Investment Forecasting

Presented by Alex Abraham, Amir Nabiyev, Azim Rahat, Md Mamun, Wen Jie Long

Fall 2025 CSc 46000

Agenda

1. Introduction and Problem Statement
2. Method and Process
3. Data Collection
4. Modeling or Analytical Approach
5. Presenting Insights

INTRODUCTION AND PROBLEM STATEMENT

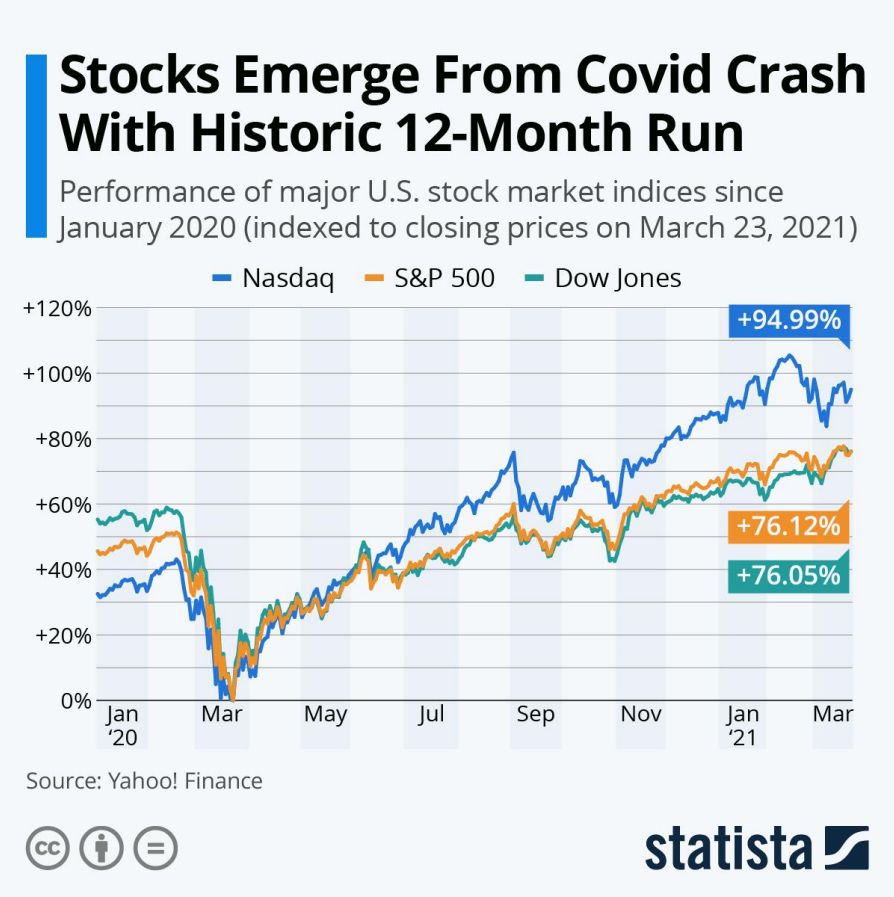
Forecasting Future Performance of Major Stock Indices Using Machine Learning.

Investors face uncertainty when choosing which major U.S. stock index offers the best long-term potential

We'll use 10+ years of market data from stock APIs and apply machine learning to forecast future performance.

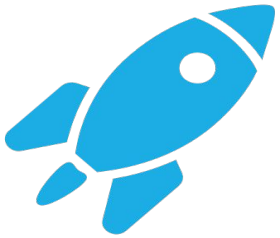
Python models and Tableau dashboards will reveal which index shows the strongest growth and stability.

Can we forecast the future performance of major stock indices (S&P 500, NASDAQ, Dow Jones) to determine which offers the best investment potential?



- Index funds and Exchange-Traded Funds (ETF) rely on stock index performance
- Tracking performance is important to judging volatility and risk
- Knowing which indices are better can yield better returns
- Being able to automate performance analytics means lower costs

Introduction and Problem



Project background and context

There are many different indices. For example, the top 3 followed indices in the US are Standard & Poor's 500, the Dow Jones Industrial Average, and the Nasdaq Composite.
Different indices focus on different things. For example, Dow Jones focuses on "blue chip" companies, which are stable, publicly traded companies with long reputations.

Problems you want to find answers

What can we use to predict future performance?

Will linear regression be a good model?

What time range should we look at for linear regression?

Stakeholders and Measurable Goal

Stakeholders

ETF Managers

Retail Investors

Index Fund Providers

Financial Analysts

Measurable Goal

Forecast 5-year growth (2025–2030)

Compare CAGR of S&P 500, NASDAQ, Dow Jones

Highlight best index for growth + stability

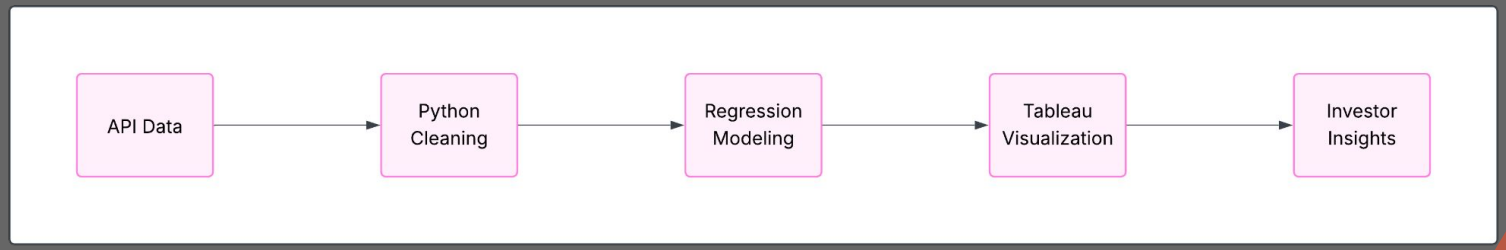
METHOD AND PROCESS

How we will Collect, Process, and Analyze the Data

How Our Process Works

- Collect 10+ years of index price data using Polygon.io API
- Clean & normalize data in Python (Pandas)
- Apply Linear Regression (scikit-learn) to forecast performance
- Visualize results using Tableau dashboards for comparison
- Extract Investment insights

Journey Map



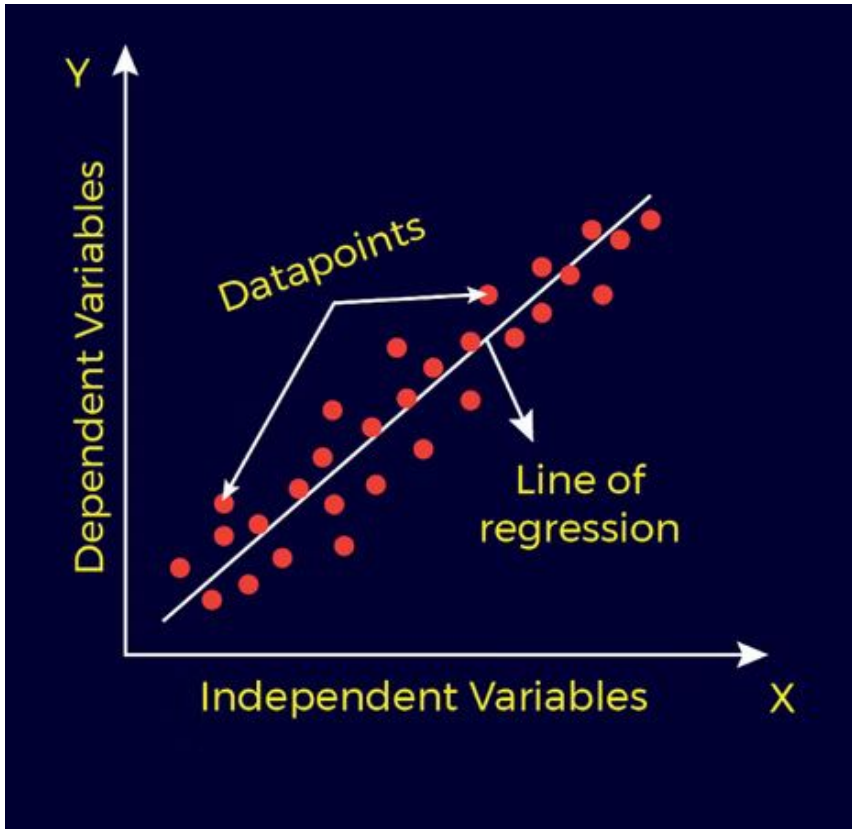
METHODOLOGY

Executive Summary

- Data collection methodology:
  - Data will be collected from Polygon.io API covering 10+ years of daily closing prices for the S&P 500, NASDAQ, and Dow Jones indices
- Perform data wrangling
  - Clean and format dataset using Python (Pandas), including cleaning nulls, aligning dates, and normalizing prices
- Perform exploratory data analysis (EDA)
  - Conduct analysis to understand historical performance, volatility, and inter-index relationships
- Modeling Approach
  - Apply Linear Regression (scikit-learn) to forecast future performance based on historical closing prices
  - Train and validate the model using R^2 (variance) and RMSE (average prediction error) as metrics
- Visualization and Insights
  - Build interactive Tableau dashboards to visualize predicted growth, stability, and returns for all three indices
  - Present key investment insights

Why Regression Analysis?

- Predicts Trends Over Time
  - Can model long-term index movement using historical closing prices.
- Works Well With Continuous Data
  - Ideal for forecasting numerical outputs like index prices or returns.
- Interpretable Results
  - Model coefficients show how past performance impacts future trends.
- Low Complexity, High Transparency
  - Easier to validate and explain than black-box models.
- Baseline Model
  - Serves as a benchmark before testing advanced methods.



DATASET

DATA COLLECTION

We plan to collect our data using live financial APIs

- Using GET requests to the Polygon.io API, we will retrieve 10+ years of daily historical prices for the S&P 500, NASDAQ, and Dow Jones indices
- We will decode the API response as JSON and convert it into a Pandas DataFrame, organizing columns with *date, open, close, high, low, volume*
- We will then clean the data, handling any missing values or misaligned dates
- To help with our regression analysis, we plan to calculate daily returns and percent changes for each index in our processing
- The cleaned dataset will feed into Tableau through a live connection for visualization

Polygon.io API

Retrieved data using a GET request with Bearer token

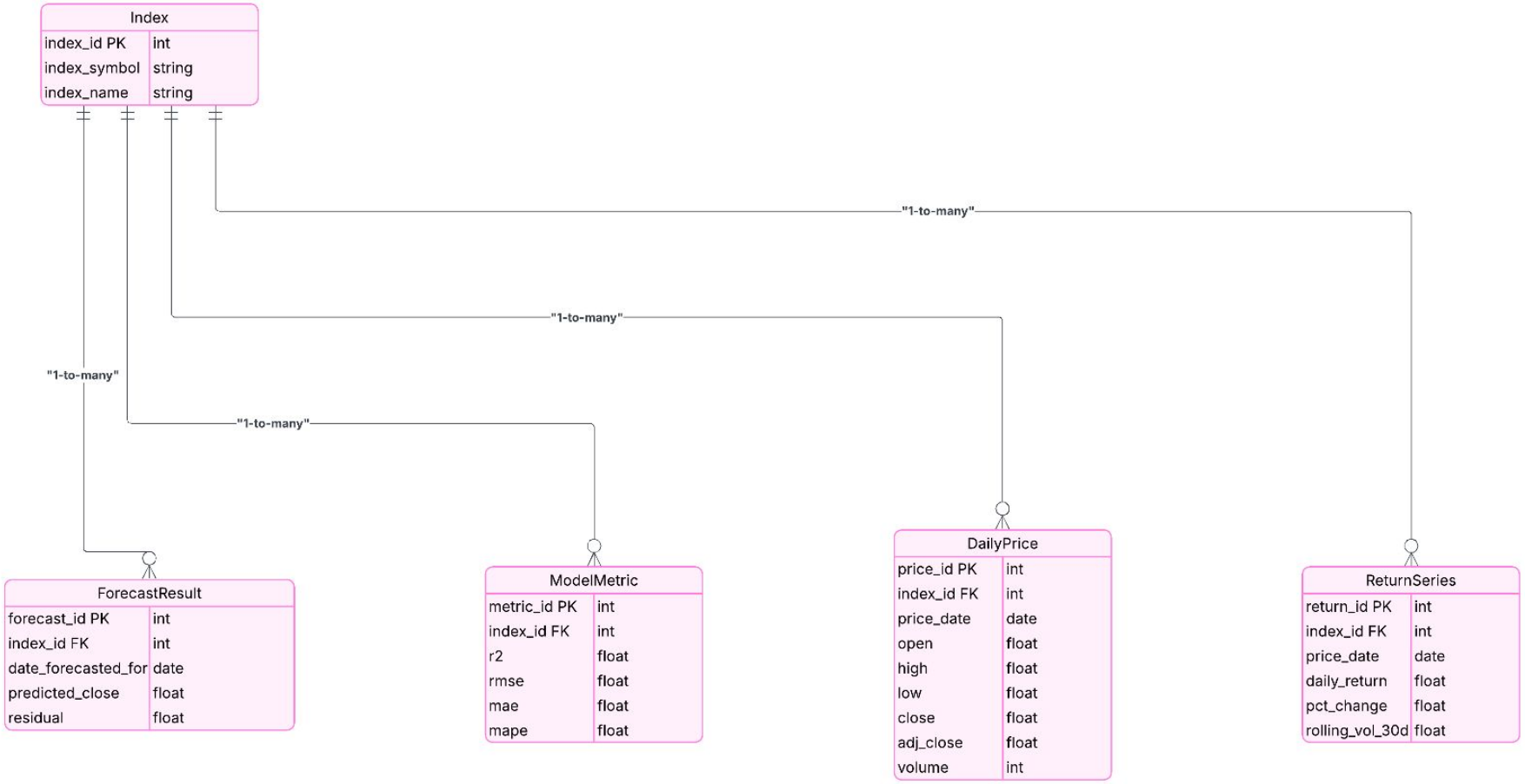
Shows payout trends across companies

Adds context to index performance

```
r = (requests.get("https://api.massive.com/v3/reference/dividends", headers={"Authorizatio
print(json.dumps(r, indent=2, ensure_ascii=False))
```


```
    {
      "cash_amount": 0.48975694000000003,
      "currency": "USD",
      "dividend_type": "CD",
      "ex_dividend_date": "2030-12-13",
      "frequency": 4,
      "id": "E396fbae341a40e1373ea57ce984c386f06778209996e5ef713783aa9455588bc",
      "pay_date": "2030-12-31",
      "record_date": "2030-12-15",
      "ticker": "GECCG"
    },
    {
      "cash_amount": 0.484375,
      "currency": "USD",
      "dividend_type": "CD",
      "ex_dividend_date": "2030-09-13",
      "frequency": 4,
      "id": "Edd95a7bbac516897f19e62104fce597fc22bed2b4fa9a88f55dba4e6b90cb84b",
      "pay_date": "2030-09-30",
      "record_date": "2030-09-15",
      "ticker": "GECCG"
    },
    {
      "cash_amount": 0.51719,
      "currency": "USD",
      "dividend_type": "CD",
      "ex_dividend_date": "2030-07-15",
      "frequency": 4,
      "id": "E423cf49e82932f378c27034ec2bd6fe3830606caa42d3053ca794e420136ef1b",
      "pay_date": "2030-07-30",
      "record_date": "2030-07-15",
```

THE DATA MODEL



BUSINESS INTELLIGENCE

Business Intelligence Dashboard

- Live Visualizations Powered By:
  - **DailyPrice** – Historical trend lines
  - **ForecastResult** – Predicted vs Actual performance
  - **ReturnSeries** – Stability vs risk analysis
  - **ModelMetrics** – KPI tiles (R^2, RMSE per index)
  - **Index** Filters – Switch between indices dynamically

APPLICATION STACK

THE STACK

Data Source: Polygon.io

Processing: Python (Pandas (cleaning), scikit-learn (regression))

Storage: PostgreSQL

Visualization: Tableau

Q&A