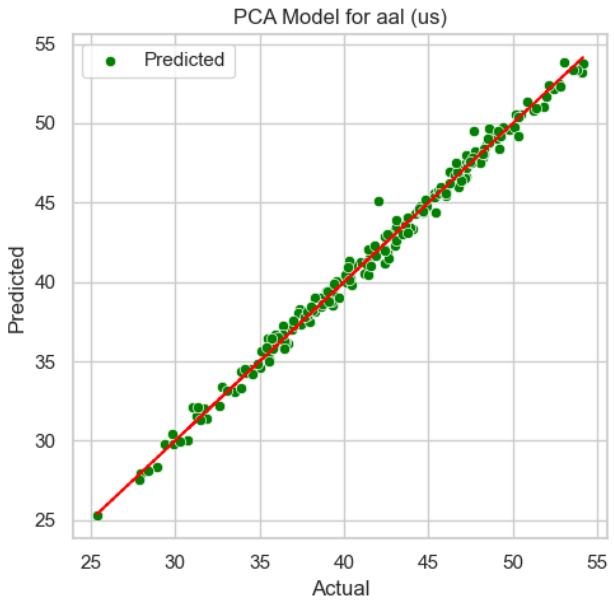

```
Results for aac (us):
  Low Variance Filter MSE: 0.1246420433269073
  PCA MSE: 0.24071217674875375

Processing Stock: aal, Country: us
```


Correlation Matrix for aal (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
-0.14
1.00
1.00
1.00
-0.12
1.00
1.00
1.00
-0.17
-0.14
-0.12
-0.17
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2
0.0

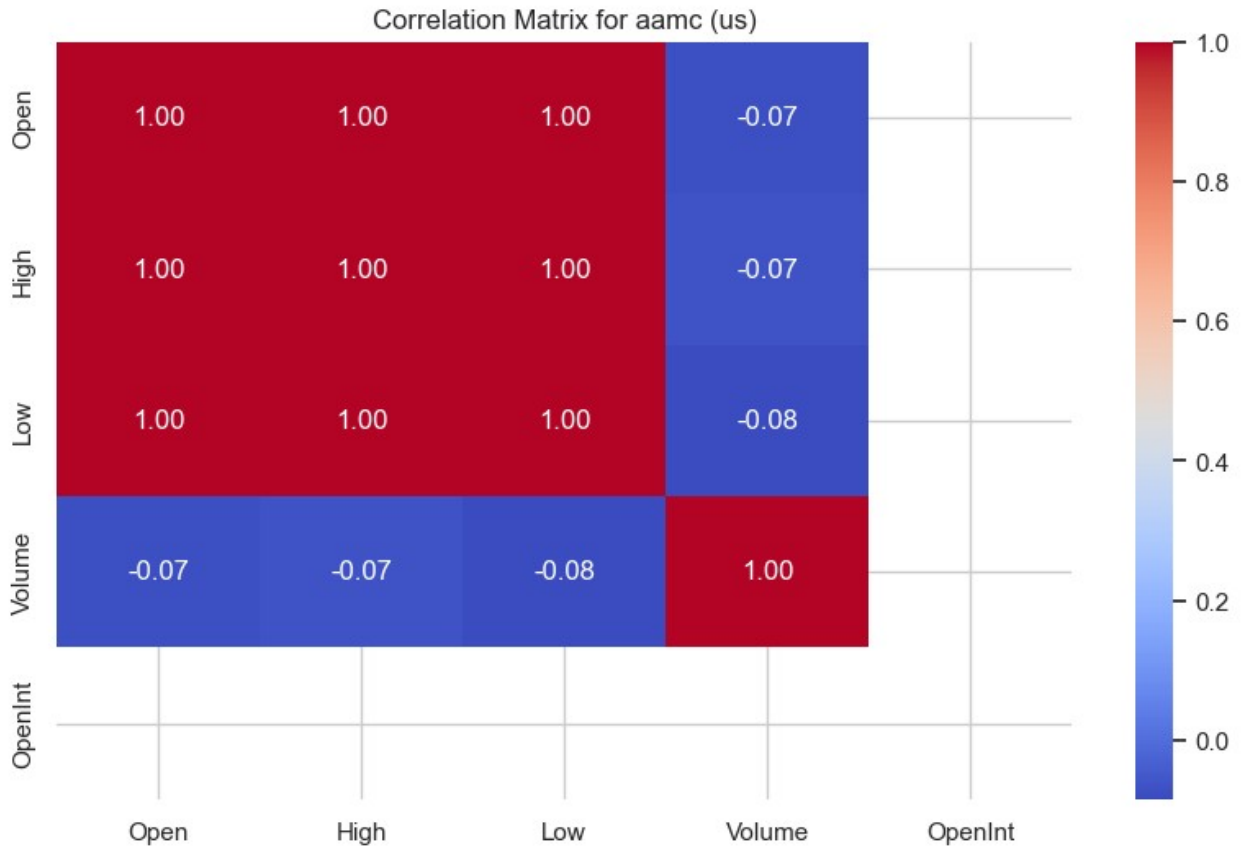

Explained Variance for aal (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

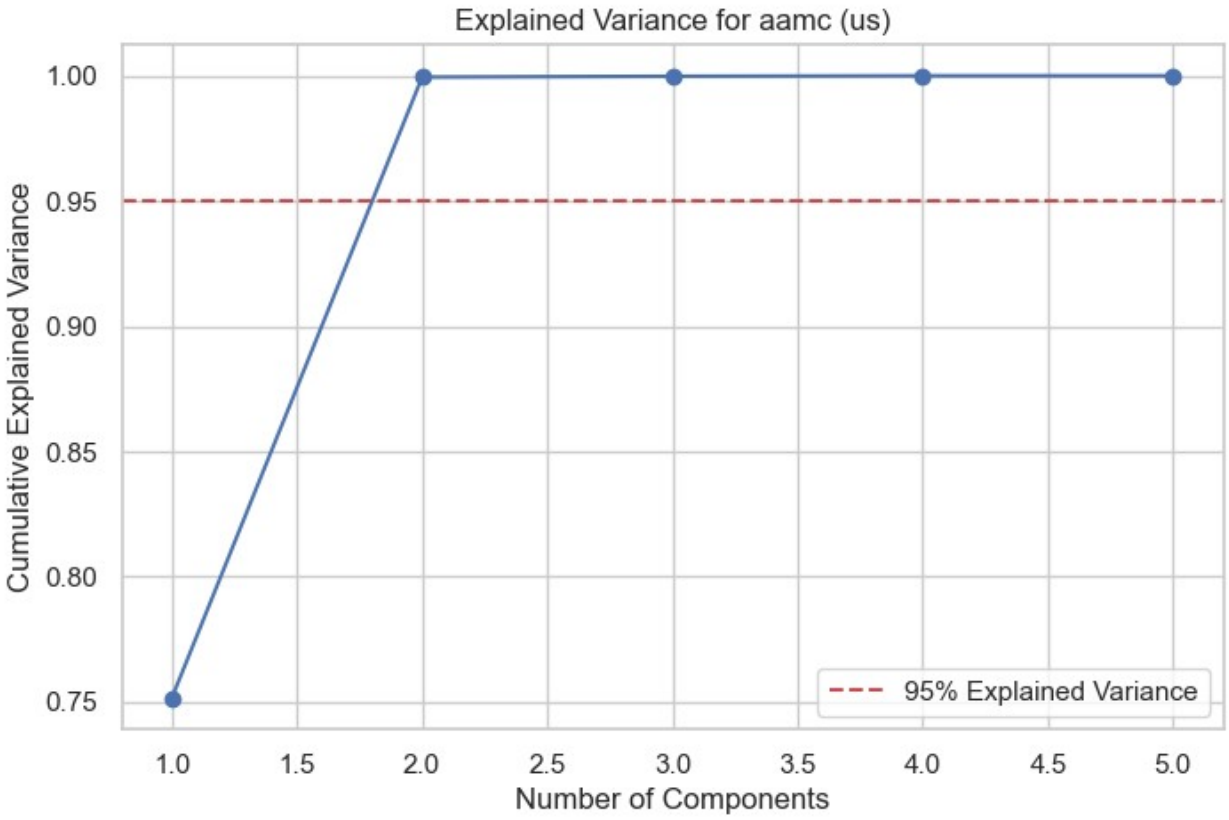


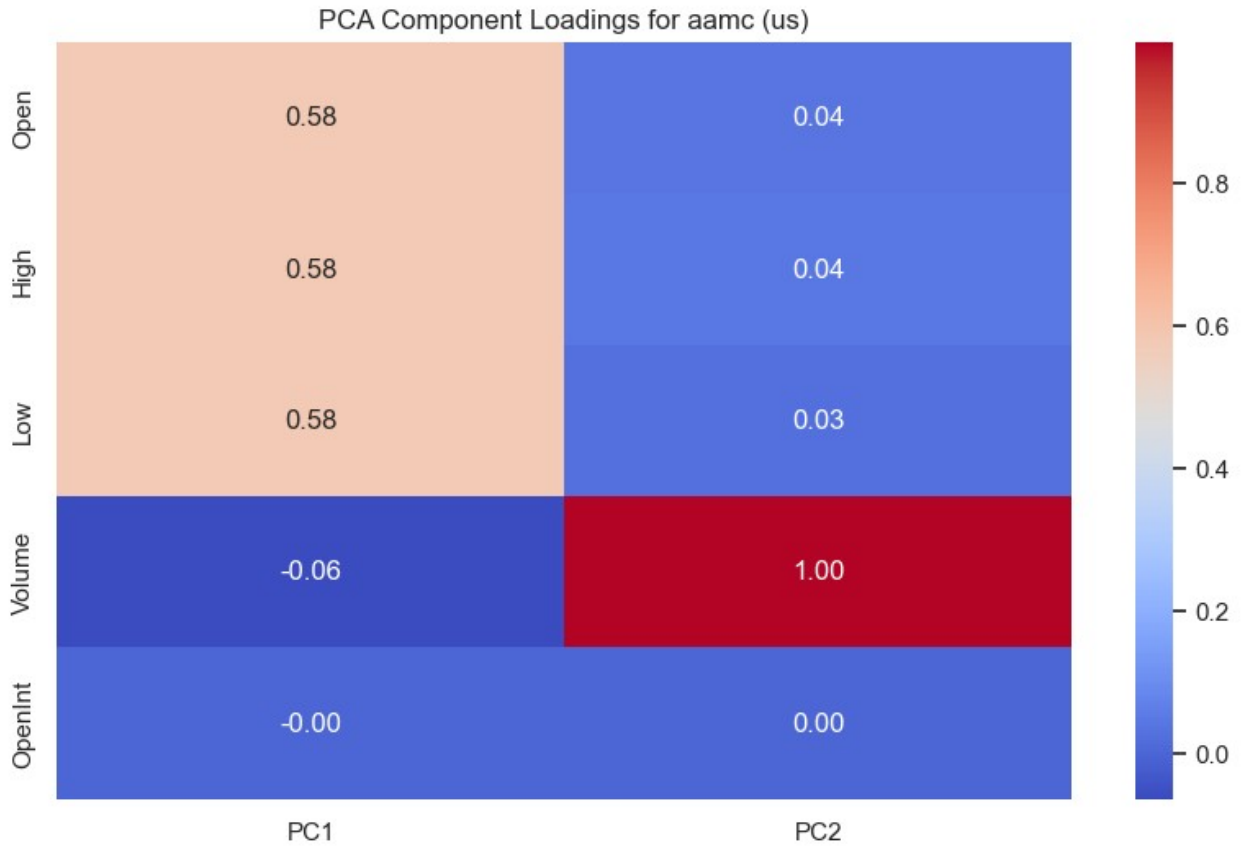


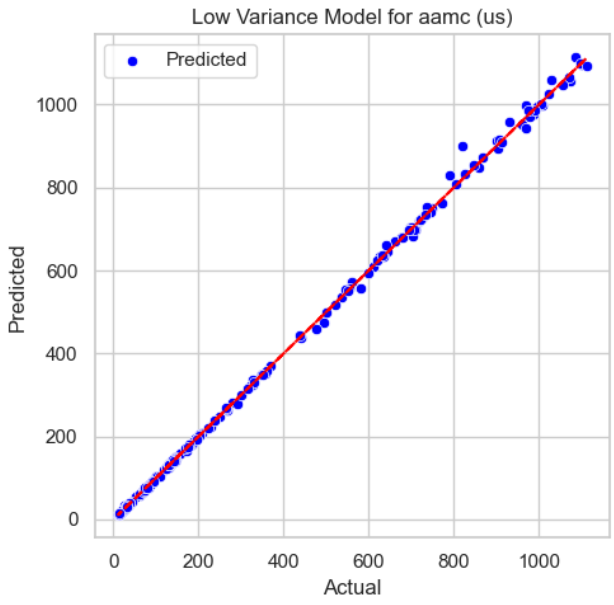


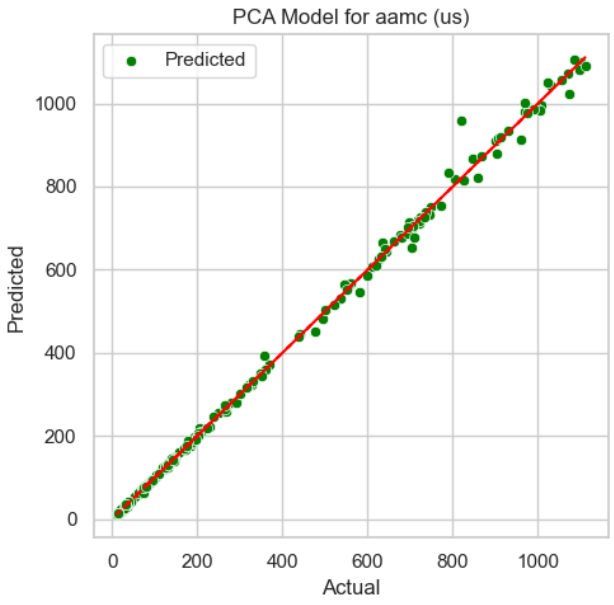

```
Results for aal (us):
  Low Variance Filter MSE: 0.13350377318297585
  PCA MSE: 0.2867004303670024

Processing Stock: aamc, Country: us
```


Correlation Matrix for aamc (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
-0.07
1.00
1.00
1.00
-0.07
1.00
1.00
1.00
-0.08
-0.07
-0.07
-0.08
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2
0.0

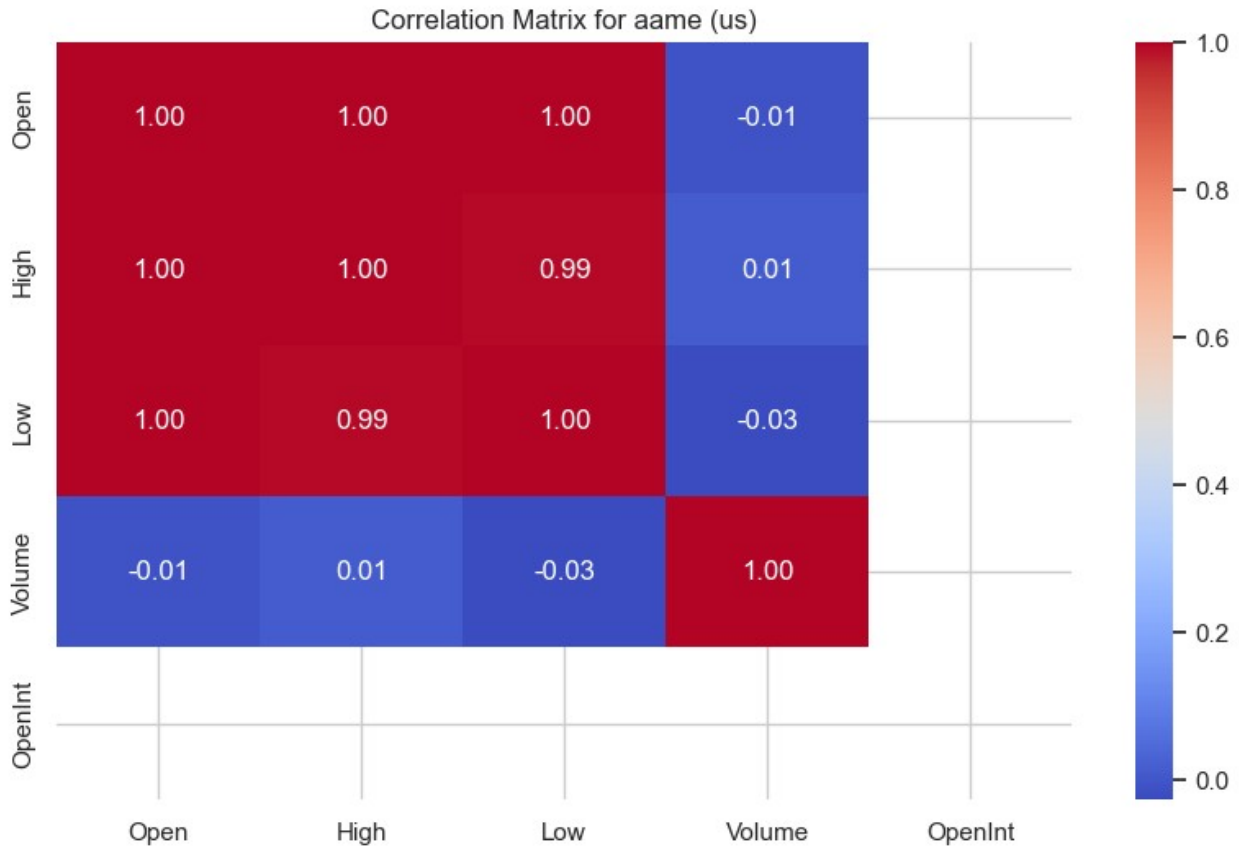

Explained Variance for aamc (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

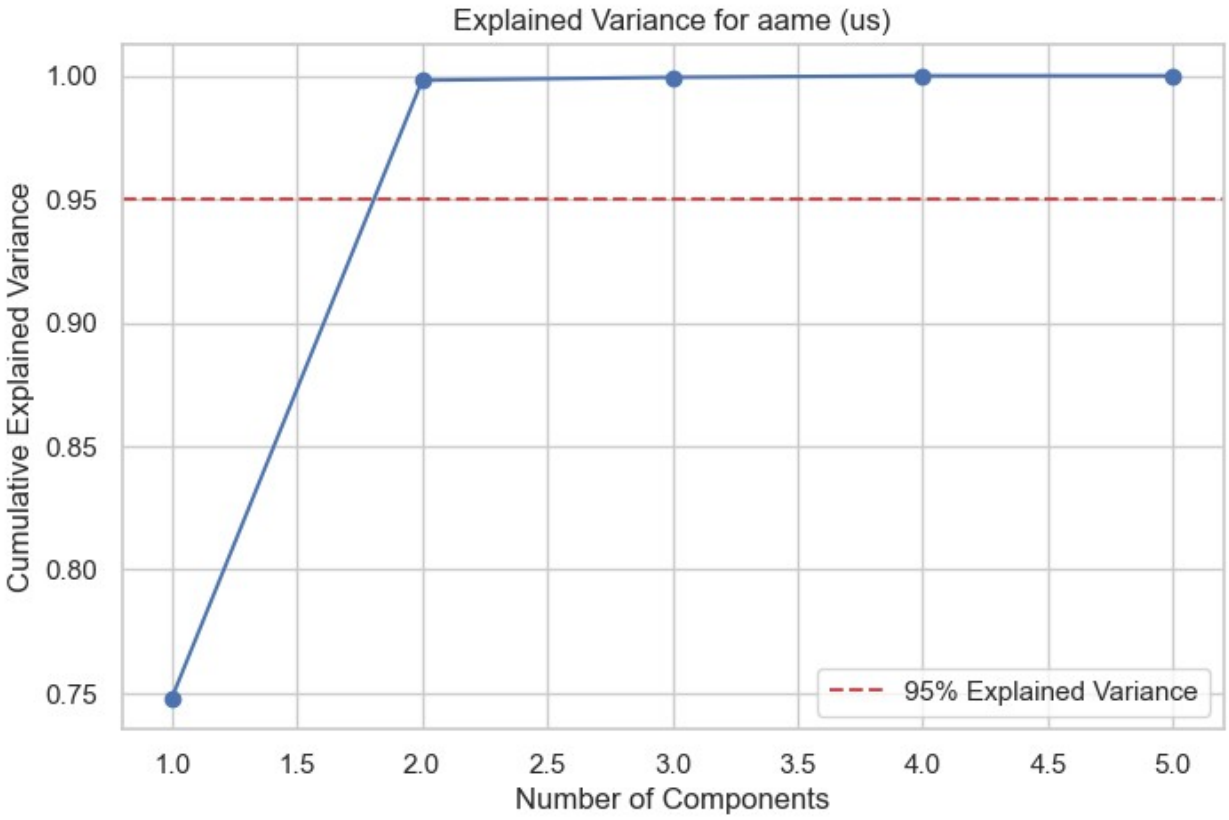


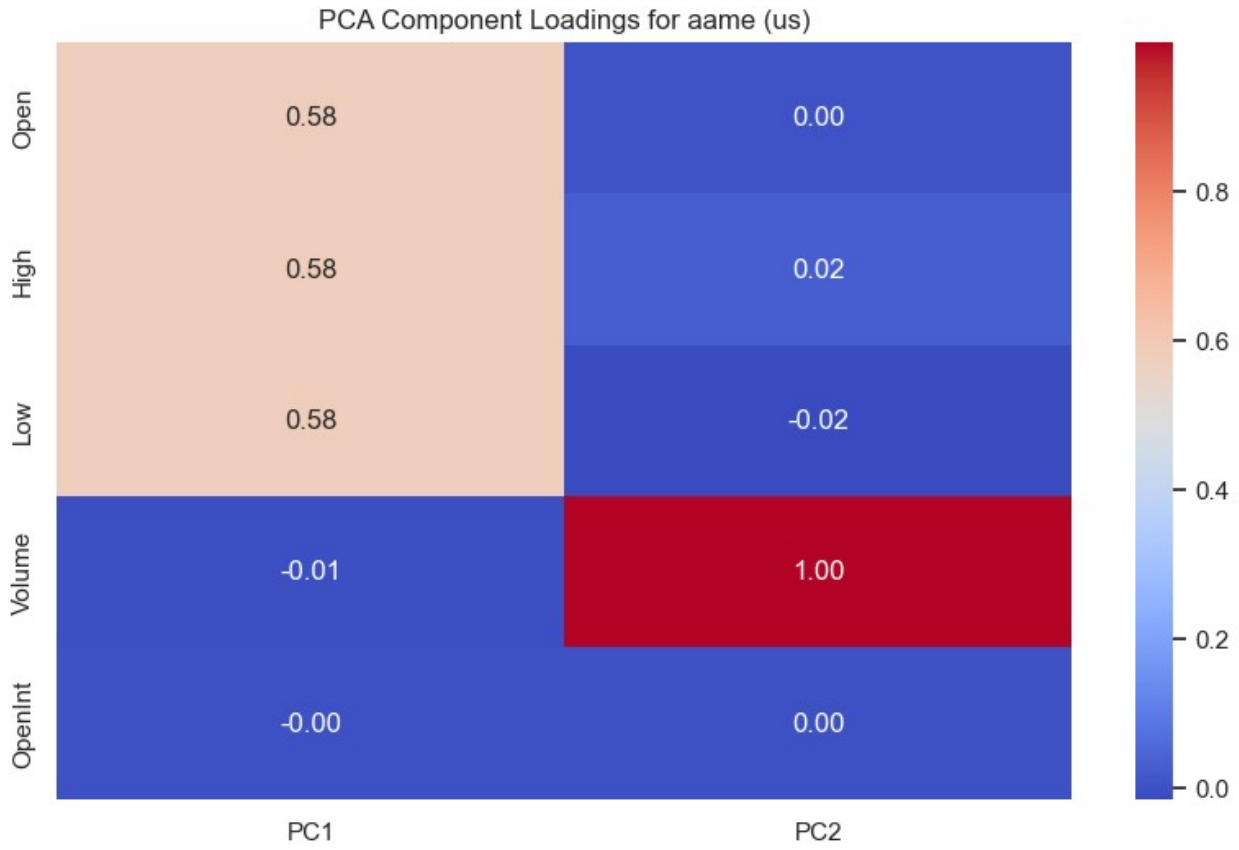


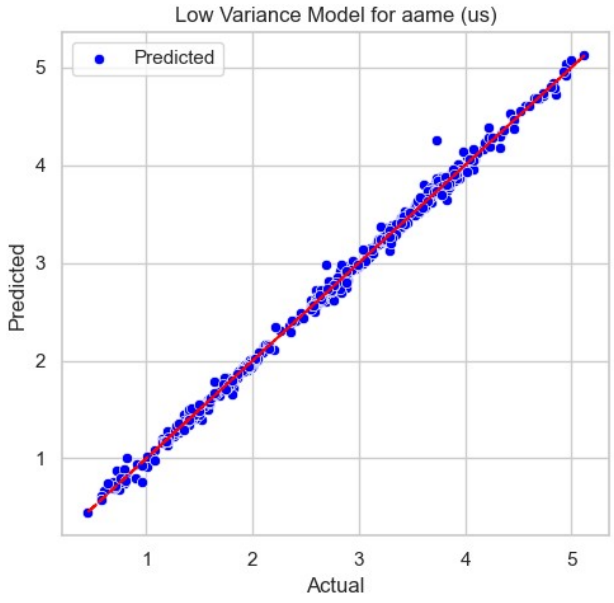


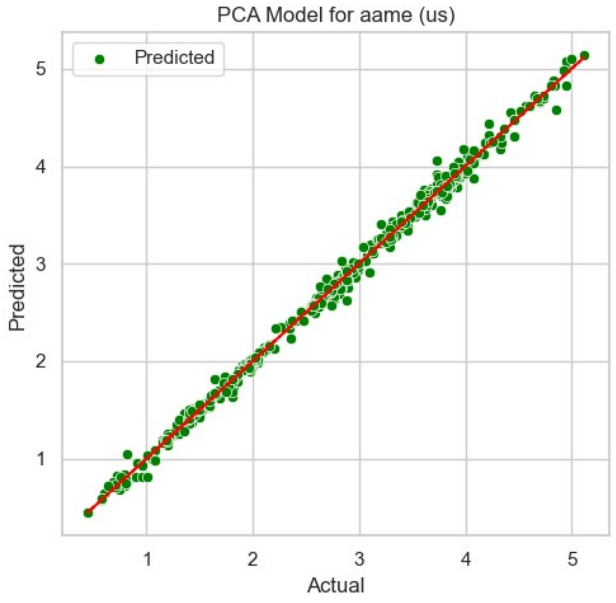

```
Results for aamc (us):
  Low Variance Filter MSE: 75.47607021170002
  PCA MSE: 184.43149173994988

Processing Stock: aame, Country: us
```


Correlation Matrix for aame (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
-0.01
1.00
1.00
0.99
0.01
1.00
0.99
1.00
-0.03
-0.01
0.01
-0.03
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2
0.0

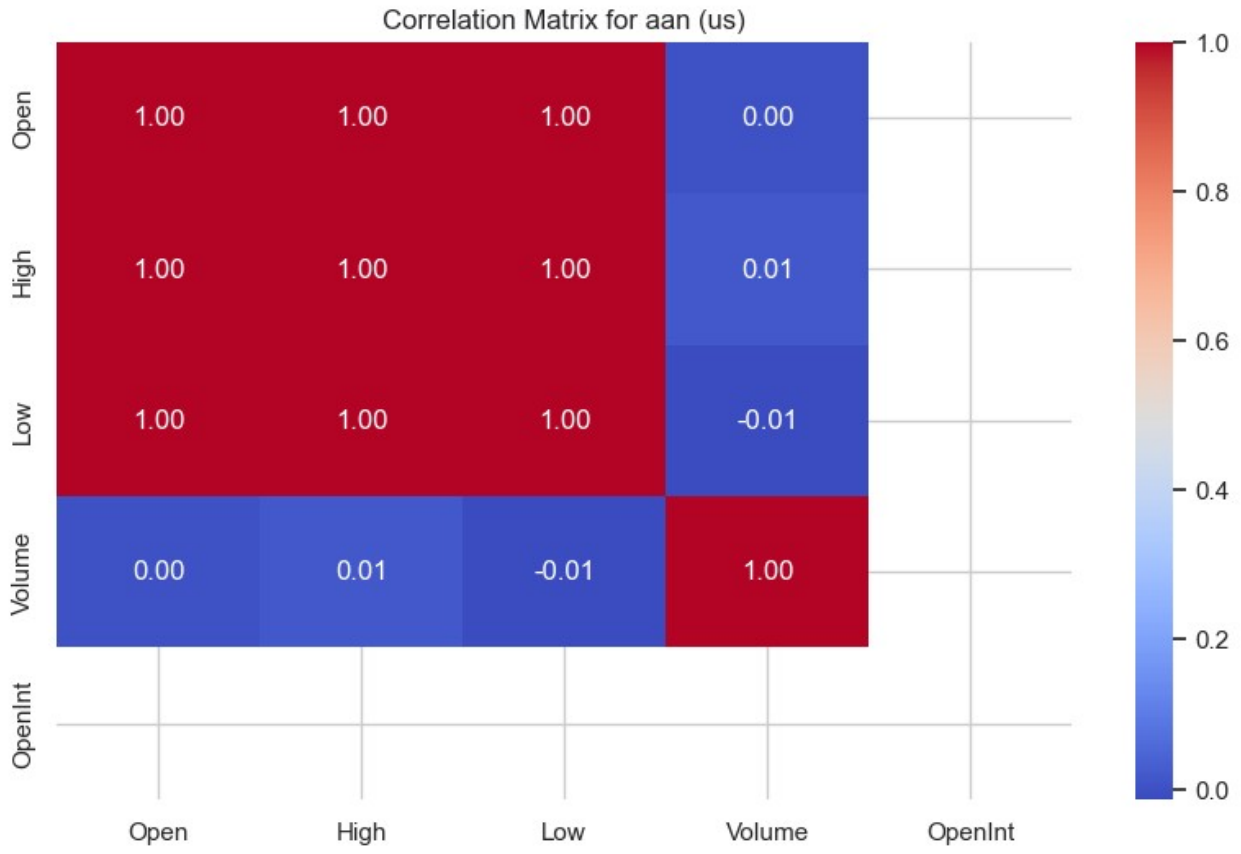

Explained Variance for aame (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

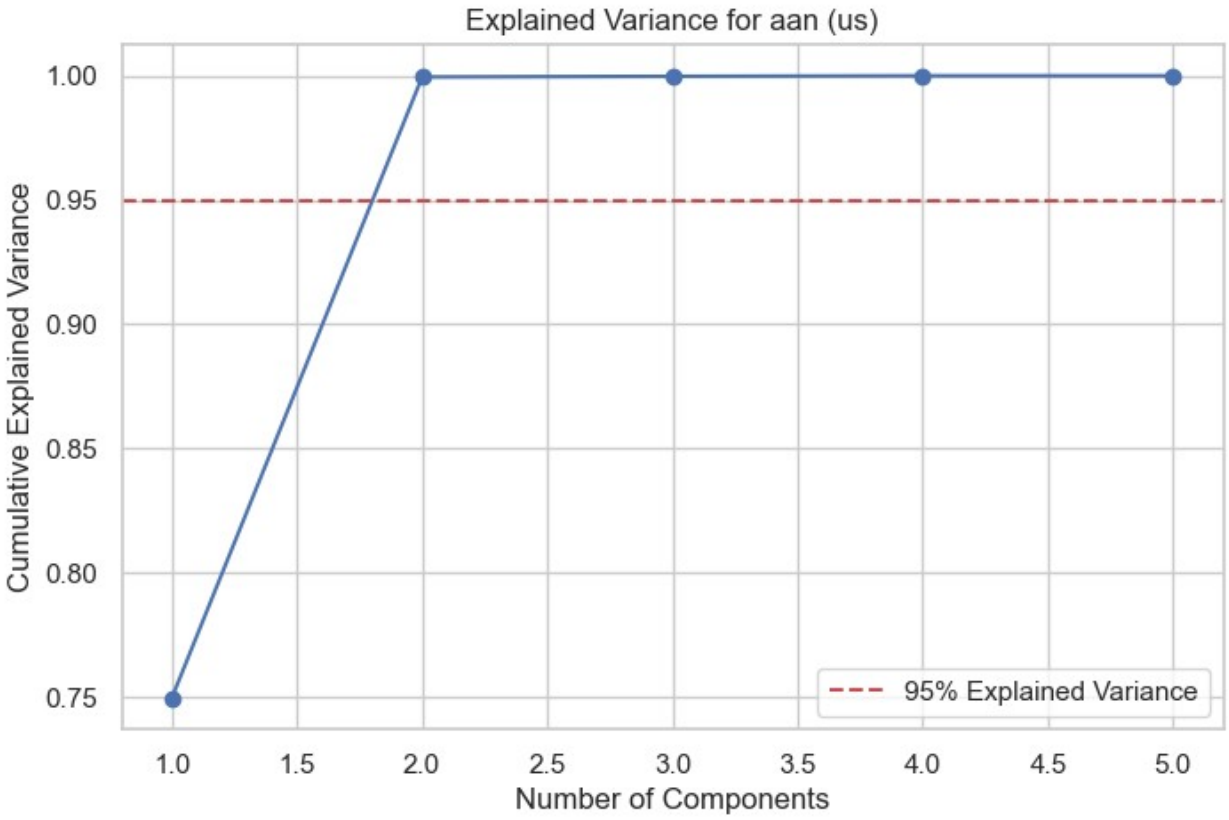


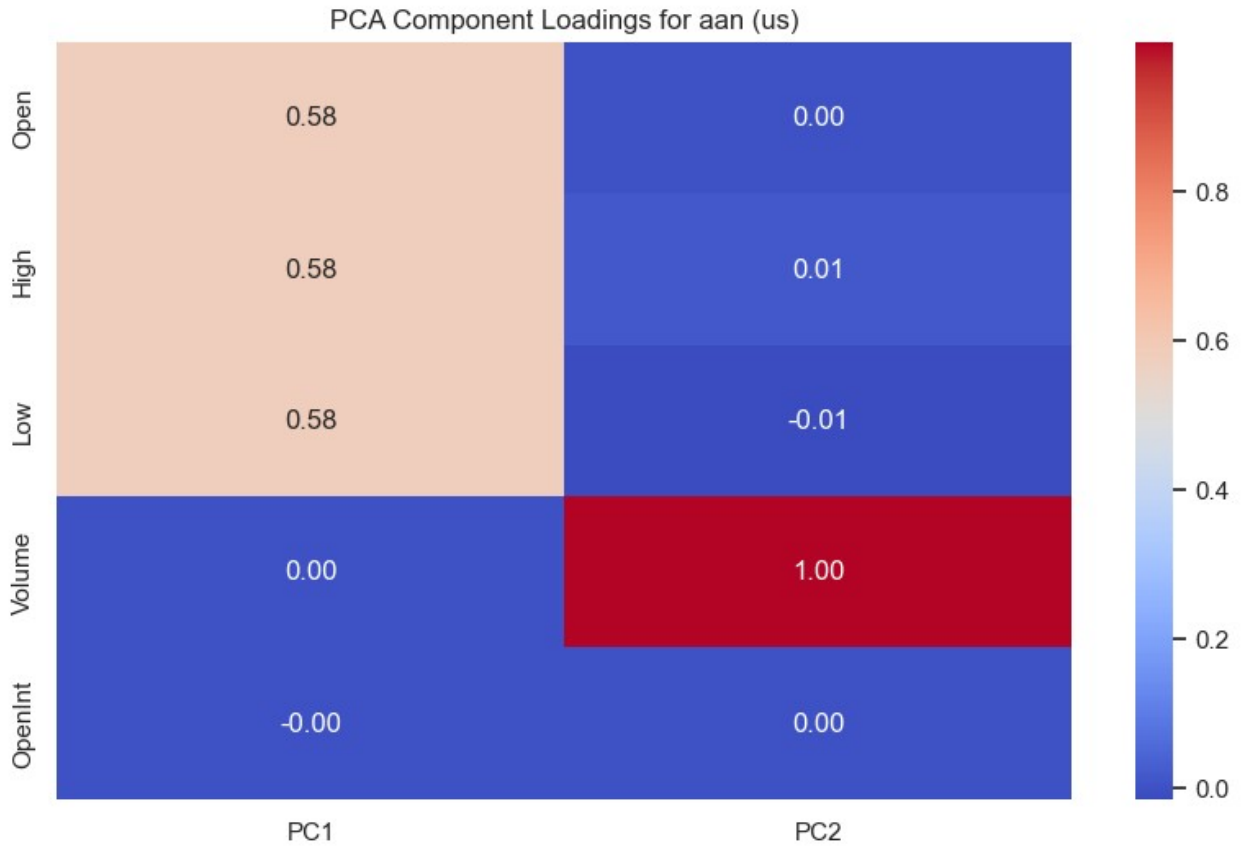


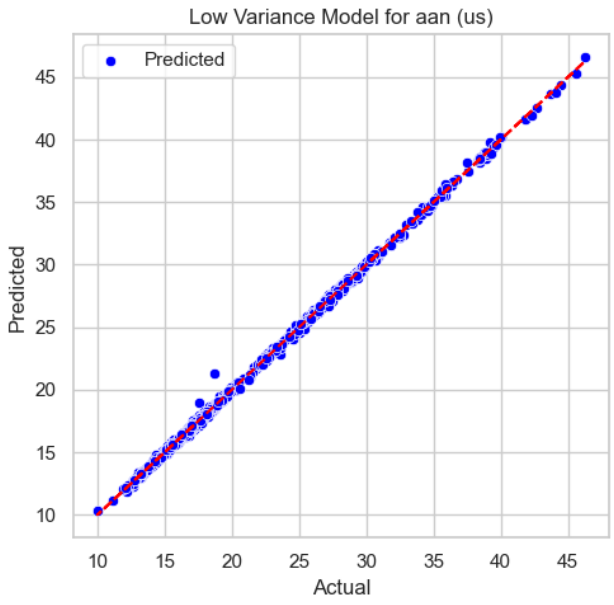


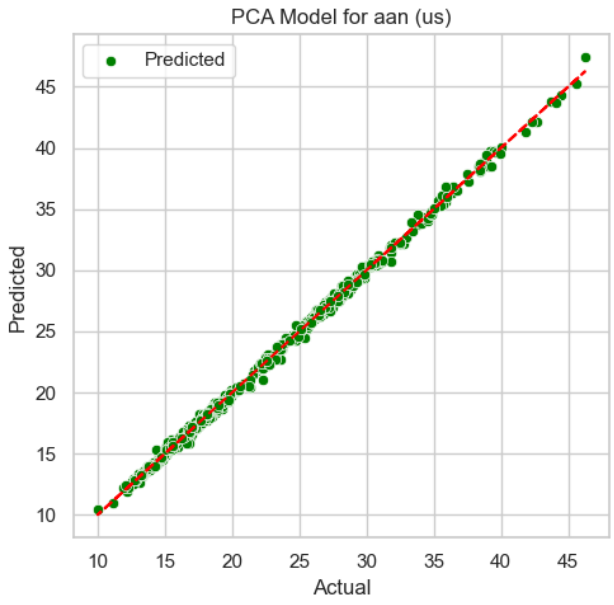

```
Results for aame (us):
  Low Variance Filter MSE: 0.0031016029682854893
  PCA MSE: 0.003679698039043131

Processing Stock: aan, Country: us
```


Correlation Matrix for aan (us)
Open
High
Low
Volume
OpenInt
1.00
1.00
1.00
0.00
1.00
1.00
1.00
0.01
1.00
1.00
1.00
-0.01
0.00
0.01
-0.01
1.00
Open
High
Low
Volume
OpenInt
1.0
0.8
0.6
0.4
0.2
0.0

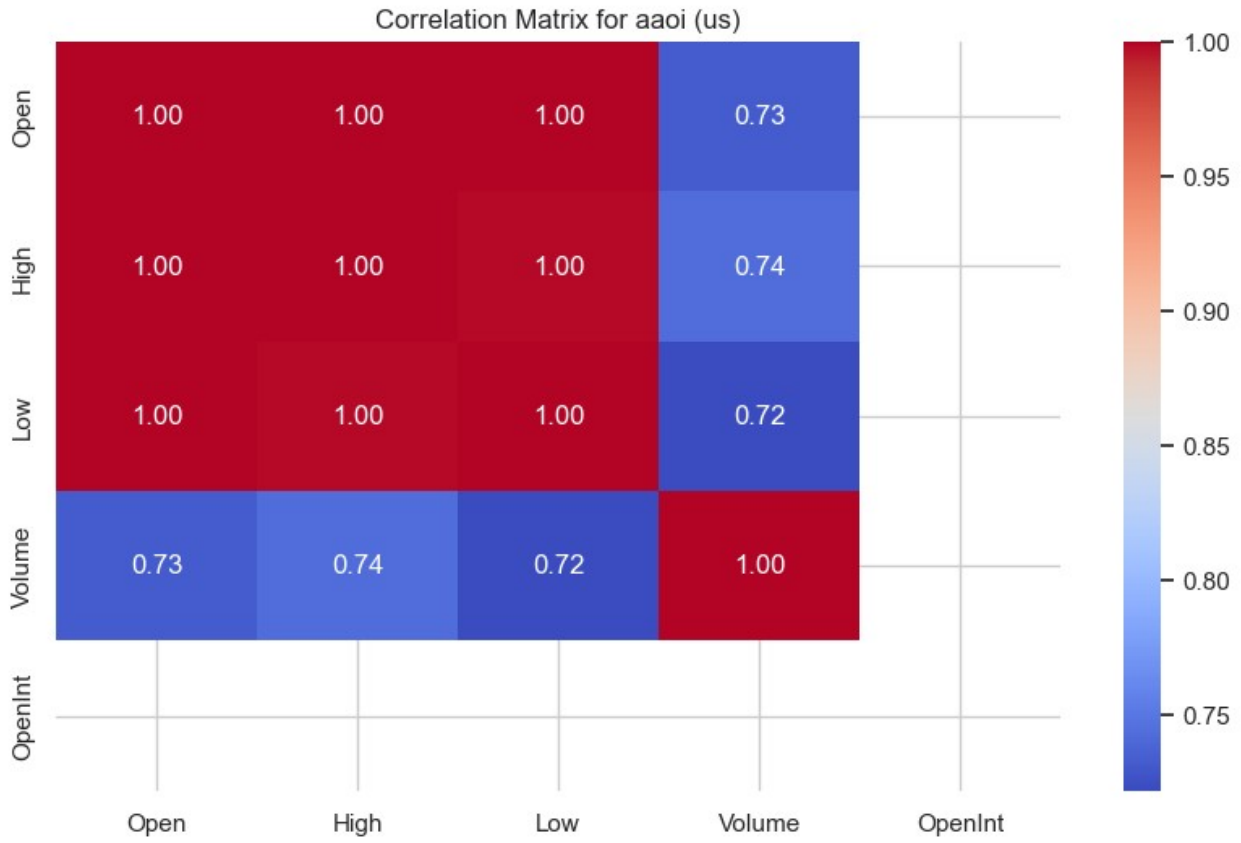

Explained Variance for aan (us)
Cumulative Explained Variance
1.00
0.95
0.90
0.85
0.80
0.75
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components

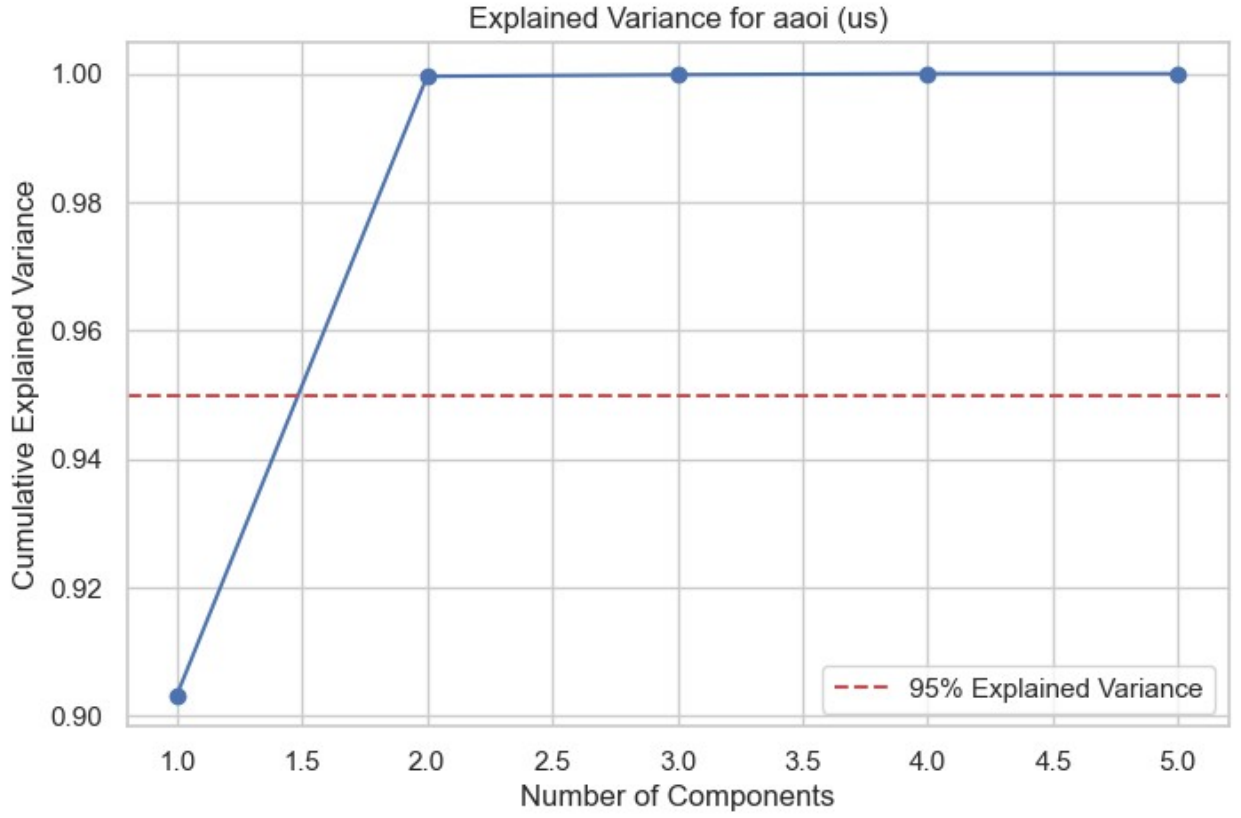


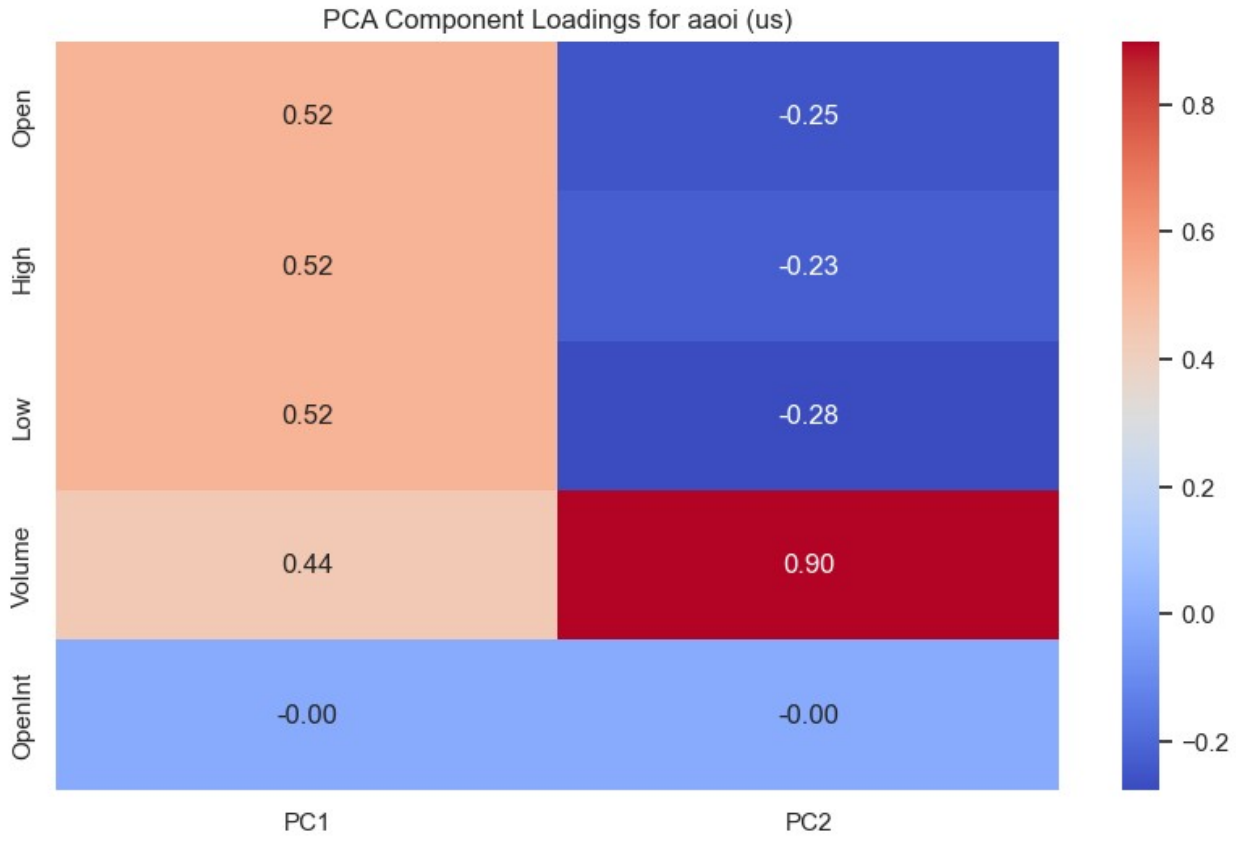


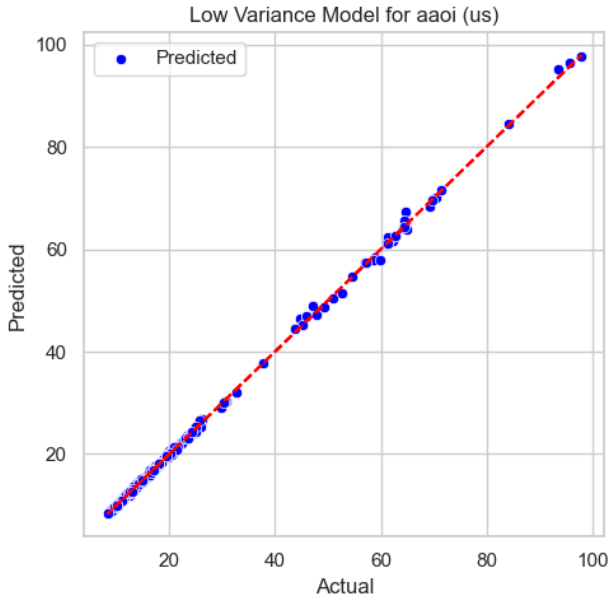


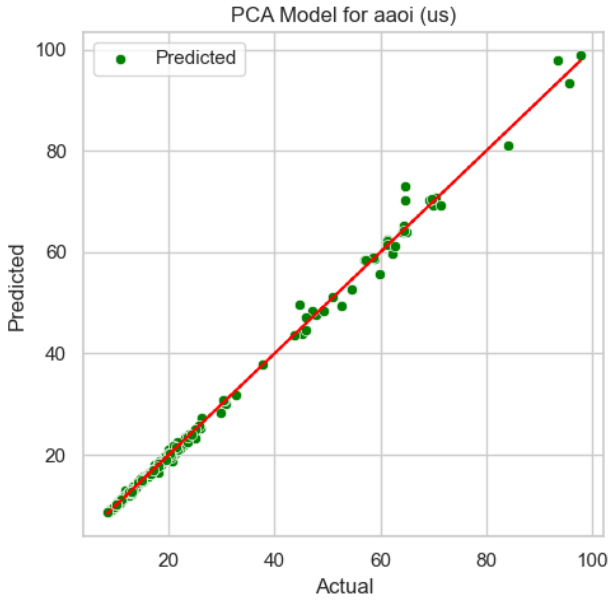

```
Results for aan (us):
  Low Variance Filter MSE: 0.04420974260877737
  PCA MSE: 0.07183079005852355

Processing Stock: aaoi, Country: us
```


Correlation Matrix for aaoi (us)
Open
High
Low
Volume
OpenInt
1.00 1.00 1.00 0.73
1.00 1.00 1.00 0.74
1.00 1.00 1.00 0.72
0.73 0.74 0.72 1.00
Open
High
Low
Volume
OpenInt
1.00
0.95
0.90
0.85
0.80
0.75


Explained Variance for aaoi (us)
Cumulative Explained Variance
1.00
0.98
0.96
0.94
0.92
0.90
95% Explained Variance
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Number of Components








```
Results for aaoi (us):
  Low Variance Filter MSE: 0.2142109920215345
  PCA MSE: 1.253506989271927

Summary of results for each Stock-Country combination:
```

```
a (us) - LV MSE: 0.1286081944181578, PCA MSE: 0.7218900236595417, PCA Components: 2
aa (us) - LV MSE: 0.09160272094390905, PCA MSE: 0.23049572282259112, PCA Components: 2
aaap (us) - LV MSE: 0.4084437577248566, PCA MSE: 0.5807660716257087, PCA Components: 2
aaba (us) - LV MSE: 0.1682450082300325, PCA MSE: 0.32529051666181724, PCA Components: 2
aac (us) - LV MSE: 0.1246420433269073, PCA MSE: 0.24071217674875375, PCA Components: 2
aal (us) - LV MSE: 0.13350377318297585, PCA MSE: 0.2867004303670024, PCA Components: 2
aamc (us) - LV MSE: 75.47607021170002, PCA MSE: 184.43149173994988, PCA Components: 2
aame (us) - LV MSE: 0.0031016029682854893, PCA MSE: 0.003679698039043131, PCA Components: 2
aan (us) - LV MSE: 0.04420974260877737, PCA MSE: 0.07183079005852355, PCA Components: 2
aaoi (us) - LV MSE: 0.2142109920215345, PCA MSE: 1.253506989271927, PCA Components: 2
```