Machine Learning for Trading Strategies

Objective

- Clean and engineer features from real market data
- Design and validate ML models for forecasting or signal classification
- Evaluate performance using robust time-series methodology
- Reflect on interpretability, ethics, and modeling pitfalls unique to finance

---

Part 1: Data Collection & Preprocessing

Tasks

Download Historical Market Data

- Use yfinance to retrieve 5 years of EOD data (e.g., AAPL, MSFT, SPY)
- Include OHLCV and optionally other indicators (e.g., VIX)

Clean the Data

- Handle missing or non-trading days
- Apply forward-fill or drop logic

Smooth and Normalize

- Remove outliers using rolling z-scores
- Apply standard scaling or min-max normalization for each feature column

Deliverables

- Cleaned DataFrame with aligned date index
- Notebook section describing cleaning logic and rationale

Part 2: Feature Engineering & Selection

Tasks

Create Technical Indicators

- SMA, EMA, RSI, Bollinger Bands, MACD
- Rolling volatility, return lags (1, 5, 10-day)

Add Derived Features

- Momentum: difference of price across lags
- Binary labels: price up/down over next 5 days
- Optional: sector ETF signals or macro indicators

Explore Feature Selection

- Plot correlation heatmap
- Apply PCA
- Drop collinear or low-value predictors

Deliverables

- Feature matrix X and label vector y
- Justified selection of top 10–20 features
- Summary chart or table of engineered features

Part 3: Model Building & Training

Tasks

Train ML Models

- Regression: LinearRegression, RandomForestRegressor
- Classification: LogisticRegression, DecisionTreeClassifier

Walk-Forward Validation

- Use rolling training/testing windows (e.g., expanding window or 80/20 split every 200 days)
- Track out-of-sample predictions over time

Avoid Look-Ahead Bias

- Ensure all features use only past data
- Lag all predictors when computing forward labels

Deliverables

- Model objects and prediction outputs
- Time-series of walk-forward performance
- Commentary on any signs of overfitting

Part 4: Model Evaluation & Interpretation

Tasks

Metrics

- Classification: Accuracy, Precision, Recall, F1
- Regression: MSE, RMSE, MAE, R^2
- Include confusion matrix and ROC curve if applicable

Interpret Results

- Use feature_importances_, coefficients, or SHAP
- Comment on which features are driving signals

Portfolio Simulation (Optional)

- Apply binary predictions to build long/flat strategy
- Track hypothetical equity curve with no leverage
- Compare with SPY benchmark

Deliverables

- Evaluation table of metrics
- Plots: ROC curve, confusion matrix, prediction curve
- Interpretability summary (100–200 words)

Part 5: Unsupervised Exploration

Tasks

Apply Clustering

- Use k-means or hierarchical clustering on feature matrix
- Group stocks by behavioral similarity

Visualize Regimes

- Cluster transitions through time
- Identify periods of volatility shift or correlation clusters

Deliverables

- Cluster plots (e.g., silhouette, dendrogram)
- Short markdown reflection on insights discovered

Part 6: Natural Language Processing for Market Sentiment

Objective

Use NLP techniques to extract sentiment from financial news headlines or articles and explore how this information can be incorporated into a trading model.

Tasks

Collect Financial News

- Use a news aggregator API (e.g., NewsAPI, Alpha Vantage, Yahoo Finance RSS)
- Retrieve headlines or brief snippets related to selected tickers (e.g., AAPL, SPY)
- Store data as a DataFrame with columns: timestamp, ticker, headline, source

Clean and Preprocess Text

- Convert to lowercase, remove punctuation, stop words
- Tokenize text and optionally apply stemming or lemmatization

Apply Sentiment Analysis Models

- Use pretrained models (e.g., VADER from nltk, TextBlob, or transformers)
- Compute polarity score or binary sentiment classification
- Aggregate sentiment scores daily per ticker:

```python
daily_sentiment = news.groupby(['date', 'ticker'])['sentiment_score'].mean()
```

Integrate Sentiment as a Feature

- Merge sentiment scores with market data
- Use as input to the ML model for forecasting or classification
- Test whether sentiment improves predictive performance

Visualize Sentiment Trends

- Plot average sentiment vs. stock price over time
- Compare high-sentiment and low-sentiment days

Deliverables

- News DataFrame with sentiment scores
- Visualization of sentiment trends
- Updated feature matrix including sentiment
- Evaluation comparison with and without sentiment features
- Brief commentary on the correlation between news tone and price movement

---

Submission Checklist

- ☐ Fully annotated Jupyter notebook
- ☐ All plots and evaluation summaries
- ☐ Commentary sections inline or as Markdown cells
- ☐ Feature matrix saved as CSV
- ☐ Final model predictions and metrics summary