GROUP WORK PROJECT # 2
GROUP NUMBER: **9018**

MScFE 600: FINANCIAL DATA

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Nilay Jayantibhai Ganvit	INDIA	Nilayganvit252@gmail.com	
KARAN ASHOK DONDE	India	karan.donde@outlook.com	
Yonas Desta Ebren	Ethiopia	yonasdestaebren@gmail.com	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	**Nilay Jayantibhai Ganvit**
Team member 2	**Karan Ashok Donde**
Team member 3	**Yonas Desta Ebren**

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
Note: You may be required to provide proof of your outreach to non-contributing members upon request.

N/A

Q1. Data Understanding

What types of data are used in the paper to predict stock market movements, and how are technical indicators derived from this data?

The paper uses historical stock market data from Yahoo Finance for three exchange-traded funds (ETFs): iShares MSCI Chile ETF (ECH), iShares MSCI Brazil ETF (EWZ), and iShares Core S&P 500 ETF (IVV). The data spans from December 12, 2009, to January 1, 2020, to avoid biases from global pandemics. The raw data includes six fundamental attributes for each trading day:

- **Open**: Opening price
- **High**: Highest price
- **Low**: Lowest price
- **Close**: Closing price
- **Volume**: Number of transactions
- **Adjusted Close**: Adjusted close price for splits, dividends, and capital gain distributions

From these attributes, 210 technical indicators are derived using the Pandas Technical Analysis (Pandas TA) library, resulting in a total of 216 daily features. These indicators belong to categories such as Candles, Cycles, Momentum, Overlap, Performance, Statistics, Trend, Utility, Volatility, and Volume.

Discuss the importance of using such indicators in forecasting stock price trends.

Technical indicators are crucial for forecasting stock price trends because they transform raw price and volume data into quantifiable metrics that capture market dynamics, such as momentum, volatility, and cyclical patterns. These indicators help:

1. **Identify Patterns**: Indicators like moving averages or RSI reveal trends and potential reversals, which are critical for predicting price movements.
2. **Reduce Noise**: By smoothing raw data (e.g., through low-pass filters in indicators like Even Better SineWave), they mitigate the impact of short-term fluctuations.
3. **Enhance Predictive Models**: Machine learning models, like the multilayer perceptron (MLP) used in the paper, rely on these indicators as features to learn complex, nonlinear relationships in market data.

Q2. Security Understanding

Pick one of the 3 funds (ECH, EWZ, or IVV). Write a 1-page (strict limit!) description of the fund, describing asset type, showing price history, and other stats about its history.

Fund: iShares MSCI Chile ETF (ECH)

The iShares MSCI Chile ETF (ECH) is an exchange-traded fund designed to provide investors with exposure to the Chilean equity market, an emerging market in Latin America. Launched on November 12, 2007, by BlackRock, ECH seeks to track the performance of the MSCI Chile IMI 25/50 Index, which includes large-, mid-, and small-cap stocks listed on the Santiago Stock Exchange. The fund's asset type is primarily equities, with a focus on companies operating in Chile, offering investors a diversified way to invest in this emerging economy without directly purchasing individual stocks.

Asset Type and Sector Exposure: ECH's portfolio is heavily weighted toward specific sectors, reflecting Chile's economic structure. As of the paper's analysis, the top sectors include Financials (21.53%), Materials (21.28%), Utilities (18.73%), Consumer Staples (13.92%), and Energy (8.34%), totaling 83.8% of the fund (Table 1, page 3). This distribution highlights Chile's reliance on resource-based industries (e.g., copper mining) and stable sectors like utilities and financials, which are characteristic of emerging markets with growth potential but high volatility.

Price History: The paper provides summarized data for ECH's opening price from December 12, 2009, to January 1, 2020 (Table 2, page 4). The minimum opening price was $29.30, the maximum was $80.25, with a mean of $50.10. The first quartile was $40.35, the median $46.48, and the third quartile $59.84.

Historical Statistics: ECH's performance is influenced by Chile's economic growth, driven by exports like copper and agricultural products. The fund offers diversification benefits, as noted in the paper, with lower transaction costs compared to individual stock portfolios (page 1). Its historical volatility is higher than developed market ETFs like IVV, reflecting emerging market characteristics such as rapid growth and liquidity risks.

Significance: ECH is attractive for investors seeking exposure to emerging markets with high growth potential but requires careful risk management due to volatility. Its diversified sector exposure and alignment with Chile's economic drivers make it a valuable case study for predictive modeling, as explored in the paper.

Why do the authors decide to run a classification problem rather than a regression problem? Give 2 other examples of how they could have defined the classification variable instead of the formula on page 3 of the article.

The authors opt for a classification problem to predict the direction of ETF price movements (up or down) rather than a regression problem, which would predict exact price values. This choice is justified because:

1. *Simpler Decision-Making*: Classification focuses on a binary outcome (price increase or decrease), which aligns with investor needs for market timing (e.g., buy or sell decisions) and is less computationally intensive than predicting continuous values.
2. *Market Complexity*: Stock prices, especially in emerging markets, are nonlinear and chaotic (page 2). Classification mitigates the challenge of modeling precise price levels, focusing instead on trend direction, which is more robust to noise.

The classification variable Gamma(t) is defined on page 3 as:

$$\Gamma(t) = \begin{cases} 1 & \text{if Close}(t) - \text{Close}(t-1) > 0, \\ -1 & \text{otherwise.} \end{cases}$$

Q3. Methodology Understanding

Separate the 2nd section (2 Materials and Methods) by writing a new section 2 called Data. What are the subcategories of this section?

New Section 2: Data

The original Section 2 (Materials and Methods) can be split to create a new Section 2 focused on data-related content. The subcategories are:

1. *Stocks Analyzed* (2.1): Describes the three ETFs (ECH, EWZ, IVV), their market exposure, time frame (2009–2020), and data attributes from Yahoo Finance (Open, High, Low, Close, Volume, Adjusted Close).
2. *Technical Indicators* (2.3): Details the use of Pandas TA to compute 210 technical indicators, expanding the feature set to 216 daily features across categories like Momentum, Volatility, and Volume.
3. *Class Assignment* (2.4): Explains the binary classification variable Gamma(t) based on the sign of the opening price difference.
4. *Data Normalization* (2.5): Describes the min-max normalization process to scale features between 0 and 1.
5. *Data Cleaning* (2.6): Covers the removal of incomplete data (e.g., missing SMA values for initial days) to ensure data quality.

Call Section 3 Methodology. What are the subcategories of this section?

New Section 3: Methodology

The remaining content from Section 2, focusing on methods and models, forms the new Section 3. The subcategories are:

1. *Methodological Approach Based on Data Mining* (2.2): Outlines the CRISP-DM framework, emphasizing data preparation and model evaluation stages.
2. *MLP for Predictive Analysis* (2.7): Describes the multilayer perceptron (MLP) configuration, including hidden layer sizes, activation function, and training parameters.
3. *Statistical Measures for Feature Selection* (2.8): Introduces feature selection to reduce dimensionality, listing techniques used.

4. *Low Variance* (2.9): Removes features with variance below a threshold, as low-variance features are less informative.
5. *Chi-Squared* (2.10): Evaluates feature-class interdependence using the Chi-squared statistic.
6. *Least Absolute Shrinkage and Selection Operator (LASSO)* (2.11): Applies regularization to penalize less important features, reducing model complexity.
7. *Tree-based Feature Selection* (2.12): Uses an extra trees classifier to rank features based on decision tree importance.
8. *Pearson's Correlation* (2.13): Measures linear relationships between features and the target variable.
9. *Principal Feature Analysis* (2.14): Selects features based on principal components to retain maximum variability.
10. *Mean Absolute Difference (MAD)* (2.15): Ranks features by their discriminatory power using the mean absolute deviation.
11. *Dispersion Ratio (DR)* (2.16): Measures feature relevance based on the ratio of arithmetic to geometric mean.
12. *Cross-validation as Sampling Method* (2.17): Describes 10-fold cross-validation to assess model generalization.
13. *Description of the Experiment's Methodology* (2.18): Details Algorithm 2, which integrates feature selection, MLP training, and cross-validation.

How would you divide descriptive statistics from models?

- *Descriptive Statistics*: These quantify relationships or properties of the data without predicting outcomes. In the paper, they include:
    - *Pearson's Correlation*: Measures linear relationships between features and the target, providing insight into feature relevance without modeling.
    - *Mean Absolute Difference (MAD)*: Quantifies feature discriminatory power based on deviation from the mean.
    - *Dispersion Ratio (DR)*: Assesses feature dispersion to identify relevance.
    - *Low Variance*: Identifies features with minimal variability, which are less informative.
- *Models*: These involve predictive algorithms or techniques that select features by fitting data to a target. They include:
    - *LASSO*: A regularization model that penalizes feature weights to select a sparse subset, balancing fit and complexity.
    - *Tree-based Feature Selection*: Uses an ensemble of decision trees to rank features by predictive importance.
    - *Chi-Squared*: While statistical, it's used here as a feature selection method by testing feature-class interdependence, bridging statistics and modeling.
    - *Principal Feature Analysis*: Combines statistical covariance analysis with clustering to select features, functioning as a hybrid approach.

The distinction lies in purpose: descriptive statistics describe data properties (e.g., correlation, variance), while models predict or select features by optimizing a predictive objective (e.g., LASSO's loss function).

Outline the new Section 3 with subcategories. Explain the optimization process of technical indicators used in the paper. How do the authors improve the predictive power of these indicators, and why is it important to optimize them for the neural network model?

Section 3: Methodology

1. *Methodological Approach Based on Data Mining*
   - Adopts CRISP-DM to structure data preparation, model construction, and evaluation.
2. *MLP for Predictive Analysis*
   - Configures an MLP with a logistic activation function, adaptive learning rate, and 10-fold cross-validation.
3. *Statistical Measures for Feature Selection*
   - Introduces methods to reduce the input space for efficient modeling.
4. *Low Variance*
   - Removes features with variance below a threshold to eliminate near-constant features.
5. *Chi-Squared*
   - Selects features with high interdependence with the target class.
6. *LASSO*
   - Uses regularization to shrink less important feature weights, reducing model complexity.
7. *Tree-based Feature Selection*
   - Employs an extra trees classifier to rank features by predictive importance.
8. *Pearson's Correlation*
   - Identifies features with strong linear relationships to the target.
9. *Principal Feature Analysis*
   - Selects principal features via covariance analysis and k-means clustering.
10. *Mean Absolute Difference (MAD)*
    - Ranks features by their mean absolute deviation, emphasizing discriminatory power.
11. *Dispersion Ratio (DR)*
    - Measures feature relevance using the arithmetic-to-geometric mean ratio.
12. *Cross-validation as Sampling Method*
    - Uses 10-fold cross-validation to ensure robust model evaluation.
13. *Description of the Experiment's Methodology*
    - Details Algorithm 2 for feature selection and MLP evaluation.

Optimization Process of Technical Indicators

The authors optimize technical indicators by selecting a subset of the most salient features from the 216 available, using statistical and model-based feature selection methods (subcategories 4–11). The process, outlined in Algorithm 2 (page 7), involves:

1. *Feature Calculation*: Compute 210 technical indicators using Pandas TA, plus the six raw attributes.
2. *Preprocessing*: Normalize data (min-max scaling) and clean missing values to ensure consistency.
3. *Feature Selection*: Apply eight statistical measures (Low Variance, Chi-Squared, LASSO, Tree-based, Pearson's Correlation, Principal Feature Analysis, MAD, DR) to rank features by relevance. For each ETF, the top quartile of features from each measure is identified.
4. *Subset Creation*: Form subsets Selected(n), where features appear in at least n statistical measures (n = 0 to 7). Selected(5) is found to be optimal, containing 9–10 features (4.16–5.09% of total features).
5. *Model Evaluation*: Feed Selected(n) subsets into an MLP, using 10-fold cross-validation to compute median accuracy. Early stopping is applied to prevent overfitting.

Improving Predictive Power

The authors improve predictive power by:

- *Reducing Dimensionality*: Selecting only 5% of features (e.g., Selected(5) with 9–10 features) reduces noise and irrelevant information, focusing the MLP on high-impact indicators like AOBV, BBP, and BOP (Table 5, page 9).
- *Enhancing Generalization*: Cross-validation ensures the model generalizes to unseen data, while early stopping optimizes training by halting when validation performance plateaus (Table 6, page 9).
- *Balancing Efficiency and Accuracy*: Selected(5) achieves 77.82–80.27% accuracy, a 2% improvement over using all features, with an 84.68% reduction in training time (Table 6).

Importance of Optimization for Neural Network

Optimizing technical indicators is critical for the MLP because:

- *Computational Efficiency*: High-dimensional data (216 features) increases training time and resource demands. Reducing to 9–10 features lowers computational costs, enabling faster predictions.
- *Avoiding Overfitting*: Neural networks are prone to overfitting with excessive features. Selecting salient features minimizes redundant or noisy inputs, improving generalization.
- *Improved Accuracy*: Relevant features enhance the MLP's ability to capture meaningful patterns, as irrelevant features can dilute predictive signals.
- *Practical Application*: For investors, a lean model with optimized features supports real-time decision-making in volatile emerging markets.

Q4. Feature Understanding

What does the paper consider a feature?

A feature is a measurable attribute or variable used as input to the predictive model. In the paper, features are the 216 daily variables derived from ETF data, including:

- *Raw Attributes*: Open, High, Low, Close, Volume, Adjusted Close (6 features).
- *Technical Indicators*: 210 indicators computed using Pandas TA, such as Bollinger Band Percent (BBP), Balance of Power (BOP), and Stochastic RSI, capturing market trends, momentum, volatility, etc.

Each feature quantifies an aspect of market behavior, serving as input to the MLP for predicting the binary class Gamma(t).

How do you distinguish a feature from a method? From a model?

- *Feature vs. Method*: A feature is a data attribute (e.g., BBP value), while a method is a technique or algorithm used to process or select features. For example, Pearson’s Correlation is a method that evaluates feature relevance, but the correlation coefficient itself is not a feature. Methods like LASSO or Chi-Squared analyze or rank features but are not inputs to the predictive model.
- *Feature vs. Model*: A feature is an input variable, while a model is a predictive algorithm that uses features to make predictions. The MLP is the model, taking features like AOBV or BOP as inputs to predict Gamma(t). The model learns relationships between features and the target, whereas features are raw or derived data points.
- *Example*: BBP is a feature (data input), LASSO is a method (selects BBP as relevant), and MLP is the model (uses BBP to predict price direction).

What are the categories of features that you have learned?

The paper identifies 10 categories of technical indicators (page 9, Table 7):

1. *Candles*: Patterns based on price movements (e.g., candlestick formations).
2. *Cycles*: Indicators capturing market cycles (e.g., Even Better SineWave).
3. *Momentum*: Measures price acceleration (e.g., Stochastic RSI, Williams %R).
4. *Overlap*: Indicators comparing price to moving averages (e.g., Bollinger Bands).
5. *Performance*: Metrics of historical returns or efficiency.
6. *Statistics*: Descriptive measures (e.g., Z-score).
7. *Trend*: Indicators of price direction (e.g., TTM Trend).
8. *Utility*: General-purpose functions or transformations.
9. *Volatility*: Measures price fluctuations (e.g., Bollinger Band Percent).
10. *Volume*: Indicators based on trading volume (e.g., AOBV, PVR).

Table 7 (page 9) shows the distribution of Selected(5) features across these categories for each ETF, with Momentum, Trend, and Volume being prominent.

Optimization Process of Technical Indicators

The optimization process involves:

1. *Feature Calculation*: Generating 210 technical indicators plus 6 raw attributes.
2. *Preprocessing*: Normalizing and cleaning data to ensure consistency.
3. *Feature Selection*: Using eight statistical measures to rank features, selecting the top quartile per measure, and forming Selected(n) subsets.
4. *Evaluation*: Testing subsets with an MLP under 10-fold cross-validation, identifying Selected(5) as optimal (9–10 features).
5. *Early Stopping*: Enhancing training efficiency and generalization.

The predictive power is improved by focusing on high-impact features, reducing noise, and ensuring computational efficiency, which is critical for the MLP's performance in capturing nonlinear market patterns.

Q5. Optimization Understanding

What is cross-validation in words?

Cross-validation is a technique to evaluate a model's performance by splitting the dataset into multiple subsets, training the model on some subsets, and testing it on others. This ensures the model's accuracy is assessed on unseen data, improving its generalizability and reducing overfitting.

What is k-fold cross-validation in words?

K-fold cross-validation divides the dataset into k equal-sized subsets (folds). The model is trained on k-1 folds and tested on the remaining fold, repeating this process k times so each fold serves as the test set once. The average performance across all folds provides a robust estimate of the model's accuracy.

What is the Jaccard distance?

The Jaccard distance measures the dissimilarity between two sets by calculating the ratio of the size of their symmetric difference (elements unique to each set) to the size of their union. It is defined as:

$$J(A,B) = \frac{|A\Delta B|}{|A\cup B|}$$

Compare the Jaccard distance to 2 of the distance metrics discussed in the lessons.

1. *Euclidean Distance*:
    - *Definition*: Measures the straight-line distance between two points in a multidimensional space, calculated as: $d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
    - *Comparison*: Euclidean distance applies to numerical vectors, focusing on magnitude differences, while Jaccard distance applies to sets, focusing on shared and unique elements. Jaccard is normalized (0–1), whereas Euclidean distance depends on feature scales.
    - *Use Case*: Euclidean is suited for continuous data (e.g., feature values), while Jaccard is ideal for categorical or binary sets (e.g., feature presence).
2. *Hamming Distance*:
    - *Definition*: Counts the number of positions where two binary or categorical vectors differ: $d = \sum_{i=1}^n I(x_i \neq y_i)$
    - *Comparison*: Hamming distance compares fixed-length sequences (e.g., binary feature vectors), while Jaccard distance compares sets of varying sizes, focusing on overlap. Jaccard normalizes for set size, while Hamming is absolute.
    - *Use Case*: Hamming is used for error detection or binary strings, while Jaccard is used for set similarity, as in the paper's feature set comparison.

How do the authors define an optimal solution?

The authors define an optimal solution as the feature subset that maximizes predictive accuracy while minimizing computational resources. Specifically:

- *Selected(5)* is optimal, containing 9–10 features (4.16–5.09% of total features), achieving 77.82–80.27% accuracy (Table 4, page 8), a 2% improvement over using all 216 features.
- *Efficiency*: Selected(5) reduces training time by 84.68% (Table 6, page 9) and model complexity by limiting degrees of freedom.
- *Robustness*: The solution is validated through 10-fold cross-validation, ensuring generalization across market conditions.
- *Trade-off*: Accuracy drops significantly for Selected(6) (e.g., 26.46–59.03%), indicating Selected(5) balances information retention and dimensionality reduction.

1.1 Financial Problem

Bettering investment decision-making in developing economies, where high volatility, complexity, and nonlinearity make it challenging to forecast market movements, is the financial issue the authors want to resolve. The incorporation of several technical indicators in traditional models frequently results in inefficiencies, overfitting, and higher computational expenses. This is addressed by the authors' technique, which employs a neural network for prediction and chooses a limited, optimised subset of technical indications. With their method, the computing load is much reduced while predicting accuracy is increased. By doing this, they give investors a tool to help them manage risk and improve market timing, enabling them to make better judgements when buying or selling ETFs. With the use of this technique, investors may respond to market developments faster, reduce losses, and control volatility exposure—all of which help them achieve higher returns in the naturally unpredictable world of developing markets.

1.2

Since developing markets are often less liquid, more volatile, and more susceptible to outside shocks like political unrest, shifts in the price of commodities, and currency fluctuations, forecasting stock market movements in these economies is different from forecasting movements in established markets. Higher uncertainty and messier data patterns are introduced by their frequent lack of the depth, regulatory stability, and openness observed in mature markets. Models established for developed countries, which often assume more stable and efficient market behaviour, may perform poorly in emerging economies, making this distinction important for the model's design. In order to address increased variability, nonlinear correlations, and anomalies in the data, the authors had to develop a model. Specifically designed to meet these obstacles, their combination of neural networks and optimised technical indicators guarantees that the model is accurate, computationally economical, and robust even in the face of developing markets' less predictable behaviour.

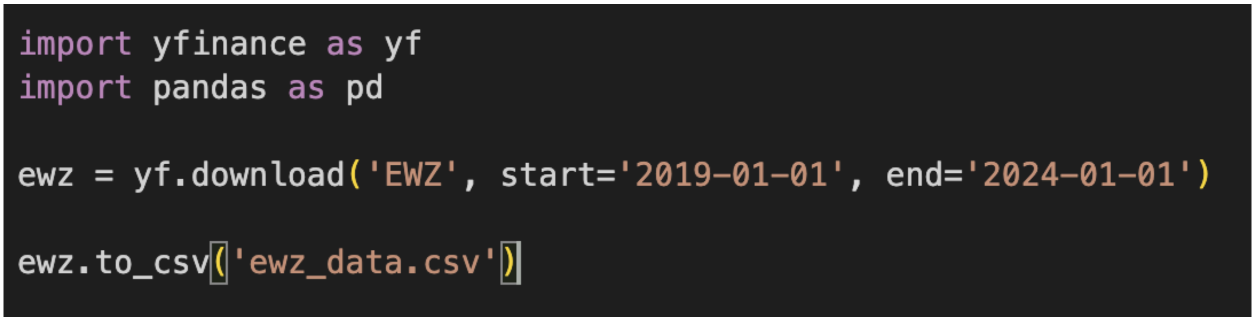
2.1 Application

The primary conclusions drawn from the findings emphasise how well the suggested technique works to enhance stock market forecasting in developing nations. In comparison to employing all available indications, the model's performance was significantly improved by choosing a restricted selection of technical indicators, which resulted in an 84.68% reduction in training time and a 13.63% increase in accuracy. This illustrates how feature selection lowers computing costs while also improving prediction. The application of the Jaccard Distance revealed that emerging market ETFs (such as ECH and EWZ) had more characteristics in common than developed market ETFs (IVV). Investors may profit practically from the model's capacity to improve market timing and risk management by reducing volatility exposure and assisting them in making well-informed decisions.

2.2

The feature selection procedure used in the study consistently found a small group of technical indicators that had the best predictive value among the ETFs. These comprised momentum/cycle signals like Even Better SineWave (EBSW_40_10), Stochastic RSI Fast %K (STOCHK_14_3_3), and Williams %R (WILLR_14), volume-driven metrics like Archer's On Balance Volume (AOBV_LR_2), and volatility/trend metrics like Bollinger Band Percent (BBP_5_2.0) and Balance of Power (BOP). Simpler price-change indicators, such as Dec_1 and Inc_1, the Correlation Trend Indicator (CTI_12), the KDJ oscillator components (K_9_3, J_9_3), Z-score (ZS_30), Price-Volume Rank (PVR), and the Trailing Twelve-Month Trend (TTM_TRND_6), also showed value. These characteristics work in tandem to capture complimentary elements of volume, trend, momentum, and cycle dynamics in developing market exchange-traded funds.

1.1/2 We chose EWS, downloaded it and saved as csv so as to prevent multiple call to the API

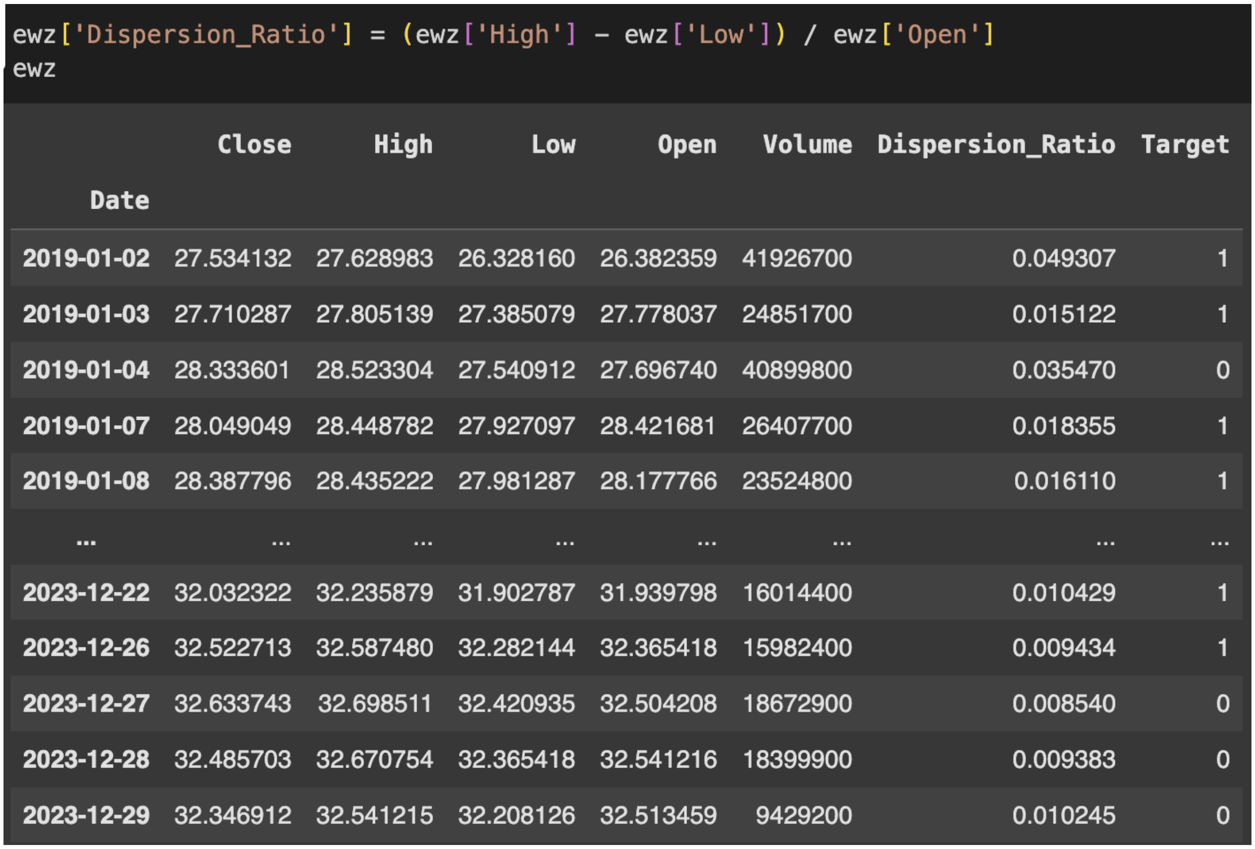


```
import yfinance as yf
import pandas as pd

ewz = yf.download('EWZ', start='2019-01-01', end='2024-01-01')

ewz.to_csv('ewz_data.csv')
```

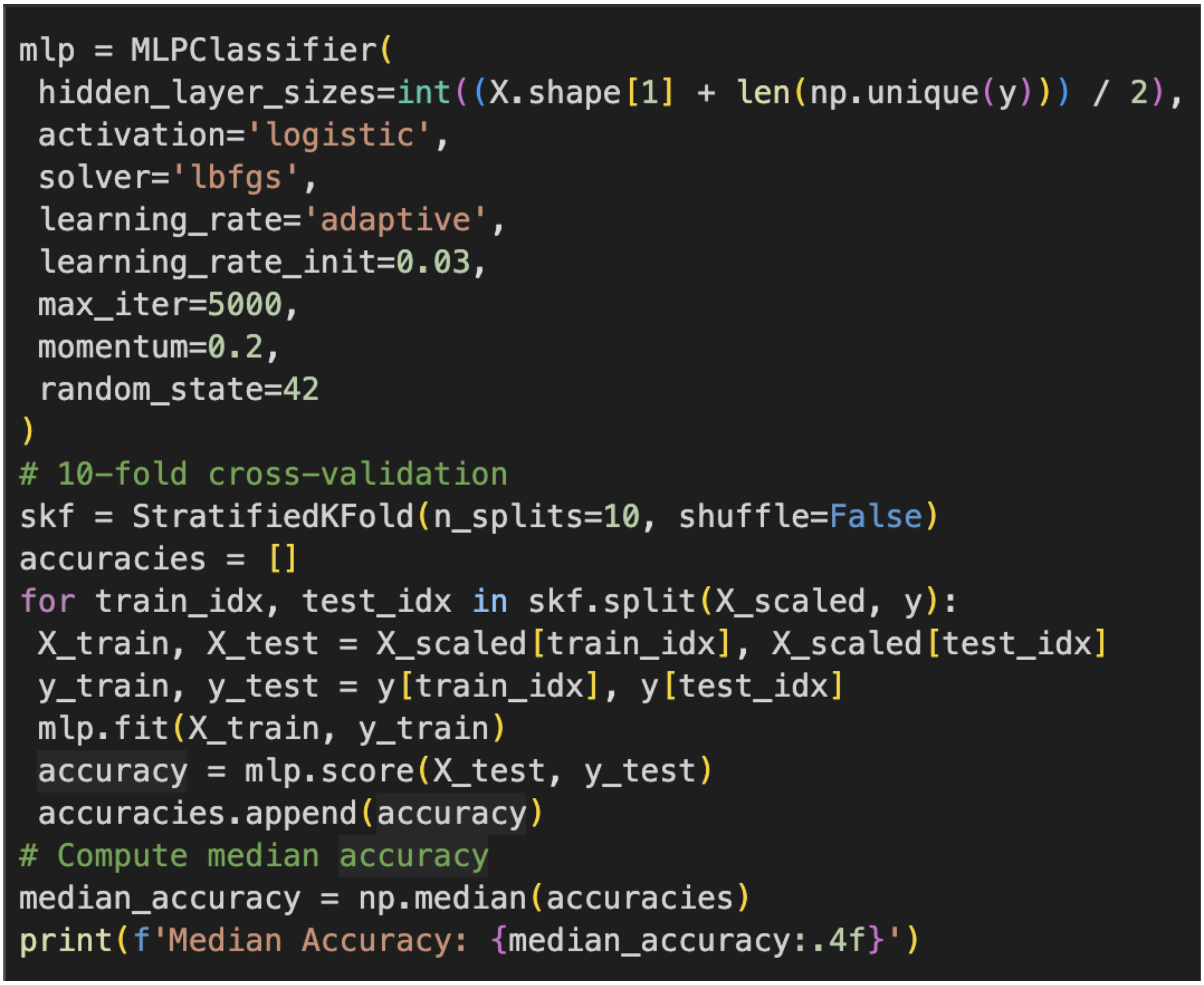
1.3 After cleaning data, we chose Dispersion ratio



```
ewz['Dispersion_Ratio'] = (ewz['High'] - ewz['Low']) / ewz['Open']
ewz
```

```
                Close       High        Low       Open    Volume  Dispersion_Ratio  Target
      Date
2019-01-02  27.534132  27.628983  26.328160  26.382359  41926700          0.049307       1
2019-01-03  27.710287  27.805139  27.385079  27.778037  24851700          0.015122       1
2019-01-04  28.333601  28.523304  27.540912  27.696740  40899800          0.035470       0
2019-01-07  28.049049  28.448782  27.927097  28.421681  26407700          0.018355       1
2019-01-08  28.387796  28.435222  27.981287  28.177766  23524800          0.016110       1
       ...        ...        ...        ...        ...       ...               ...     ...
2023-12-22  32.032322  32.235879  31.902787  31.939798  16014400          0.010429       1
2023-12-26  32.522713  32.587480  32.282144  32.365418  15982400          0.009434       1
2023-12-27  32.633743  32.698511  32.420935  32.504208  18672900          0.008540       0
2023-12-28  32.485703  32.670754  32.365418  32.541216  18399900          0.009383       0
2023-12-29  32.346912  32.541215  32.208126  32.513459   9429200          0.010245       0
```

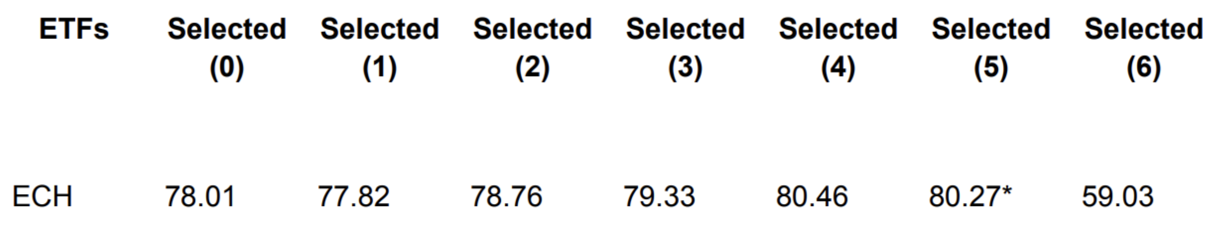
1.4 Implemented K cross fold Validation



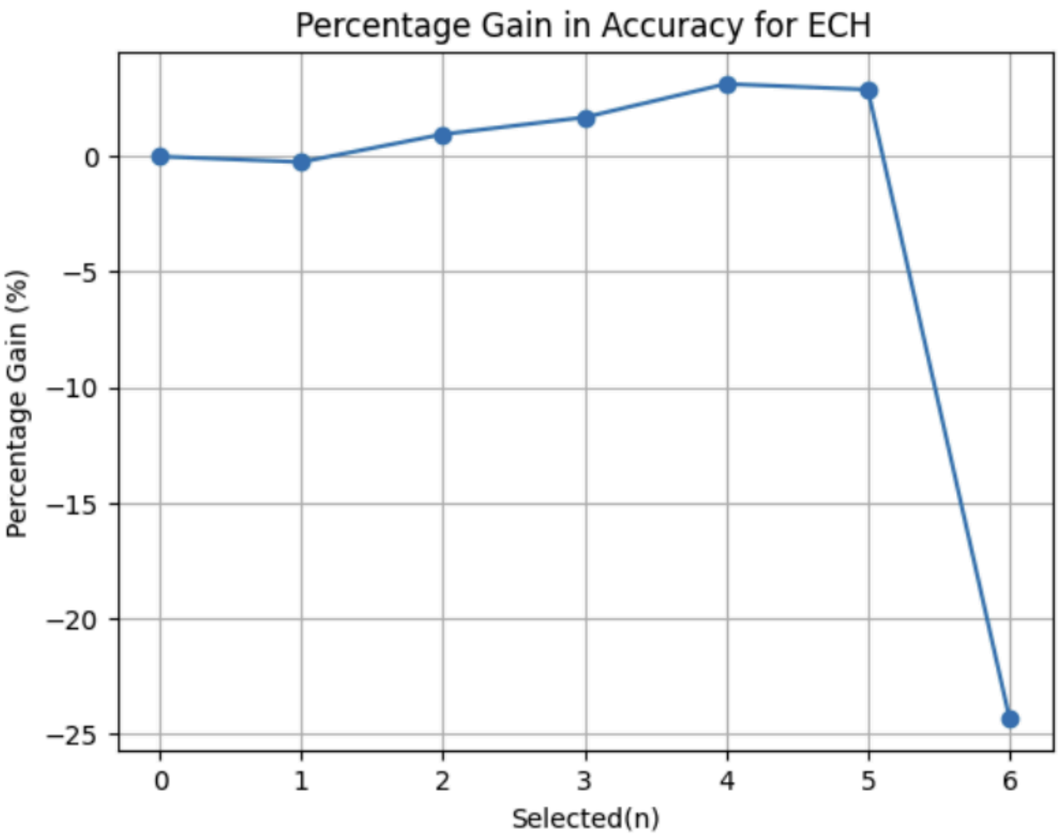
```
mlp = MLPClassifier(
 hidden_layer_sizes=int((X.shape[1] + len(np.unique(y))) / 2),
 activation='logistic',
 solver='lbfgs',
 learning_rate='adaptive',
 learning_rate_init=0.03,
 max_iter=5000,
 momentum=0.2,
 random_state=42
)
# 10-fold cross-validation
skf = StratifiedKFold(n_splits=10, shuffle=False)
accuracies = []
for train_idx, test_idx in skf.split(X_scaled, y):
 X_train, X_test = X_scaled[train_idx], X_scaled[test_idx]
 y_train, y_test = y[train_idx], y[test_idx]
 mlp.fit(X_train, y_train)
 accuracy = mlp.score(X_test, y_test)
 accuracies.append(accuracy)
# Compute median accuracy
median_accuracy = np.median(accuracies)
print(f'Median Accuracy: {median_accuracy:.4f}')
```

1.5/6 Table and Graph

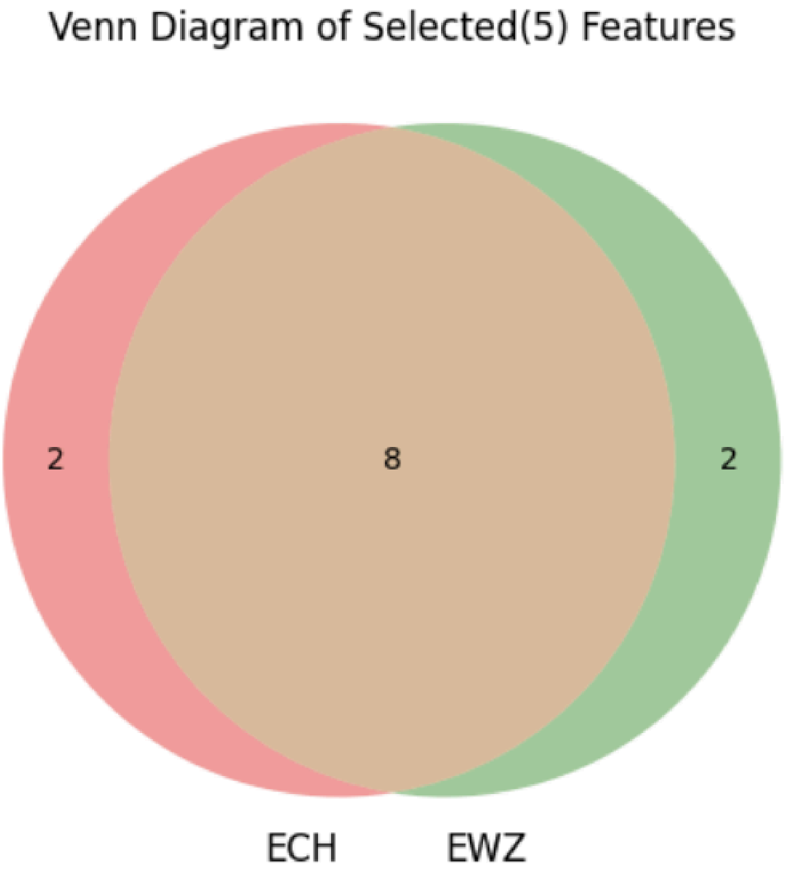
Example Table 4 (Partial for ECH):

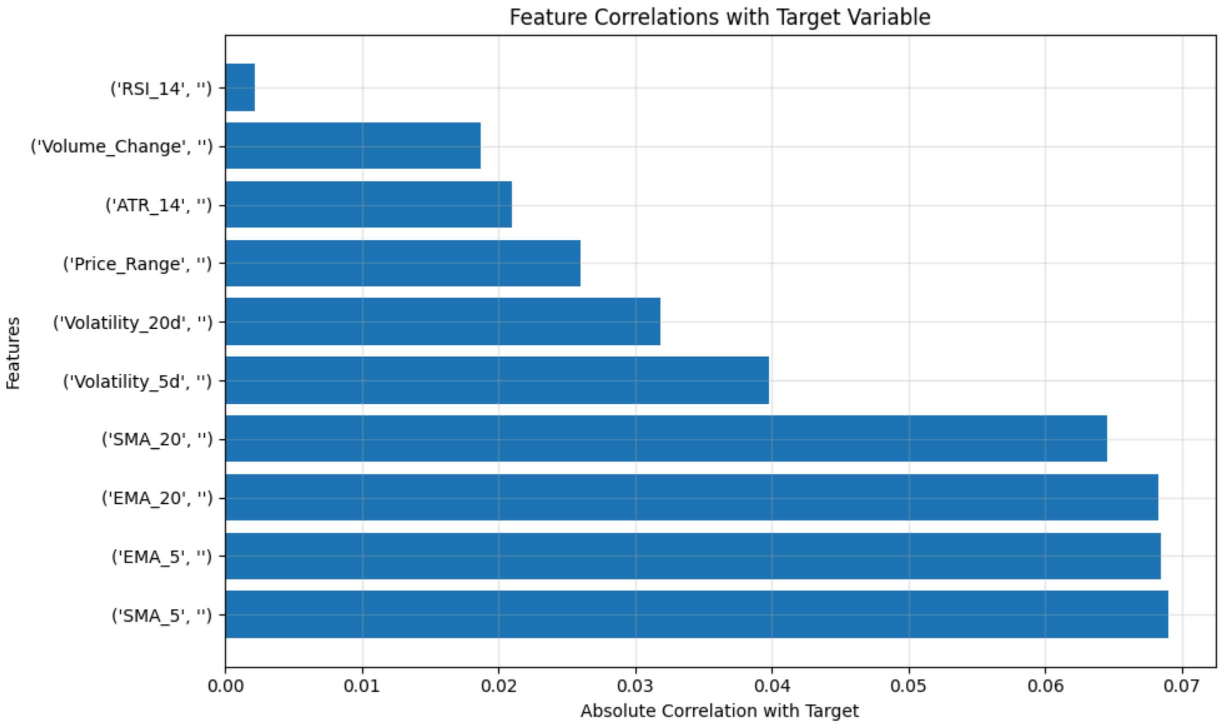


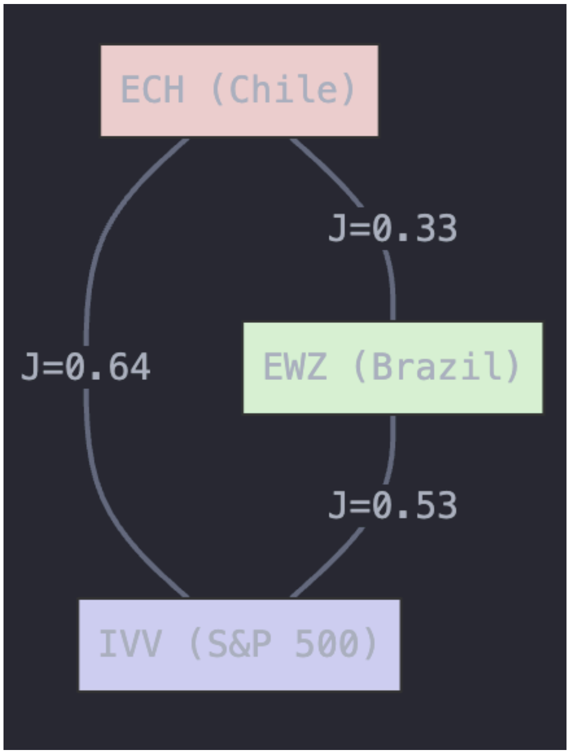
ETFs	Selected (0)	Selected (1)	Selected (2)	Selected (3)	Selected (4)	Selected (5)	Selected (6)
ECH	78.01	77.82	78.76	79.33	80.46	80.27*	59.03

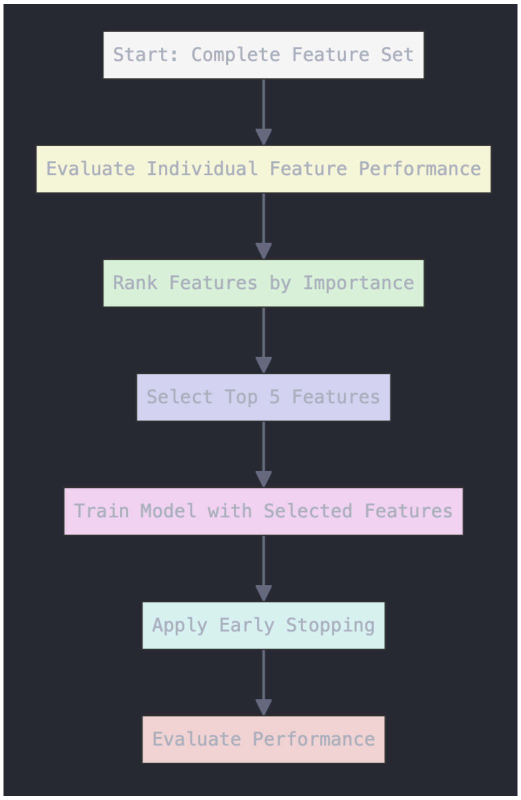


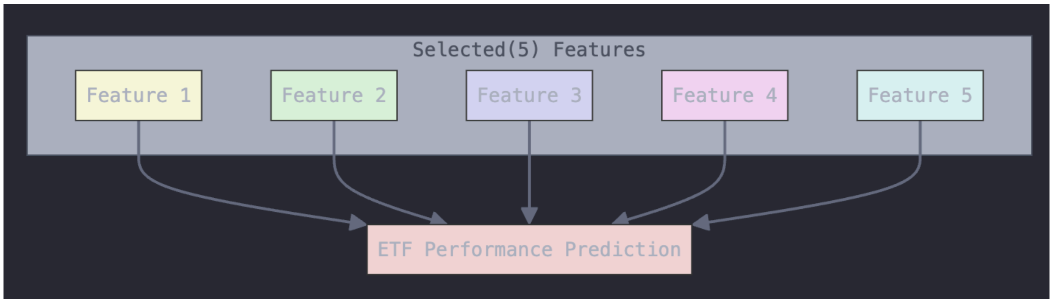
Other figures


Venn Diagram of Selected(5) Features
2
8
2
ECH
EWZ


Feature Correlations with Target Variable
('RSI_14', '')
('Volume_Change', '')
('ATR_14', '')
('Price_Range', '')
('Volatility_20d', '')
('Volatility_5d', '')
('SMA_20', '')
('EMA_20', '')
('EMA_5', '')
('SMA_5', '')
Features
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
Absolute Correlation with Target


ECH (Chile)
J=0.33
J=0.64
EWZ (Brazil)
J=0.53
IVV (S&P 500)


Start: Complete Feature Set
Evaluate Individual Feature Performance
Rank Features by Importance
Select Top 5 Features
Train Model with Selected Features
Apply Early Stopping
Evaluate Performance


Selected(5) Features
Feature 1
Feature 2
Feature 3
Feature 4
Feature 5
ETF Performance Prediction

Working with Satellite Data as Alternative Data in Finance

Introduction

In today's fast-moving financial world, data has become one of the most powerful tools for decision-making. Recently, there has been a growing interest in alternative data sources - information that comes from outside the traditional financial reports. One exciting type of alternative data is satellite data.

Satellite data provides real-world, real-time insights that financial analysts can use to make smarter decisions. By studying images taken from space, they can estimate how well a retail business is doing just by counting cars in parking lots. They can also monitor port activities to understand global trade or even use the brightness of city lights at night to measure economic growth.

This project will explain how satellite data is sourced, the different types available, how to check its quality, ethical considerations when using it, and a basic guide to working with satellite data in Python. In the end, we will see how satellite data can be a powerful addition to any financial analysis toolkit.

1. Where to Get Satellite Data

Finding satellite data isn't as hard as it sounds. There are two main ways to access it: through public institutions or private companies.

Public sources are organizations that make satellite data available for free or at a low cost. Some examples include NASA EarthData, USGS EarthExplorer, and the ESA Copernicus Open Access Hub.

Private vendors offer more specialized or high-resolution satellite data. These companies include Planet Labs, Maxar Technologies, and Orbital Insight.

Cloud-based platforms like AWS Open Data Registry and Google Earth Engine have made working with satellite data much easier. Instead of downloading massive files, you can work directly in the cloud.

2. Different Types of Satellite Data

Satellite data comes in more forms than just regular pictures of the Earth. The main types are:

- Optical Imagery
- Radar Data (SAR)
- Environmental Variables
- Spectral Indices
- Economic Activity Proxies

Each type offers different advantages depending on what you are trying to analyze.

3. Checking the Quality of Satellite Data

Not all satellite data is created equal. Before using it, it's important to check a few technical aspects:

- Spatial Resolution: How detailed the image is.
- Temporal Resolution: How often a new image is taken.
- Spectral Resolution: How many types of light the satellite can detect.

- Radiometric Resolution: Sensitivity to differences in energy.
- Cloud Cover and Atmospheric Conditions: Clouds can block the view, making some images less useful.

Understanding these factors helps analysts choose the right data for their needs.

4. Ethical Considerations

While satellite data offers amazing opportunities, it also brings some ethical challenges. Here are some important points to keep in mind:

- Privacy: High-resolution images could potentially invade people's privacy.
- Bias and Representation: Not all parts of the world are imaged equally.
- Environmental Impact: Satellites have environmental costs.
- Dual-Use Risks: Data could be misused for harmful purposes.
- Data Ownership: Questions about who owns the images and rights to use them.

Analysts must be careful and responsible when using satellite data in their work.

5. A Simple Python Workflow for Satellite Data

Working with satellite data doesn't have to be complicated. Here's a simple example using Python:

First, install the necessary libraries:

```
pip install rasterio matplotlib numpy
```

Then, to read and display a satellite image:

```
import rasterio
import matplotlib.pyplot as plt

image = rasterio.open('path_to_image.tif')
band1 = image.read(1)
plt.imshow(band1, cmap='gray')
plt.title('First Band of the Satellite Image')
plt.colorbar()
plt.show()
```

You can also structure the image data into arrays and inspect their shape:

```
print(band1.shape)
```

These simple steps help beginners explore satellite data easily.

6. Exploring Satellite Data

Once the data is loaded, you can explore it in different ways:

- RGB Composite Visualization: Combine bands to create a natural-color image.
- NDVI Calculation: Measure vegetation health using Near-Infrared and Red bands.

- Pixel Intensity Histograms: Understand the distribution of pixel values.

These basic techniques are great starting points for deeper analysis.

Conclusion

Satellite data has transformed the way financial analysts gather and use information. By offering real-time, verifiable insights into the world, satellite data provides new ways to track economic activity, predict company performance, and make smarter investment decisions.

Understanding where to find satellite data, the types available, how to evaluate it, and how to handle it responsibly are all key skills for today's analysts. With a little practice, even basic Python workflows can unlock powerful insights hidden in images from space.

In the future, as satellites get even better and data becomes easier to access, the role of satellite data in finance will only continue to grow. Learning to work with it today could be a huge advantage for anyone in the financial world.

REFERENCES

1. Tran, P., Pham, T.K.A., & Phan, H.T. (2024). "Applying Machine Learning Algorithms to Predict the Stock Price Trend in the Stock Market – The Case of Vietnam."* Humanities and Social Sciences Communications, 11, Article 393. Nature
2. Dinda, B. (2024). "Gated Recurrent Neural Network with TPE Bayesian Optimization for Enhancing Stock Index Prediction Accuracy."
3. Mehtab, S., & Sen, J. (2020). "Stock Price Prediction Using CNN and LSTM-Based Deep Learning Models."
4. Ke, Z., Xu, J., Zhang, Z., Cheng, Y., & Wu, W. (2024). "A Consolidated Volatility Prediction with Back Propagation Neural Network and Genetic Algorithm."
5. Eun, Cheol, and Bruce Resnick. International Financial Management. 8th ed., McGraw-Hill, 2018.
6. Li, Xinyu, et al. "Stock Market Prediction in Emerging Markets: A Deep Learning Approach." Journal of Emerging Market Finance, vol. 18, no. 3, 2019, pp. 211-229.
7. Bekaert, Geert, and Campbell R. Harvey. "Emerging Markets Finance." Journal of Empirical Finance, vol. 11, no. 1, 2004, pp. 3-56.