```
!pip install yfinance seaborn scikit-learn matplotlib
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.11/dist-packages (0
Requirement already satisfied: seaborn in /usr/local/lib/python3.11/dist-packages (0.
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.11/dist-
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.11/di
Requirement already satisfied: curl_cffi>=0.7 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: protobuf>=3.19.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: websockets>=13.0 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/
Requirement already satisfied: cffi>=1.12.0 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: certifi>=2024.2.2 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (
```

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

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler

plt.style.use('seaborn-v0_8-whitegrid')
```

```
plt.style.use('seaborn-v0_8-whitegrid')
```

```
df = yf.download("AAPL", start="2018-01-01", end="2023-12-31")
df.head()
```

```
YF.download() has changed argument auto_adjust default to True
[*********************100%***********************]  1 of 1 completed
```

| Price | Close | High | Low | Open | Volume |
|---|---|---|---|---|---|
| Ticker | AAPL | AAPL | AAPL | AAPL | AAPL |
| Date | | | | | |
| **2018-01-02** | 40.426819 | 40.436208 | 39.722764 | 39.933983 | 102223600 |
| **2018-01-03** | 40.419773 | 40.964244 | 40.356411 | 40.490179 | 118071600 |
| **2018-01-04** | 40.607533 | 40.710794 | 40.384583 | 40.492536 | 89738400 |
| **2018-01-05** | 41.069866 | 41.156698 | 40.612231 | 40.703758 | 94640000 |
| **2018-01-08** | 40.917316 | 41.213018 | 40.818746 | 40.917316 | 82271200 |

Next steps: Generate code with df | View recommended plots | New interactive sheet

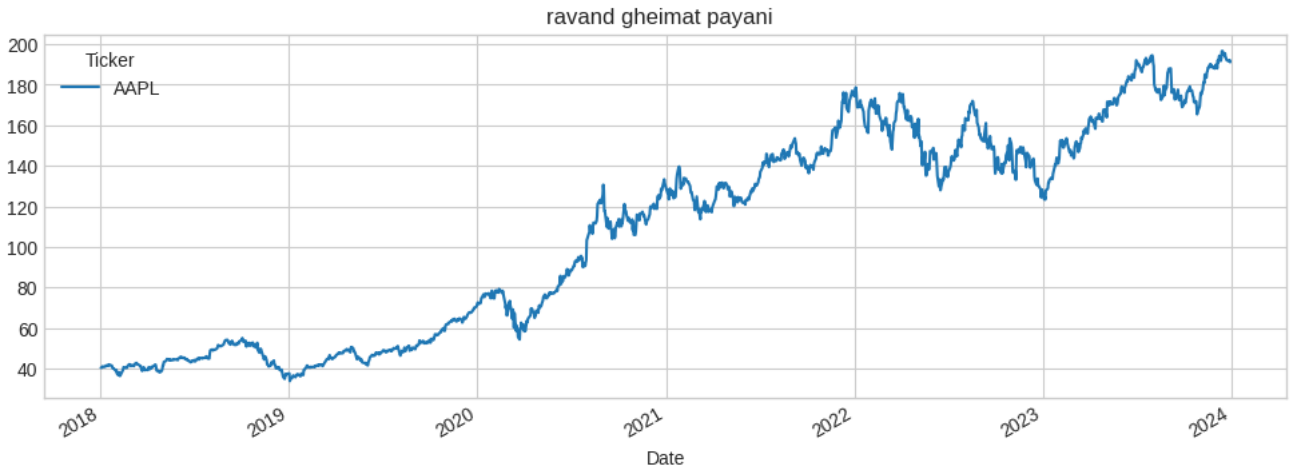
```
# baresi maghadir gom shode
print(df.isnull().sum())

# hazf radif hayi k maghadire gom shode darand
df.dropna(inplace=True)

# tabdil tarikh be datetime index
df.index = pd.to_datetime(df.index)
```

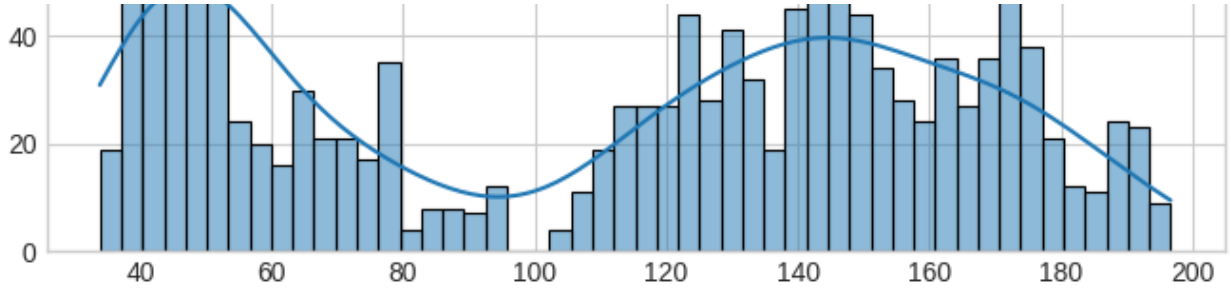
```
Price   Ticker
Close   AAPL      0
High    AAPL      0
Low     AAPL      0
Open    AAPL      0
Volume  AAPL      0
dtype: int64
```

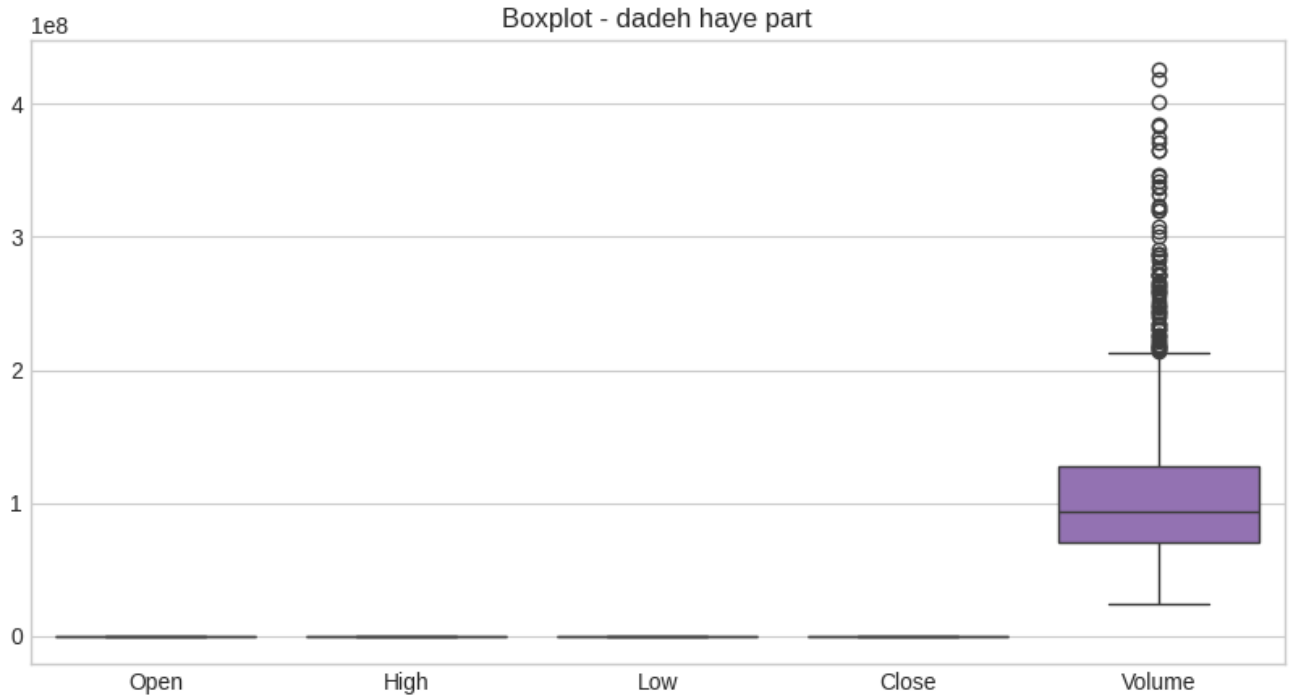
```
scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(df[['Open', 'High', 'Low', 'Close', 'Volume']])
scaled_df = pd.DataFrame(scaled_data, columns=['Open', 'High', 'Low', 'Close', 'Volume'],
```

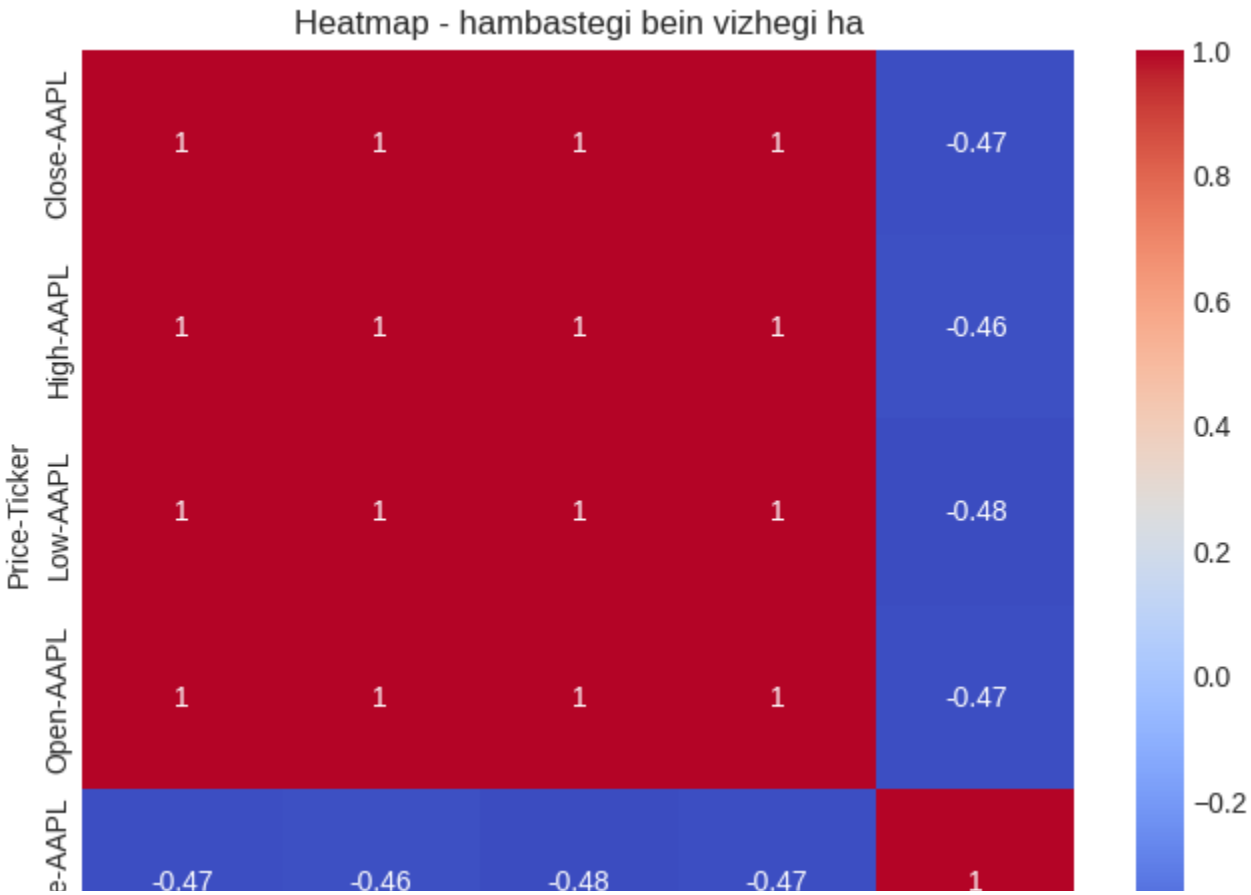
```
# 1. ravand gheimat payani
df['Close'].plot(figsize=(12,4), title='ravand gheimat payani ')
```

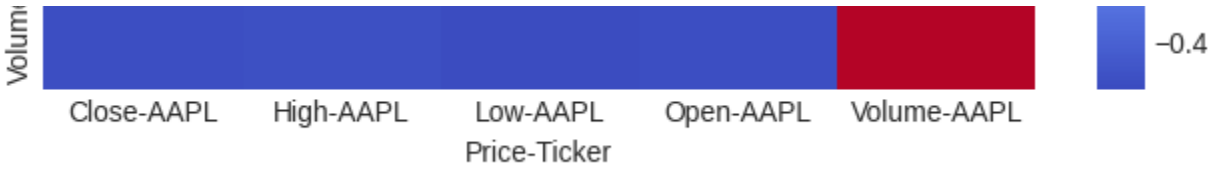
```
df['Close'].plot(figsize=(12,4), title='ravand gheimat payani')

# 2. tozi gheimat payani
plt.figure(figsize=(8,4))
sns.histplot(df['Close'], bins=50, kde=True)
plt.title("Histogram -  tozi gheimat payani")
plt.show()

# 3. Boxplot baraye outliers
plt.figure(figsize=(10,5))
sns.boxplot(data=df[['Open', 'High', 'Low', 'Close', 'Volume']])
plt.title("Boxplot - dadeh haye part")
plt.show()

# 4. Heatmap hambastegui
plt.figure(figsize=(8,6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Heatmap - hambastegi bein vizhegi ha")
plt.show()

# 5. miangin moteharek
df['MA20'] = df['Close'].rolling(window=20).mean()
df['MA50'] = df['Close'].rolling(window=50).mean()
df[['Close', 'MA20', 'MA50']].plot(figsize=(12,4), title='Moving Averages')
plt.show()
```

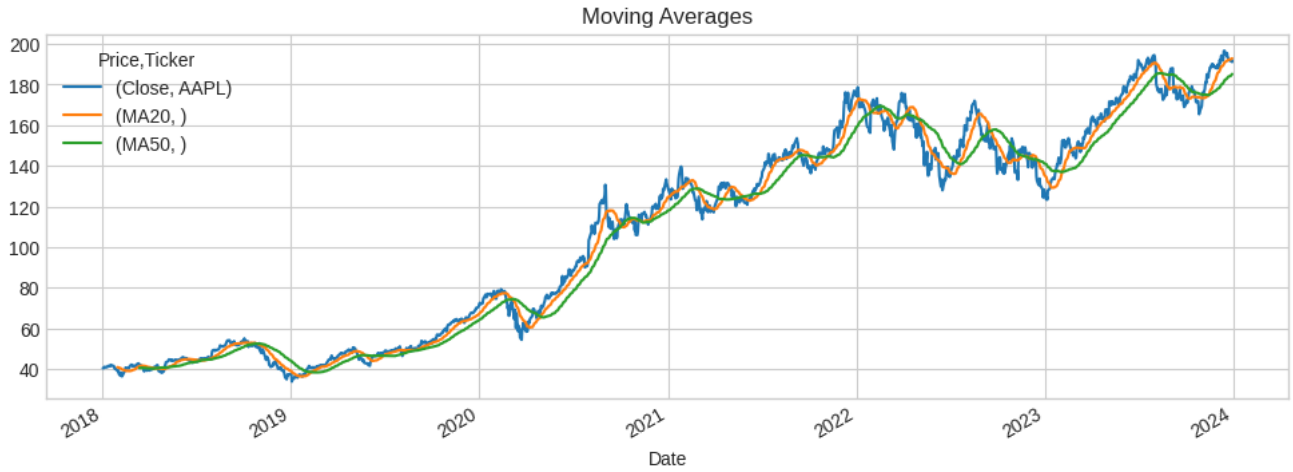




40
20
0
40
60
80
100
120
140
160
180
200


Boxplot - dadeh haye part
1e8
4
3
2
1
0
Open
High
Low
Close
Volume


Heatmap - hambastegi bein vizhegi ha
Price-Ticker
Close-AAPL
High-AAPL
Low-AAPL
Open-AAPL
1 1 1 1 -0.47
1 1 1 1 -0.46
1 1 1 1 -0.48
1 1 1 1 -0.47
-0.47 -0.46 -0.48 -0.47 1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2



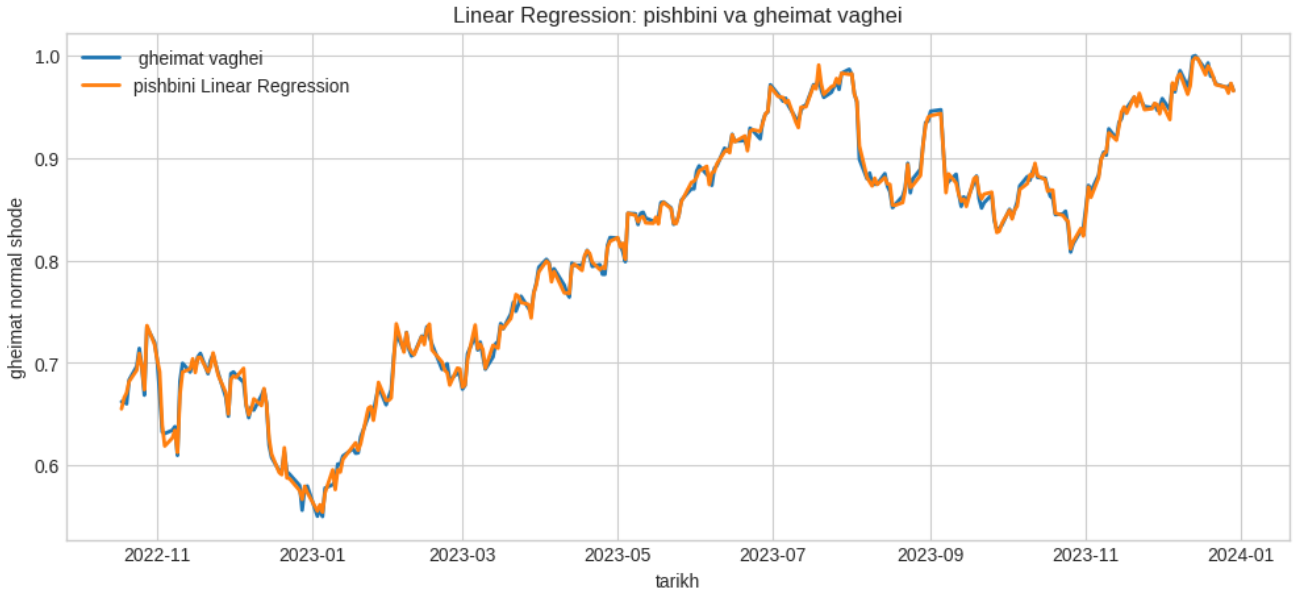




```
X = scaled_df[['Open', 'High', 'Low', 'Volume']]
y = scaled_df['Close']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)
```

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

lr_model = LinearRegression()
```

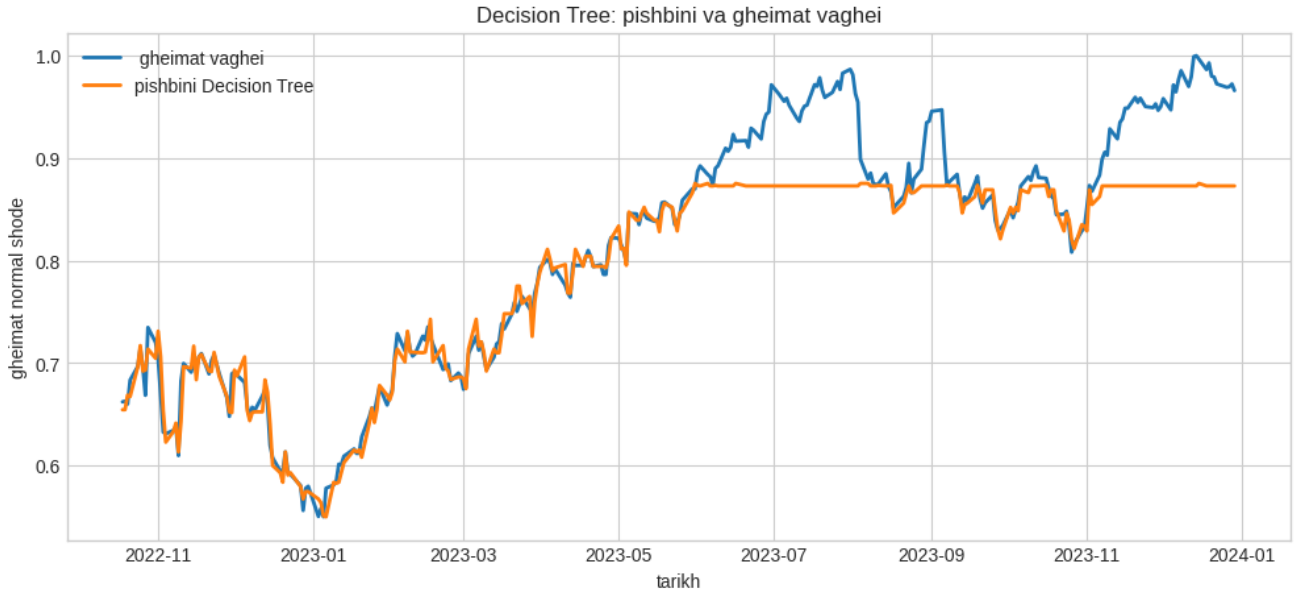
```
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
lr_pred = lr_model.predict(X_test)
lr_mae = mean_absolute_error(y_test, lr_pred)
lr_rmse = np.sqrt(mean_squared_error(y_test, lr_pred))
lr_r2 = r2_score(y_test, lr_pred)

print("Linear Regression")
print(f"MAE: {lr_mae:.4f}")
print(f"RMSE: {lr_rmse:.4f}")
print(f"R² Score: {lr_r2:.4f}")

# nemodare moghayese
plt.figure(figsize=(12,5))
plt.plot(y_test.index, y_test, label=' gheimat vaghei', linewidth=2)
plt.plot(y_test.index, lr_pred, label='pishbini Linear Regression', linewidth=2)
plt.title("Linear Regression: pishbini va gheimat vaghei")
plt.xlabel("tarikh")
plt.ylabel("gheimat normal shode")
plt.legend()
plt.grid(True)
plt.show()
```

```
Linear Regression
MAE: 0.0041
RMSE: 0.0052
R² Score: 0.9981
```



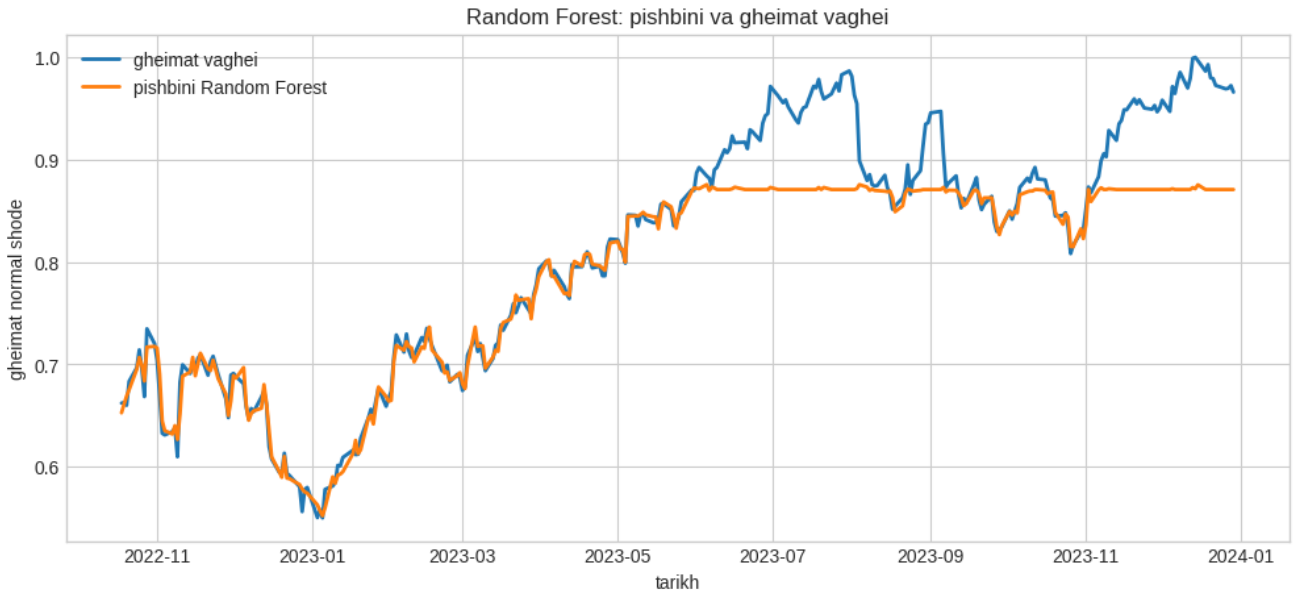
```
from sklearn.tree import DecisionTreeRegressor

dt_model = DecisionTreeRegressor()
dt_model.fit(X_train, y_train)
dt_pred = dt_model.predict(X_test)
dt_mae = mean_absolute_error(y_test, dt_pred)
dt_rmse = np.sqrt(mean_squared_error(y_test, dt_pred))
dt_r2 = r2_score(y_test, dt_pred)

print("Decision Tree")
print(f"MAE: {dt_mae:.4f}")
print(f"RMSE: {dt_rmse:.4f}")
print(f"R² Score: {dt_r2:.4f}")

plt.figure(figsize=(12,5))
plt.plot(y_test.index, y_test, label=' gheimat vaghei', linewidth=2)
plt.plot(y_test.index, dt_pred, label='pishbini Decision Tree', linewidth=2)
plt.title("Decision Tree: pishbini va gheimat vaghei")
plt.xlabel("tarikh")
plt.ylabel("gheimat normal shode")
plt.legend()
plt.grid(True)
plt.show()
```

```
Decision Tree
MAE: 0.0269
RMSE: 0.0438
R² Score: 0.8674
```



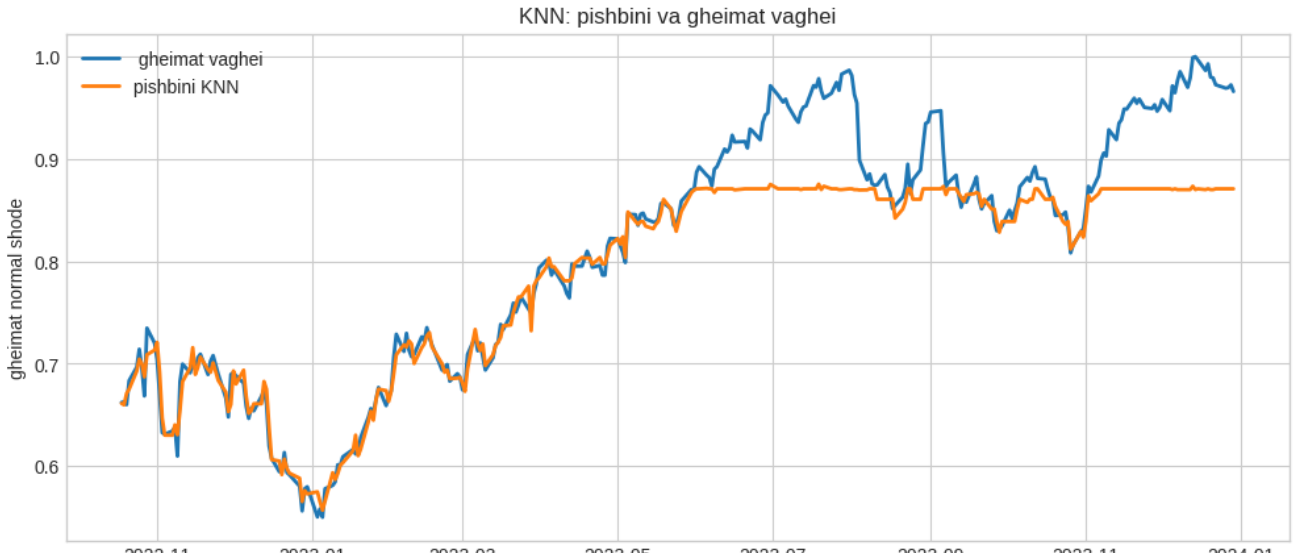
```
from sklearn.ensemble import RandomForestRegressor

rf_model = RandomForestRegressor(n_estimators=100)
rf_model.fit(X_train, y_train)
rf_pred = rf_model.predict(X_test)
rf_mae = mean_absolute_error(y_test, rf_pred)
rf_rmse = np.sqrt(mean_squared_error(y_test, rf_pred))
rf_r2 = r2_score(y_test, rf_pred)

print("Random Forest")
print(f"MAE: {rf_mae:.4f}")
print(f"RMSE: {rf_rmse:.4f}")
print(f"R² Score: {rf_r2:.4f}")

plt.figure(figsize=(12,5))
plt.plot(y_test.index, y_test, label='gheimat vaghei ', linewidth=2)
plt.plot(y_test.index, rf_pred, label='pishbini Random Forest', linewidth=2)
plt.title("Random Forest: pishbini va gheimat vaghei")
plt.xlabel("tarikh")
plt.ylabel("gheimat normal shode")
plt.legend()
plt.grid(True)
plt.show()
```

```
Random Forest
MAE: 0.0265
RMSE: 0.0443
R² Score: 0.8641
```



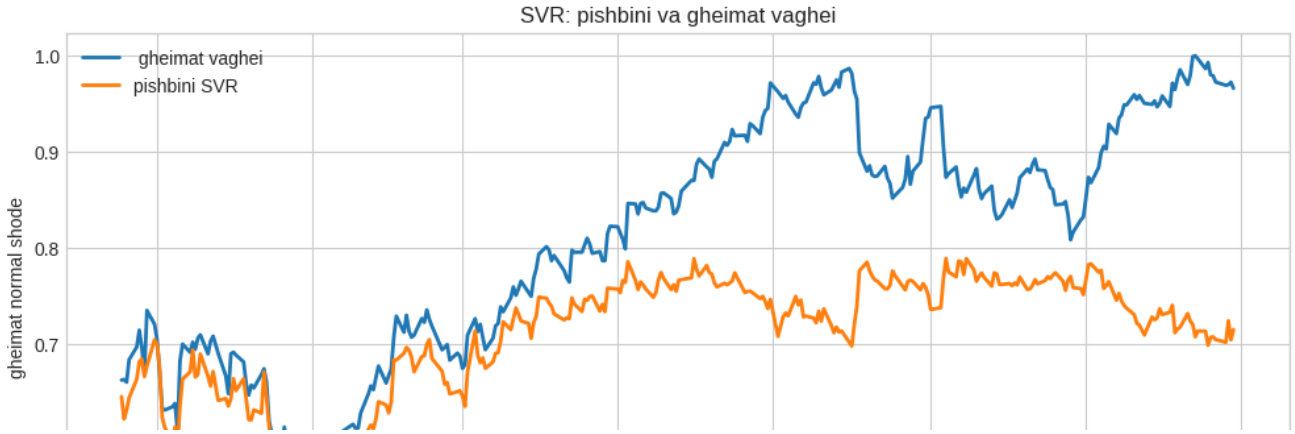
```
from sklearn.neighbors import KNeighborsRegressor
knn_model = KNeighborsRegressor()
knn_model.fit(X_train, y_train)
knn_pred = knn_model.predict(X_test)
knn_mae = mean_absolute_error(y_test, knn_pred)
knn_rmse = np.sqrt(mean_squared_error(y_test, knn_pred))
knn_r2 = r2_score(y_test, knn_pred)

print("KNN")
print(f"MAE: {knn_mae:.4f}")
print(f"RMSE: {knn_rmse:.4f}")
print(f"R² Score: {knn_r2:.4f}")

plt.figure(figsize=(12,5))
plt.plot(y_test.index, y_test, label=' gheimat vaghei', linewidth=2)
plt.plot(y_test.index, knn_pred, label='pishbini KNN', linewidth=2)
plt.title("KNN: pishbini va gheimat vaghei")
plt.xlabel("tarikh")
plt.ylabel("gheimat normal shode")
plt.legend()
plt.grid(True)
plt.show()
```

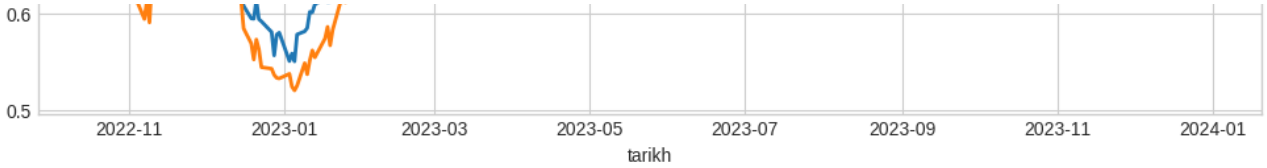
```
KNN
MAE: 0.0279
RMSE: 0.0447
R² Score: 0.8614
```




```
from sklearn.svm import SVR

svr_model = SVR()
svr_model.fit(X_train, y_train)
svr_pred = svr_model.predict(X_test)
svr_mae = mean_absolute_error(y_test, svr_pred)
svr_rmse = np.sqrt(mean_squared_error(y_test, svr_pred))
svr_r2 = r2_score(y_test, svr_pred)

print("SVR")
print(f"MAE: {svr_mae:.4f}")
print(f"RMSE: {svr_rmse:.4f}")
print(f"R² Score: {svr_r2:.4f}")

plt.figure(figsize=(12,5))
plt.plot(y_test.index, y_test, label=' gheimat vaghei', linewidth=2)
plt.plot(y_test.index, svr_pred, label='pishbini SVR', linewidth=2)
plt.title("SVR: pishbini va gheimat vaghei")
plt.xlabel("tarikh")
plt.ylabel("gheimat normal shode")
plt.legend()
plt.grid(True)
plt.show()
```

```
SVR
MAE: 0.1004
RMSE: 0.1284
R² Score: -0.1406
```



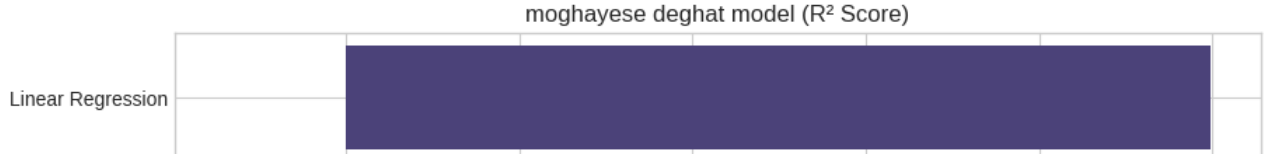


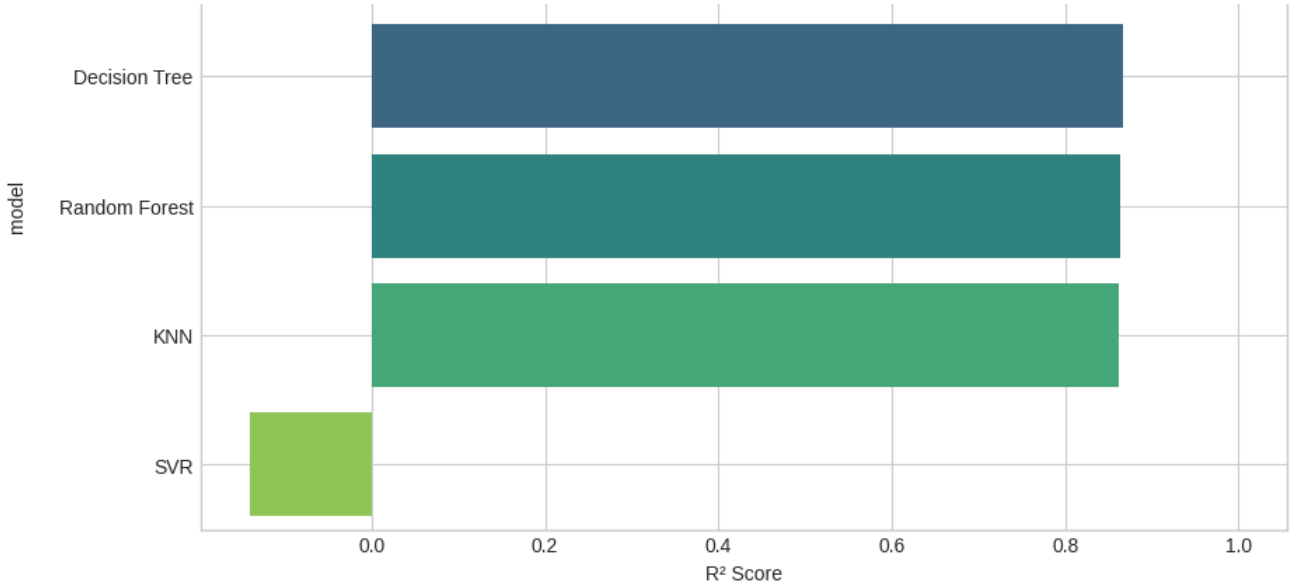

```
comparison_df = pd.DataFrame({ 'Model': ['Linear Regression', 'Decision Tree', 'Random Fo
                               'MAE': [lr_mae, dt_mae, rf_mae, knn_mae, svr_mae],
                               'RMSE': [lr_rmse, dt_rmse, rf_rmse, knn_rmse, svr_rmse],
                                'R² Score': [lr_r2, dt_r2, rf_r2, knn_r2, svr_r2]
                                })

comparison_df = comparison_df.sort_values(by='R² Score', ascending=False)
print(" moghayese amalkarde model ha:")
print(comparison_df)
```

```
 moghayese amalkarde model ha:
               Model       MAE      RMSE  R² Score
0  Linear Regression  0.004136  0.005223  0.998111
1      Decision Tree  0.026852  0.043766  0.867408
2      Random Forest  0.026480  0.044304  0.864124
3                KNN  0.027933  0.044749  0.861379
4                SVR  0.100394  0.128364 -0.140621
```

```
plt.figure(figsize=(10,6))
sns.barplot(data=comparison_df, x='R² Score', y='Model', palette='viridis')
plt.title("moghayese deghat model (R² Score)")
plt.xlabel("R² Score")
plt.ylabel("model")
plt.grid(True)
plt.show()
```

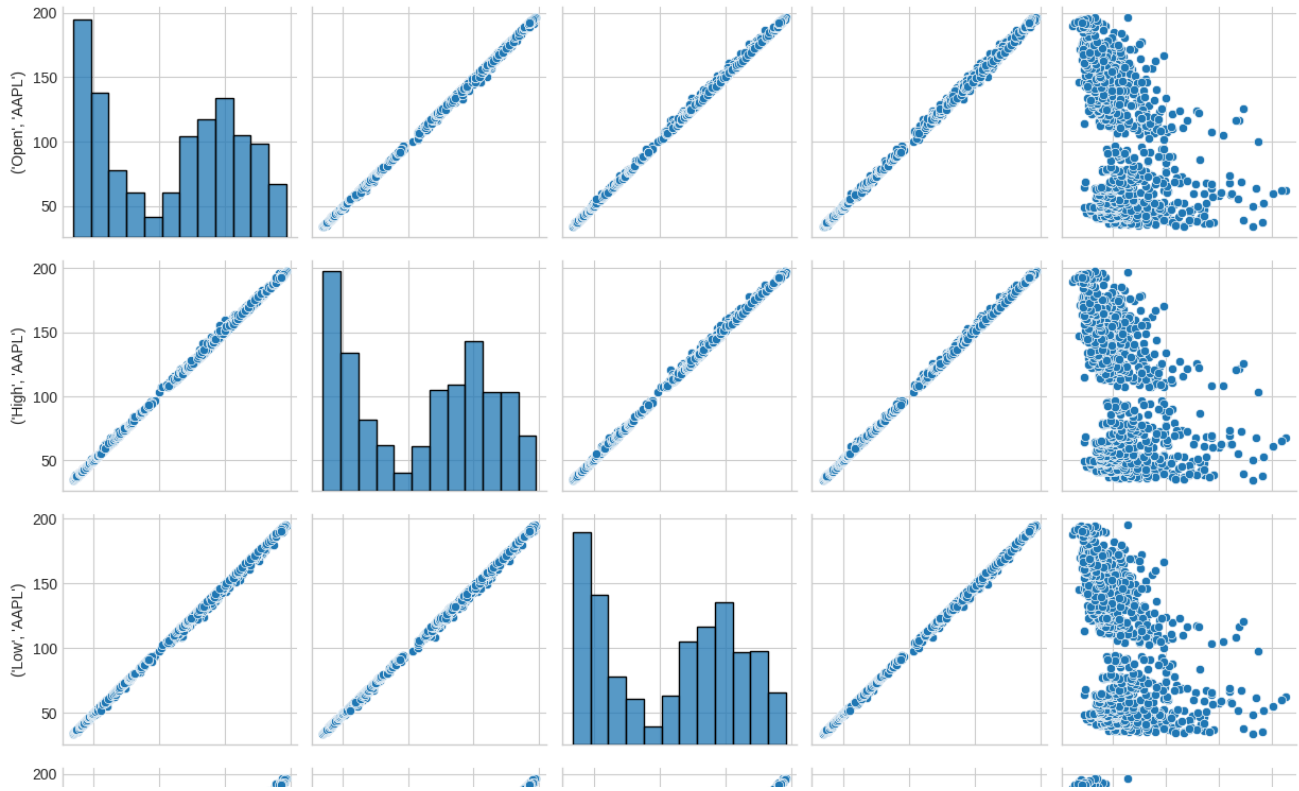
```
<ipython-input-16-8a7236082465>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.

  sns.barplot(data=comparison_df, x='R² Score', y='Model', palette='viridis')
```

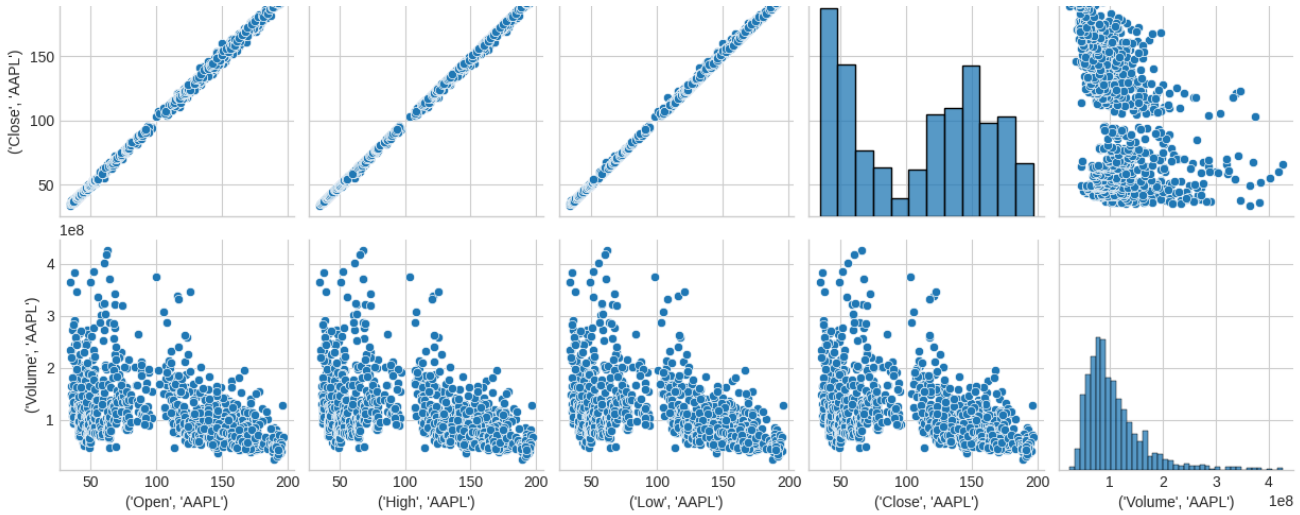





```
import seaborn as sns
sns.pairplot(df[['Open', 'High', 'Low', 'Close', 'Volume']])
```

```
<seaborn.axisgrid.PairGrid at 0x7d6511203910>
```




('Close', 'AAPL')
('Volume', 'AAPL')
1e8
('Open', 'AAPL')
('High', 'AAPL')
('Low', 'AAPL')
('Close', 'AAPL')
('Volume', 'AAPL')
1e8