

Python Code:

```
# Python Code for Sentiment Analysis of Political Movements on Twitter
```

```
# Dataset: 'df.csv'
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.tsa.arima_model import ARIMA
```

```
# Load dataset
df = pd.read_csv('df.csv')
```

```
# Preprocessing
X = df['Tweet_Text']
y = df['Sentiment']
```

```
# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
random_state=42)
```

```
# Vectorization using TF-IDF
vectorizer = TfidfVectorizer(max_features=1000)
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)
```

```
# Random Forest Classifier for Sentiment Analysis
model = RandomForestClassifier()
model.fit(X_train_vec, y_train)
```

```
# Predictions
y_pred = model.predict(X_test)
```

```
# Evaluation
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
# Time-series Sentiment Shift Analysis
df['Timestamp'] = pd.to_datetime(df['Timestamp'])
sentiment_over_time = df.resample('D', on='Timestamp')['Sentiment'].apply(lambda x:
x.value_counts(normalize=True))
```

```
# Plot sentiment over time
sentiment_over_time.plot(kind='line', figsize=(10, 6))
plt.title('Sentiment Over Time')
plt.xlabel('Date')
plt.ylabel('Sentiment Proportion')
plt.show()
```

```
# Clustering Opinion Groups
kmeans = KMeans(n_clusters=3, random_state=42)
```

```
df['Cluster'] = kmeans.fit_predict(X_train_vec)

# Visualize clusters
plt.scatter(df['Retweets'], df['Likes'], c=df['Cluster'])
plt.title('Opinion Clusters')
plt.xlabel('Retweets')
plt.ylabel('Likes')
plt.show()

# ARIMA for Sentiment Shift Prediction
model = ARIMA(df['Sentiment'], order=(5,1,0))
model_fit = model.fit(dispatch=0)
print(model_fit.summary())

# Plot residuals for ARIMA model
residuals = pd.DataFrame(model_fit.resid)
residuals.plot(title="ARIMA Residuals")
plt.show()
```