

Real-Time Investment Decision System

1. Install Required Packages

```
pip install yfinance pandas

Requirement already satisfied: yfinance in /usr/local/lib/python3.11/dist-packages (0.2.65)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.0.2)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.11/dist-packages (from yfinance) (0.0.12)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.3.8)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2025.2)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.11/dist-packages (from yfinance) (3.18.2)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.13.4)
Requirement already satisfied: curl_cffi>=0.7 in /usr/local/lib/python3.11/dist-packages (from yfinance) (0.12.0)
Requirement already satisfied: protobuf>=3.19.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (5.29.5)
Requirement already satisfied: websockets>=13.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (15.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.2)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.7)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (4.14.1)
Requirement already satisfied: cffi>=1.12.0 in /usr/local/lib/python3.11/dist-packages (from curl_cffi>=0.7->yfinance) (1.17.1)
Requirement already satisfied: certifi>=2024.2.2 in /usr/local/lib/python3.11/dist-packages (from curl_cffi>=0.7->yfinance) (2025.8.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (2.5.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.12.0->curl_cffi>=0.7->yfinance) (2.22)
```

```
pip install ta
```

```
Collecting ta
  Downloading ta-0.11.0.tar.gz (25 kB)
    Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from ta) (2.0.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from ta) (2.2.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->ta) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->ta) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->ta) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (2.5.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>=1.12.0->curl_cffi>=0.7->yfinance) (2.22)

Building wheels for collected packages: ta
  Building wheel for ta (setup.py) ... done
  Created wheel for ta: filename=ta-0.11.0-py3-none-any.whl size=29412 sha256=64965d8cf7af07d655ff2c345d0ef592511cdb06619715820adc7bf8b590ddc8
  Stored in directory: /root/.cache/pip/wheels/a1/d7/29/7781cc5eb9a3659d032d7d15bdd0f49d07d2b24fec29f44bc4
Successfully built ta
Installing collected packages: ta
  Successfully installed ta-0.11.0
```

```
import yfinance as yf
import pandas as pd
from ta import add_all_ta_features
```

2. Fetch stock data

```
def fetch_stock_data(ticker_symbol):
    # Fetch historical data (last 3 months)
    ticker = yf.Ticker(ticker_symbol)
    df = ticker.history(period="3mo", interval="1d")

    # Drop rows with missing values
    df.dropna(inplace=True)

    # Add technical indicators (MACD, RSI, EMA, SMA, etc.)
    df = add_all_ta_features(
        df,
        open="Open", high="High", low="Low",
        close="Close", volume="Volume",
        fillna=True
    )

    # Fetch fundamental data
    info = ticker.info
    summary = {
        "Company": info.get("longName", "N/A"),
        "Sector": info.get("sector", "N/A"),
        "Market Cap": info.get("marketCap", "N/A"),
        "PE Ratio (TTM)": info.get("trailingPE", "N/A"),
        "Forward PE": info.get("forwardPE", "N/A"),
        "Price to Book": info.get("priceToBook", "N/A"),
        "EPS": info.get("trailingEps", "N/A"),
        "Beta": info.get("beta", "N/A"),
        "Summary": info.get("longBusinessSummary", "N/A")
    }

    return df, summary
```

```
ticker_symbol = "MSFT"
df_with_indicators, company_info = fetch_stock_data(ticker_symbol)

# Show data
print("Technical + Price Data:")
print(df_with_indicators.tail())

print("\nCompany Fundamentals:")
for key, value in company_info.items():
    print(f"{key}: {value}")
```

```
2025-08-05 00:00:00-04:00 19171600 0.0 0.0 1.202108e+08
2025-08-06 00:00:00-04:00 21355700 0.0 0.0 1.039225e+08
2025-08-07 00:00:00-04:00 16056600 0.0 0.0 9.788989e+07
2025-08-08 00:00:00-04:00 5401185 0.0 0.0 9.262760e+07
```

```
volume_olv volume_cmf ... momentum_ppo \
Date ...
2025-08-04 00:00:00-04:00 388915700 -0.054039 ... 2.036492
2025-08-05 00:00:00-04:00 369744100 -0.103559 ... 2.035744
2025-08-06 00:00:00-04:00 348388400 -0.147222 ... 1.968054
2025-08-07 00:00:00-04:00 332331800 -0.167383 ... 1.829343
2025-08-08 00:00:00-04:00 326930615 -0.208335 ... 1.688567
```

```
momentum_ppo_signal momentum_ppo_hist \
Date ...
2025-08-04 00:00:00-04:00 1.801793 0.234699
2025-08-05 00:00:00-04:00 1.848583 0.187161
2025-08-06 00:00:00-04:00 1.872477 0.095577
2025-08-07 00:00:00-04:00 1.863850 -0.034507
2025-08-08 00:00:00-04:00 1.828794 -0.140226
```

```
momentum_pvo momentum_pvo_signal \
Date ...
2025-08-04 00:00:00-04:00 13.267236 4.263749
2025-08-05 00:00:00-04:00 10.479614 5.506922
2025-08-06 00:00:00-04:00 8.912525 6.188043
2025-08-07 00:00:00-04:00 5.699140 6.090262
2025-08-08 00:00:00-04:00 -1.151662 4.641878
```

```
momentum_pvo_hist momentum_kama others_dr \
Date ...
2025-08-04 00:00:00-04:00 9.003487 514.441478 2.199926
2025-08-05 00:00:00-04:00 4.972692 515.575541 -1.473807
2025-08-06 00:00:00-04:00 2.724482 516.166055 -0.532449
2025-08-07 00:00:00-04:00 -0.391122 516.290578 -0.781037
2025-08-08 00:00:00-04:00 -5.793539 516.355337 -0.140166
```

```
others_dlr others_cr \
Date ...
2025-08-04 00:00:00-04:00 2.176076 22.469216
2025-08-05 00:00:00-04:00 -1.483963 20.665236
2025-08-06 00:00:00-04:00 -0.533871 20.022755
2025-08-07 00:00:00-04:00 -0.784103 19.085333
2025-08-08 00:00:00-04:00 -0.140264 18.918416
```

[5 rows x 93 columns]

```
Company Fundamentals:
Company: Microsoft Corporation
Sector: Technology
Market Cap: 3866065895424
PE Ratio (TTM): 38.1033
Forward PE: 34.789967
Price to Book: 11.256818
EPS: 13.65
Beta: 1.055
Summary: Microsoft Corporation develops and supports software, services, devices, and solutions worldwide. The company's Productivity and Business Processes segment offers Microsoft
```

3. Adding Target Labels to Data

```
def add_investment_label(df, threshold=0.05, window=15):
    """
    Add a binary label: 1 = Invest, 0 = Don't Invest
    Label is 1 if price increases by more than `threshold` in next `window` days
    """
    df = df.copy()
    df['future_close'] = df['Close'].shift(-window)
    df['future_return'] = (df['future_close'] - df['Close']) / df['Close']
    df['Label'] = (df['future_return'] > threshold).astype(int)
    df.dropna(inplace=True)
    return df
```

labeled_df = add_investment_label(df_with_indicators)

```
# Checking it out
print(labeled_df[['Close', 'future_close', 'future_return', 'Label']].tail())
```

Date	Close	future_close	future_return	Label
2025-07-14 00:00:00-04:00	503.019989	535.640015	0.064848	1
2025-07-15 00:00:00-04:00	505.820007	527.750000	0.043355	0
2025-07-16 00:00:00-04:00	505.619995	524.940002	0.038211	0
2025-07-17 00:00:00-04:00	511.700012	520.840027	0.017862	0
2025-07-18 00:00:00-04:00	510.049988	520.109985	0.019724	0

4. Installing packages for model training

pip install xgboost scikit-learn

```
Requirement already satisfied: xgboost in /usr/local/lib/python3.11/dist-packages (3.0.3)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (1.6.1)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from xgboost) (2.0.2)
Requirement already satisfied: nvidia-ncc1-cu12 in /usr/local/lib/python3.11/dist-packages (from xgboost) (2.23.4)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from xgboost) (1.16.1)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn) (3.6.0)
```

```
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

5. Training Data

```
def train_model(data):
    # Drop unused columns
    data = data.drop(columns=["Open", "High", "Low", "Volume", "Dividends", "Stock Splits",
                             "future_close", "future_return"], errors='ignore')

    # Drop rows with missing values (from indicators)
    data = data.dropna()

    # Split features & target
    X = data.drop(columns=["Label"])
    y = data["Label"]

    # Train-test split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Train XGBoost model
    model = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss')
    model.fit(X_train, y_train)

    # Evaluate
    y_pred = model.predict(X_test)
    print("Classification Report:\n", classification_report(y_test, y_pred))
    print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
    print("Accuracy Score:", accuracy_score(y_test, y_pred))

    return model, X_train.columns

#example
model, features_used = train_model(labeled_df)
```

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.80	0.73	5
1	0.75	0.60	0.67	5
accuracy			0.70	10
macro avg	0.71	0.70	0.70	10
weighted avg	0.71	0.70	0.70	10

Confusion Matrix:

```
[4 1]
[2 3]
```

Accuracy Score: 0.7

/usr/local/lib/python3.11/dist-packages/xgboost/training.py:183: UserWarning: [15:38:55] WARNING: /workspace/src/learner.cc:738: Parameters: { "use_label_encoder" } are not used.

```
bst.update(dtrain, iteration=i, fobj=obj)
```

6. Installing packages for Model Explainability using SHAP

```
pip install shap
```

Requirement already satisfied: shap in /usr/local/lib/python3.11/dist-packages (0.48.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from shap) (2.0.2)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from shap) (1.16.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/dist-packages (from shap) (1.6.1)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from shap) (2.2.2)
Requirement already satisfied: tqdm>=4.27.0 in /usr/local/lib/python3.11/dist-packages (from shap) (4.27.1)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.11/dist-packages (from shap) (25.0)
Requirement already satisfied: slicer==0.0.8 in /usr/local/lib/python3.11/dist-packages (from shap) (0.0.8)
Requirement already satisfied: numba>=0.54 in /usr/local/lib/python3.11/dist-packages (from shap) (0.60.0)
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.11/dist-packages (from shap) (3.1.1)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.11/dist-packages (from shap) (4.14.1)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.11/dist-packages (from numba>=0.54->shap) (0.43.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>shap) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas>shap) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>shap) (2025.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>shap) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>shap) (3.6.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas>shap) (1.17.0)

```
import shap
```

7. Shap Visualization

```
def explain_model_with_shap(model, X_train):
    # Use TreeExplainer for XGBoost
    explainer = shap.Explainer(model, X_train)
    shap_values = explainer(X_train)

    # Summary plot (feature importance)
    shap.summary_plot(shap_values, X_train)

    # Force plot for one instance (e.g., index 10)
    shap.plots.force(shap_values[10])
```

```
# After training
model, features = train_model(labeled_df)

# Get feature DataFrame
X = labeled_df[features].dropna()

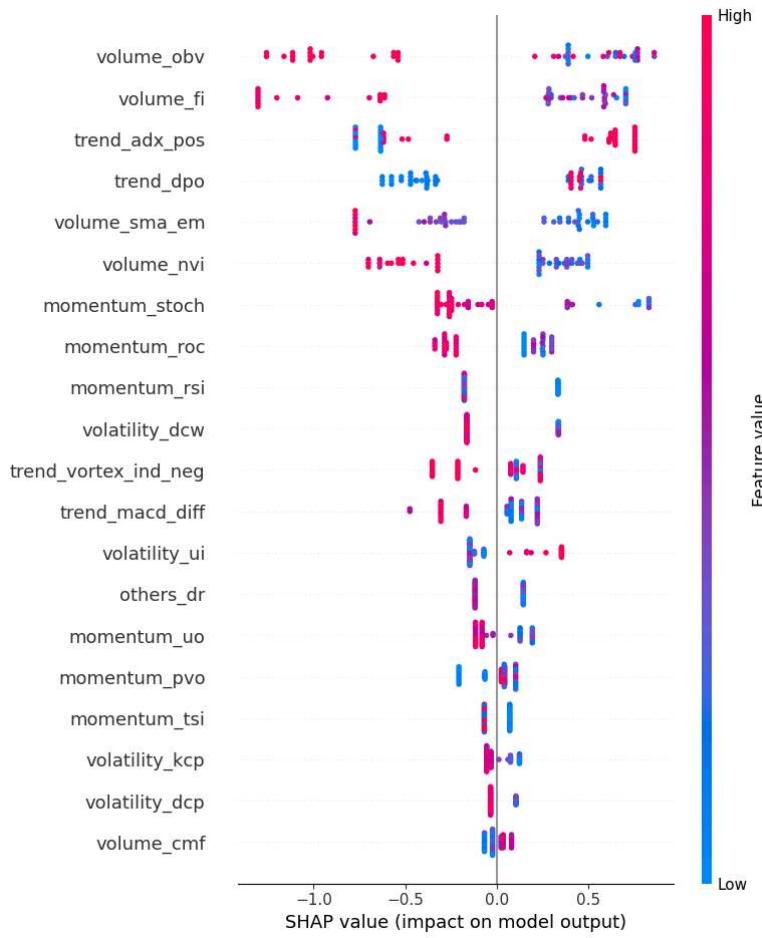
# Run SHAP explainability
explain_model_with_shap(model, X)

/usr/local/lib/python3.11/dist-packages/xgboost/training.py:183: UserWarning: [15:39:11] WARNING: /workspace/src/learner.cc:738:
Parameters: { "use_label_encoder" } are not used.
```

Classification Report:

	precision	recall	f1-score	support
0	0.67	0.80	0.73	5
1	0.75	0.60	0.67	5
accuracy			0.70	10
macro avg	0.71	0.70	0.70	10
weighted avg	0.71	0.70	0.70	10

Confusion Matrix:
[[4 1]
 [2 3]]
 Accuracy Score: 0.7



8. Installing libraries for Building streamline app

```
pip install streamlit
```

```
Collecting streamlit
  Downloading streamlit-1.48.0-py3-none-any.whl.metadata (9.5 kB)
Requirement already satisfied: altair!=5.4.0,!=5.4.1,<6,>=4.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (5.5.0)
Requirement already satisfied: blinker<2,>1.5.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (1.9.0)
Requirement already satisfied: cachetools<7,>4.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (5.5.2)
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (8.2.1)
Requirement already satisfied: pandas<3,>=1.4.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (2.2.2)
Requirement already satisfied: packaging<26,>=20 in /usr/local/lib/python3.11/dist-packages (from streamlit) (25.0)
Requirement already satisfied: pillow<12,>=7.1.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (11.3.0)
Requirement already satisfied: protobuf<7,>=3.20 in /usr/local/lib/python3.11/dist-packages (from streamlit) (5.29.5)
Requirement already satisfied: pyarrow<7.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (18.1.0)
Requirement already satisfied: requests<3,>=2.27 in /usr/local/lib/python3.11/dist-packages (from streamlit) (2.32.3)
Requirement already satisfied: tenacity<10,>=1.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (8.5.0)
Requirement already satisfied: toml<2,>=0.10.1 in /usr/local/lib/python3.11/dist-packages (from streamlit) (0.10.2)
Requirement already satisfied: typing-extensions<5,>=4.4.0 in /usr/local/lib/python3.11/dist-packages (from streamlit) (4.14.1)
Collecting watchdog<7,>=2.1.5 (from streamlit)
  Downloading watchdog-2.1.5-py3-none-manylinux2014_x86_64.whl.metadata (44 kB)
  44.3/44.3 kB 2.9 MB/s eta 0:00:00
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in /usr/local/lib/python3.11/dist-packages (from streamlit) (3.1.45)
Collecting pydeck<1,>=0.8.0b4 (from streamlit)
```

```

Downloading pydeck-0.9.1-py2.py3-none-any.whl.metadata (4.1 kB)
Requirement already satisfied: tornado!=6.5.0,<7,>=6.0.3 in /usr/local/lib/python3.11/dist-packages (from streamlit) (6.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (3.1.6)
Requirement already satisfied: jsonschema>=3.0 in /usr/local/lib/python3.11/dist-packages (from altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (4.25.0)
Requirement already satisfied: narwhals>=1.14.2 in /usr/local/lib/python3.11/dist-packages (from altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (2.0.1)
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.11/dist-packages (from gitpython!=3.1.19,<4,>=3.0.7->streamlit) (4.0.12)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas<3,>=1.4.0->streamlit) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas<3,>=1.4.0->streamlit) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas<3,>=1.4.0->streamlit) (2025.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.27->streamlit) (3.4.2)
Requirement already satisfied: idna<4,>=3.0.5 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.27->streamlit) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.27->streamlit) (2.5.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2.27->streamlit) (2025.8.3)
Requirement already satisfied: smmap<6,>=3.0.1 in /usr/local/lib/python3.11/dist-packages (from gitdb<5,>=4.0.1->gitpython!=3.1.19,<4,>=3.0.7->streamlit) (5.0.2)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2>altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (3.0.2)
Requirement already satisfied: attrs>=22.2.0 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (25.3.0)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (2025.4)
Requirement already satisfied: referencing>=0.28.4 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (0.36.2)
Requirement already satisfied: rpdspy>=0.7.1 in /usr/local/lib/python3.11/dist-packages (from jsonschema>=3.0->altair!=5.4.0,!5.4.1,<6,>=4.0->streamlit) (0.26.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas<3,>=1.4.0->streamlit) (1.17.0)
Successfully installed pydeck-0.9.1 streamlit-1.48.0 watchdog-6.0.0
Downloaded streamlit-1.48.0-py3-none-any.whl (9.9 MB)
  9.9/9.9 MB 83.5 MB/s eta 0:00:00
Downloading pydeck-0.9.1-py2.py3-none-any.whl (6.9 MB)
  6.9/6.9 MB 141.1 MB/s eta 0:00:00
Downloading watchdog-6.0.0-py3-none-manylinux2014_x86_64.whl (79 kB)
  79.1/79.1 kB 6.7 MB/s eta 0:00:00
Installing collected packages: watchdog, pydeck, streamlit
Successfully installed pydeck-0.9.1 streamlit-1.48.0 watchdog-6.0.0

pip install pandas-ta==0.3.14b0

→ Collecting pandas-ta==0.3.14b0
  Downloading pandas_ta-0.3.14b0.tar.gz (115 kB)
    115.1/115.1 kB 6.4 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from pandas-ta==0.3.14b0) (2.2.2)
Requirement already satisfied: numpy>=1.23.2 in /usr/local/lib/python3.11/dist-packages (from pandas->pandas-ta==0.3.14b0) (2.0.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->pandas-ta==0.3.14b0) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->pandas-ta==0.3.14b0) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->pandas-ta==0.3.14b0) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->pandas-ta==0.3.14b0) (1.17.0)
Building wheels for collected packages: pandas-ta
  Building wheel for pandas-ta (setup.py) ... done
  Created wheel for pandas-ta: filename=pandas_ta-0.3.14b0-py3-none-any.whl size=218910 sha256=22ad6ec6bc9b3eba2b591deae5698af9d5d839c9c6bebbf4fa2f7238931087
  Stored in directory: /root/.cache/pip/wheels/7f/33/8b/5b0245c5c65433cd8f5cb24ac15d97e5a3db2d41a8b6ae957d
Successfully built pandas-ta
Installing collected packages: pandas-ta
Successfully installed pandas-ta-0.3.14b0

```

Downloading squeeze pro code manually as it is not available in pandas ta new version or can say that its been deleted in new version.

```

squeeze_pro_code = """
from pandas_ta import overlap, volatility
from pandas_ta.utils import get_drift, get_offset, verify_series

def squeeze_pro(high, low, close, length=None, bb=None, kc=None, mamode=None, mult=None, drift=None, offset=None, **kwargs):
    # Validate
    high = verify_series(high)
    low = verify_series(low)
    close = verify_series(close)
    length = int(length) if length and length > 0 else 20
    bb = float(bb) if bb and bb > 0 else 2.0
    kc = float(kc) if kc and kc > 0 else 1.5
    mamode = mamode if mamode else "sma"
    mult = float(mult) if mult and mult > 0 else 1.5
    drift = get_drift(drift)
    offset = get_offset(offset)

    # Calculate
    basis = overlap.ma(mamode, close=close, length=length)
    dev = bb * close.rolling(length).std()
    upper_bb = basis + dev
    lower_bb = basis - dev

    range_high = high.rolling(length).max()
    range_low = low.rolling(length).min()
    range_ = range_high - range_low
    ma_range = overlap.ma(mamode, close=range_, length=length)
    upper_kc = basis + ma_range * kc
    lower_kc = basis - ma_range * kc

    squeeze_on = (lower_bb > lower_kc) & (upper_bb < upper_kc)
    squeeze_off = (lower_bb < lower_kc) & (upper_bb > upper_kc)

    # Result
    squeeze_on.name = f"SQUEEZE_ON_{length}"
    squeeze_off.name = f"SQUEEZE_OFF_{length}"
    squeeze_on.category = squeeze_off.category = "momentum"

    return squeeze_on, squeeze_off
"""

# Save to a file
with open('/usr/local/lib/python3.11/dist-packages/pandas_ta/momentum/squeeze_pro.py', "w") as f:
    f.write(squeeze_pro_code)

```

```
from pandas_ta.momentum.squeeze_pro import squeeze_pro
```

```

import pickle
import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split

data = pd.DataFrame({
    'P/E Ratio': [10, 25, 30, 8, 15],
    'Debt/Equity': [0.5, 2.1, 1.2, 0.3, 1.0],
    'ROE': [12, 7, 5, 15, 10],
    'Sentiment_Score': [0.8, 0.3, 0.4, 0.9, 0.6],
    'Invest': [1, 0, 0, 1, 1]
})

X = data.drop('Invest', axis=1)
y = data['Invest']

with open('features.pkl', 'wb') as f:
    pickle.dump(list(X.columns), f)

# Train model
model = RandomForestClassifier()
model.fit(X, y)

```

RandomForestClassifier [?](#)

```

# Save model
with open('investment_model.pkl', 'wb') as f:
    pickle.dump(model, f)

print("Model and feature list saved.")

```

Model and feature list saved.

```

import streamlit as st
import pandas as pd
import pickle
import numpy as np

```

```

# Load model and features
@st.cache_data
def load_model():
    model = pickle.load(open("investment_model.pkl", "rb"))
    features = pickle.load(open("features.pkl", "rb"))
    return model, features

model, feature_names = load_model()

```

2025-08-08 15:39:54.998 WARNING streamlit.runtime.caching.cache_data_api: No runtime found, using MemoryCacheStorageManager
 2025-08-08 15:39:55.000 WARNING streamlit.runtime.caching.cache_data_api: No runtime found, using MemoryCacheStorageManager
 2025-08-08 15:39:55.001 WARNING streamlit.runtime.scriptrunner_utils.script_run_context: Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode
 2025-08-08 15:39:55.165 Warning: to view this Streamlit app on a browser, run it with the following command:
 streamlit run /usr/local/lib/python3.11/dist-packages/colab_kernel_launcher.py [ARGUMENTS]

2025-08-08 15:39:55.167 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:55.168 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:55.169 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:55.170 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:55.171 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:55.172 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:55.173 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.

```

st.title("Real-Time Investment Decision App")
st.write("Predict whether to invest in a firm based on its fundamental, technical, and sentiment metrics.")

```

2025-08-08 15:39:57.960 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:57.961 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:57.962 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:57.963 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:39:57.964 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.

```

# Create input fields dynamically based on feature list
user_input = {}
st.subheader("Input Firm Metrics")
for feature in feature_names:
    user_input[feature] = st.number_input(f"{feature}", value=0.0)

```

2025-08-08 15:40:00.595 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:40:00.596 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:40:00.597 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:40:00.598 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:40:00.599 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:40:00.600 Session state does not function when running a script without 'streamlit run'
 2025-08-08 15:40:00.601 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
 2025-08-08 15:40:00.602 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.

```
2025-08-08 15:40:00.601 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.601 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.602 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.602 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.603 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.603 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.604 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.604 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.605 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.605 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.606 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.607 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.607 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.608 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.608 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.609 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.609 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.610 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.610 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.611 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:00.611 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
```

```
# Convert input to DataFrame
input_df = pd.DataFrame([user_input])
```

```
# Prediction
if st.button("Predict Investment Decision"):
    prediction = model.predict(input_df)[0]
    proba = model.predict_proba(input_df)[0]

st.subheader("Prediction")
if prediction == 1:
    st.success("✅ Recommended to INVEST in this firm.")
else:
    st.error("❌ NOT Recommended to invest at this moment.")
```

```
st.write("Confidence Score:")
st.write(f"Invest: {proba[1]*100:.2f}% | Don't Invest: {proba[0]*100:.2f}%")
```

```
2025-08-08 15:40:03.877 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:03.879 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:03.879 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:03.880 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:03.881 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
2025-08-08 15:40:03.881 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
```

```
# Load model and features
with open("investment_model.pkl", "rb") as f:
    model = pickle.load(f)
```

```
with open("features.pkl", "rb") as f:
    feature_names = pickle.load(f)
```

```
# Sample real-time input
real_time_input = {
    'PE Ratio': 15.0,
    'ROE': 18.5,
    'Debt to Equity': 0.4,
    'Current Ratio': 1.8,
    'MACD': 1.2,
    'RSI': 55,
    'Sentiment Score': 0.7
}
```

```
# Convert to DataFrame
input_df = pd.DataFrame([real_time_input])
```

```
# Ensure all required features are present
for col in feature_names:
    if col not in input_df.columns:
        input_df[col] = 0 # Fill missing with 0

# Align column order
input_df = input_df[feature_names]

# Make prediction
prediction = model.predict(input_df)[0]
```

```
# Show result
if prediction == 1:
    print("✅ Recommendation: INVEST in this company.")
else:
    print("❌ Recommendation: DO NOT INVEST in this company.")
```

```
2025-08-08 15:40:03.882 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
```

Testing the model

9. Installing libraries for real time data

```
pip install yfinance pandas-ta textblob
```

```
2025-08-08 15:40:03.883 Thread 'MainThread': missing ScriptRunContext! This warning can be ignored when running in bare mode.
Requirement already satisfied: yfinance in /usr/local/lib/python3.11/dist-packages (0.2.65)
Requirement already satisfied: pandas-ta in /usr/local/lib/python3.11/dist-packages (0.3.14b0)
Requirement already satisfied: textblob in /usr/local/lib/python3.11/dist-packages (0.19.0)
```

```

Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.2.2)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.0.2)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.32.3)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.11/dist-packages (from yfinance) (0.0.12)
Requirement already satisfied: platformdirs>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.3.8)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2025.2)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.11/dist-packages (from yfinance) (2.4.6)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.11/dist-packages (from yfinance) (3.18.2)
Requirement already satisfied: beautifulsoup4>4.11.1 in /usr/local/lib/python3.11/dist-packages (from yfinance) (4.13.4)
Requirement already satisfied: curl_cffi>0.7 in /usr/local/lib/python3.11/dist-packages (from yfinance) (0.12.0)
Requirement already satisfied: protobuf>=3.19.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (5.29.5)
Requirement already satisfied: websockets>=13.0 in /usr/local/lib/python3.11/dist-packages (from yfinance) (15.0.1)
Requirement already satisfied: nltk>=3.9 in /usr/local/lib/python3.11/dist-packages (from textblob) (3.9.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.7)
Requirement already satisfied: typing-extensions>=4.0.0 in /usr/local/lib/python3.11/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (4.14.1)
Requirement already satisfied: cffi>1.12.0 in /usr/local/lib/python3.11/dist-packages (from curl_cffi>0.7->yfinance) (1.17.1)
Requirement already satisfied: certifi>2024.2.2 in /usr/local/lib/python3.11/dist-packages (from curl_cffi>0.7->yfinance) (2025.8.3)
Requirement already satisfied: click in /usr/local/lib/python3.11/dist-packages (from nltk>=3.9->textblob) (8.2.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from nltk>=3.9->textblob) (1.5.1)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.11/dist-packages (from nltk>=3.9->textblob) (2024.11.6)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from nltk>=3.9->textblob) (4.67.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.3.0->yfinance) (2.9.0.post0)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas>=1.3.0->yfinance) (2025.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.4.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.31->yfinance) (2.5.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.11/dist-packages (from cffi>1.12.0->curl_cffi>0.7->yfinance) (2.22)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas>=1.3.0->yfinance) (1.17.0)

```

```

from textblob import TextBlob

def get_company_data(ticker='MSFT'):
    # Fetch fundamental data
    stock = yf.Ticker(ticker)
    info = stock.info

    pe_ratio = info.get('trailingPE', 0)
    roe = info.get('returnOnEquity', 0) * 100 if info.get('returnOnEquity') else 0
    debt_to_equity = info.get('debtToEquity', 0)
    current_ratio = info.get('currentRatio', 0)

    # Fetch technical indicators
    df = stock.history(period='6mo')
    df.ta.macd(append=True)
    df.ta.rsi(append=True)

    macd = df['MACD_12_26_9'].iloc[-1]
    rsi = df['RSI_14'].iloc[-1]

    # Sentiment Analysis (simulate for now)
    news = [
        "Microsoft's new product launch gets positive reviews.",
        "Microsoft stock rises after strong earnings report."
    ]
    sentiment = sum(TextBlob(article).sentiment.polarity for article in news) / len(news)

    # Combine into input dictionary
    return {
        'PE Ratio': pe_ratio,
        'ROE': roe,
        'Debt to Equity': debt_to_equity,
        'Current Ratio': current_ratio,
        'MACD': macd,
        'RSI': rsi,
        'Sentiment Score': sentiment
    }

import joblib
model = joblib.load("investment_model.pkl")
features = joblib.load("features.pkl")

def get_live_data(ticker='MSFT'):
    stock = yf.Ticker(ticker)
    info = stock.info

    pe_ratio = info.get('trailingPE', 0)
    roe = info.get('returnOnEquity', 0) * 100 if info.get('returnOnEquity') else 0
    debt_to_equity = info.get('debtToEquity', 0)
    current_ratio = info.get('currentRatio', 0)

    df = stock.history(period='6mo')
    df.ta.macd(append=True)
    df.ta.rsi(append=True)

    macd = df['MACD_12_26_9'].iloc[-1]
    rsi = df['RSI_14'].iloc[-1]

    # Simulated sentiment score from news headlines
    news = [
        "Microsoft's sees strong demand in Q3.",
        "Positive outlook for Microsoft's innovation pipeline."
    ]
    sentiment = sum(TextBlob(article).sentiment.polarity for article in news) / len(news)

    input_data = {
        'P/E Ratio': pe_ratio,
        'ROE': roe,
        'Debt/Equity': debt_to_equity,
        'Current Ratio': current_ratio,
        'MACD': macd,
        'RSI': rsi,
    }

```

```
'Sentiment_Score': sentiment
}

return input_data

def predict_investment(ticker='MSFT'):
    input_dict = get_live_data(ticker)

    # Ensure all required features are present
    data = pd.DataFrame([input_dict])[features]

    # Handle missing or infinite values if any
    data = data.replace([np.inf, -np.inf], np.nan).fillna(0)

    prediction = model.predict(data)[0]
    decision = "✅ Invest" if prediction == 1 else "❌ Do Not Invest"

    print(f"\n🔍 Investment Decision for {ticker}: {decision}")
    print(f"\n📊 Input Data:\n{data.T}")

# Example usage
predict_investment('MSFT')
```

→ Investment Decision for MSFT: ❌ Do Not Invest

	0
P/E Ratio	38.096706
Debt/Equity	32.661000
ROE	33.280998
Sentiment_Score	0.330303

10.. Visualization result

```
pip install plotly
```

→ Requirement already satisfied: plotly in /usr/local/lib/python3.11/dist-packages (5.24.1)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.11/dist-packages (from plotly) (8.5.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from plotly) (25.0)

```
import plotly.graph_objects as go

def visualize_prediction(input_data, prediction):
    # Bar chart for features
    fig, ax = plt.subplots(figsize=(10, 6))
    ax.bart(list(input_data.keys()), list(input_data.values()), color='skyblue')
    ax.set_title('Input Features for Investment Decision')
    ax.set_xlabel('Value')
    ax.grid(True)
    plt.tight_layout()
    plt.show()

    # Sentiment Score gauge
    sentiment = input_data['Sentiment_Score']
    gauge = go.Figure(go.Indicator(
        mode="gauge+number",
        value=sentiment,
        title={'text': "Sentiment Score"},
        gauge={'axis': {'range': [-1, 1]},
               'bar': {'color': "green" if sentiment > 0 else "red"}}
    ))
    gauge.show()

    # Final Prediction
    print("\n📊 Investment Recommendation:")
    if prediction == 1:
        print("✅ Invest in this firm (Positive Signal)")
    else:
        print("❌ Do NOT invest in this firm (Negative Signal)")

def predict_investment(ticker):
    input_data = get_live_data(ticker)
    input_df = pd.DataFrame([input_data])[features]

    with open('investment_model.pkl', 'rb') as f:
        model = pickle.load(f)

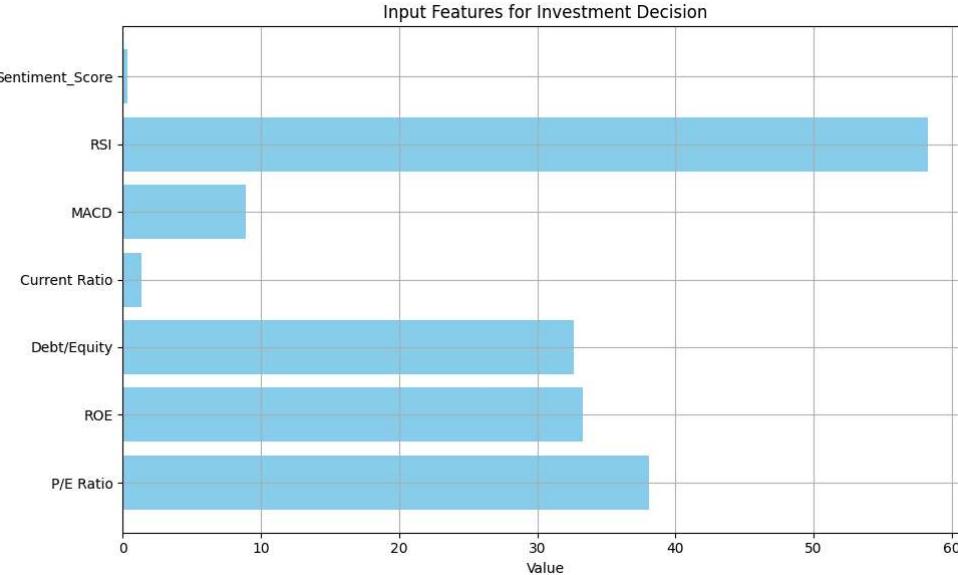
    prediction = model.predict(input_df)[0]

    visualize_prediction(input_data, prediction)

    return prediction

predict_investment('MSFT')
```

☒



■ Investment Recommendation:
✖ Do NOT invest in this firm (Negative Signal)
np.int64(0)

After seeing the result we should not invest in this company as this will not give us positive returns.

11. Testing another model

```
with open('investment_model.pkl', 'rb') as f:
    model = pickle.load(f)
```

```
# Load feature list used during training
with open('features.pkl', 'rb') as f:
    required_features = pickle.load(f)
```

```
test_data = pd.DataFrame([
    'ROE': 15.2,
    'P/E Ratio': 22.4,
    'Debt/Equity': 0.45,
    'Sentiment_Score': 0.3,
    'Profit Margin': 12.5,
    'Revenue Growth': 8.1
])
```

```
test_data.rename(columns={
    'ROE': 'Return on Equity'
}, inplace=True)
```

```
# Add missing features with default value 0
for col in required_features:
    if col not in test_data.columns:
        test_data[col] = 0
```

```
# Reorder columns
test_data = test_data[required_features]
```

```

prediction = model.predict(test_data)[0]

if prediction == 1:
    print("✅ You should INVEST in this company.")
else:
    print("❌ Do NOT invest in this company.")

➡️ ❌ Do NOT invest in this company.

import seaborn as sns

# Make prediction
prediction = model.predict(test_data)[0]

# Plot feature values
plt.figure(figsize=(10, 6))
sns.barplot(x=test_data.columns, y=test_data.iloc[0], palette="Blues_d")
plt.xticks(rotation=45, ha='right')
plt.title("Investment Decision: ('✅ INVEST' if prediction == 1 else '❌ DO NOT INVEST')", fontsize=14)
plt.xlabel("Financial & Sentiment Indicators")
plt.ylabel("Value")
plt.tight_layout()
plt.grid(True)
plt.show()

```

➡️ /tmp/ipython-input-1182490046.py:8: FutureWarning:

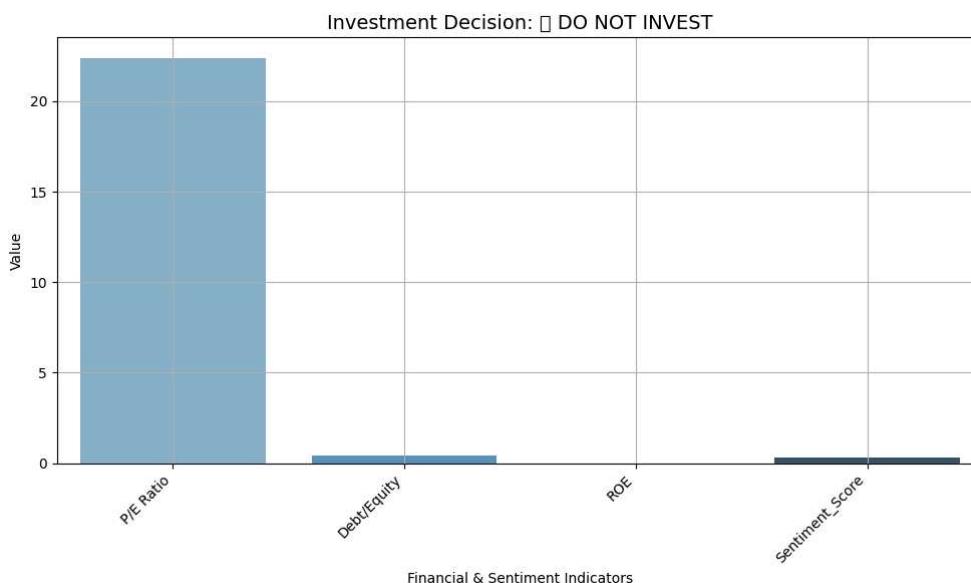
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

/tmp/ipython-input-1182490046.py:13: UserWarning:

Glyph 10060 (\N{CROSS MARK}) missing from font(s) DejaVu Sans.

/usr/local/lib/python3.11/dist-packages/IPython/core/pylabtools.py:151: UserWarning:

Glyph 10060 (\N{CROSS MARK}) missing from font(s) DejaVu Sans.



Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.