BOARDROOM SENTIMENT

Reading the Room

Group 7

Agenda

Our Team

Andrew Carson

Manuel Parra

AlGhalia Alsammak

Jie Zhu

Syed Shah

You know me ...from work.

Our Goal

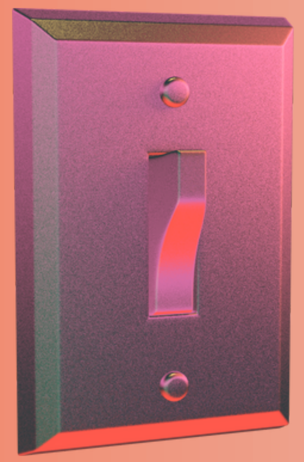
Our goal is to predict stock price performance by analyzing the sentiment of public earnings call transcripts. we used advanced generative AI and ML techniques to uncover correlations between sentiment, EPS growth, and price movements. By integrating these insights with financial metrics, the project provides sentiment-driven investment recommendations to support informed decision-making.

Solution

This project is focused on predicting the stock price growth based on earnings data, sentiment analysis from earnings call transcripts, and financial features. We utilized two models: Random Forest and XGBoost, to predict stock growth and made a comparative analysis between these two models to select the most effective one.

Tools & Technologies

Multiple technologies and statistical models are used to build the The insider Application



A Python

B Pandas, Numpy

C NINJA ,Alpha Vantage , Open AI APIs

D yfinance

E Stremlit

F Github

G Sklearn

H Matplotlib

I Seaborn

J Gradio

K XGBoost

L RandomForest

Data collection

Timeframe: Data was collected from 2019 to 2023 for each ticker. This provides a comprehensive historical view of financial data and stock price performance across different economic conditions.

Ticker Selection: The tickers were selected across various sectors to ensure diversity and represent different industries. Specifically, we chose two tickers from each of the 11 sectors total 22 companies.

- information Technology: Apple (AAPL), Microsoft (MSFT)
- Healthcare: Johnson & Johnson (JNJ), Pfizer (PFE)
- Financials: JPMorgan Chase (JPM), Bank of America (BAC)
- Consumer Discretionary: Amazon (AMZN), Tesla (TSLA)
- Consumer Staples: Procter & Gamble (PG), Coca-Cola (KO)
- Energy: ExxonMobil (XOM), Chevron (CVX)
- Industrials: Boeing (BA), Caterpillar (CAT)
- Materials: Linde (LIN), Dow Inc. (DOW) Real Estate: American Tower (AMT), Simon
- Property Group (SPG) Utilities: NextEra Energy (NEE), Duke Energy (DUK) Communication
- Services: Alphabet (GOOGL), Meta (META)

Data Cleaning & Analysis

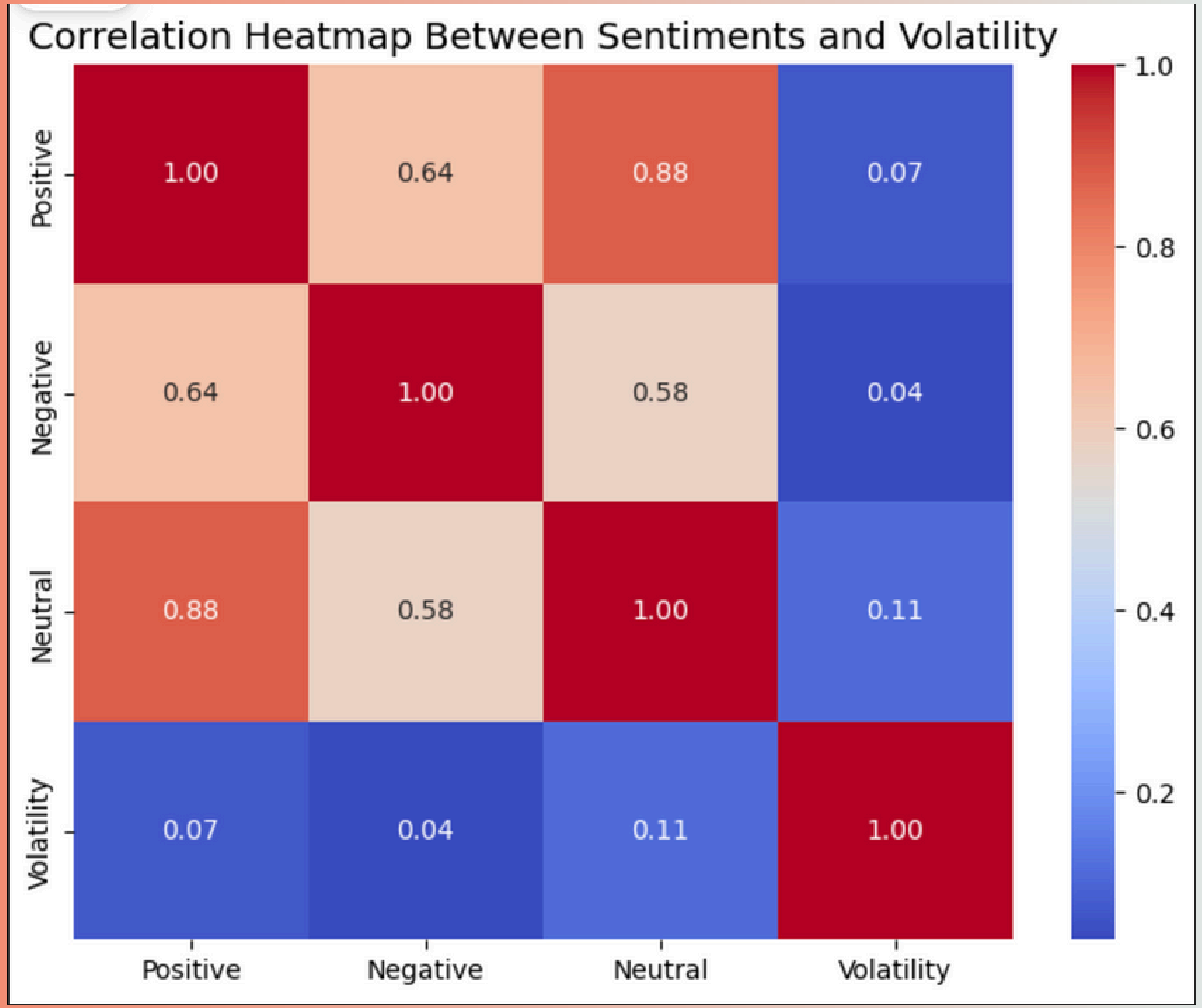
<u>Text Cleaning:</u> The transcripts are cleaned by removing unnecessary formalities, and opening remarks, This is essential for the sentiment analysis step.

<u>Feature Engineering:</u> Created features like EPS surprise (difference between reported and estimated EPS), Sentiment (positive, neutral, negative), and stock growth (7-day growth, 14-day growth, etc.).

<u>Sentiment Analysis</u>: We analyzed the earnings call of 440 transcripts and by using OpenAI's GPT-3.5 Turbo model to classify the sentiment as either positive, negative, or neutral with specified exact weights.

<u>Financial Analysis</u>:
EPS Analysis, stock price growth over multiple timeframes (7-day, 14-day, 30-day) to understand how stock prices move in response to earnings reports and sentiment.

Correlation between Sentiment & market Volatility:



- Sentiment Analysis and Market Volatility: We explored the correlation between sentiment and market volatility (measured by stock price fluctuations).

- The analysis showed that strong positive sentiment tends to lead to more stable stock price movements, while negative sentiment can cause sharper fluctuations and higher volatility in the market.

- Correlation with Volatility:
- Positive Sentiment: 0.07
- Negative Sentiment: 0.04
- Neutral Sentiment: 0.11

Model selection

We evaluated two machine learning models for predicting stock price growth: **Random Forest**, an ensemble method that handles nonlinear data and overfitting well, and **XGBoost,** a gradient-boosting framework known for its high performance and accuracy with structured data.

Train-Test Split: 80% of the data is used for training, and 20% is used for testing.

Fit the Model: we fit on the training data using the .fit() method:

We trained the model by passing the training features (X_train) and the target variable (y_train).

- X = data[['7_day_growth', '14_day_growth', '30_day_growth', 'Quarterly_Growth', 'Price_Momentum', 'EPS_Growth', 'sentiment', 'reportedEPS', 'estimatedEPS']
- y = data['7_day_growth']

```python
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

# Assuming your dataset is loaded into the 'data' DataFrame
# Feature Engineering (include the interaction features)
data['Positive_EPS_Surprise'] = data['Positive'] * data['EPS_Surprise']
data['Neutral_EPS_Surprise'] = data['Neutral'] * data['EPS_Surprise']
data['Negative_EPS_Surprise'] = data['Negative'] * data['EPS_Surprise']

# Prepare the feature set (including the interaction terms)
X = data[['7_day_growth', '14_day_growth', '30_day_growth', 'Price_Momentum', 'EPS_Growth',
          'Positive_EPS_Surprise', 'Neutral_EPS_Surprise', 'Negative_EPS_Surprise']]

# Target variable (you can change this based on your prediction target
y = data['7_day_growth']

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a pipeline with a StandardScaler for feature scaling
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('model', RandomForestRegressor(n_estimators=100, random_state=42))
])

# Train the model
pipeline.fit(X_train, y_train)

# Make predictions
y_pred = pipeline.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print the evaluation metrics
print(f"Mean Squared Error (MSE): {mse}")
print(f"Mean Absolute Error (MAE): {mae}")
print(f"R-squared (R²): {r2}")

# Save the model pipeline if necessary (for future use)
import pickle
with open('best_model_pipeline.pkl', 'wb') as f:
    pickle.dump(pipeline, f)
```

Model Evaluation

Metric	Random Forest	XGBoost
Mean Squared Error (MSE)	0.3360	0.4791
Mean Absolute Error (MAE)	0.1148	0.1944
R-squared (R^2)	0.9793	0.9705

Model Fine-Tuning

- **Hyperparameter Tuning**: We performed hyperparameter tuning using RandomizedSearchCV to find the best configuration for both models. This step helped optimize parameters like the number of estimators, maximum depth, learning rate, and others.

Regression Performance Metrics

- **Mean Squared Error (MSE): Indicates the average squared difference between predicted and actual values. A lower MSE means more accurate predictions. The Random Forest model has a lower MSE, indicating better accuracy.**

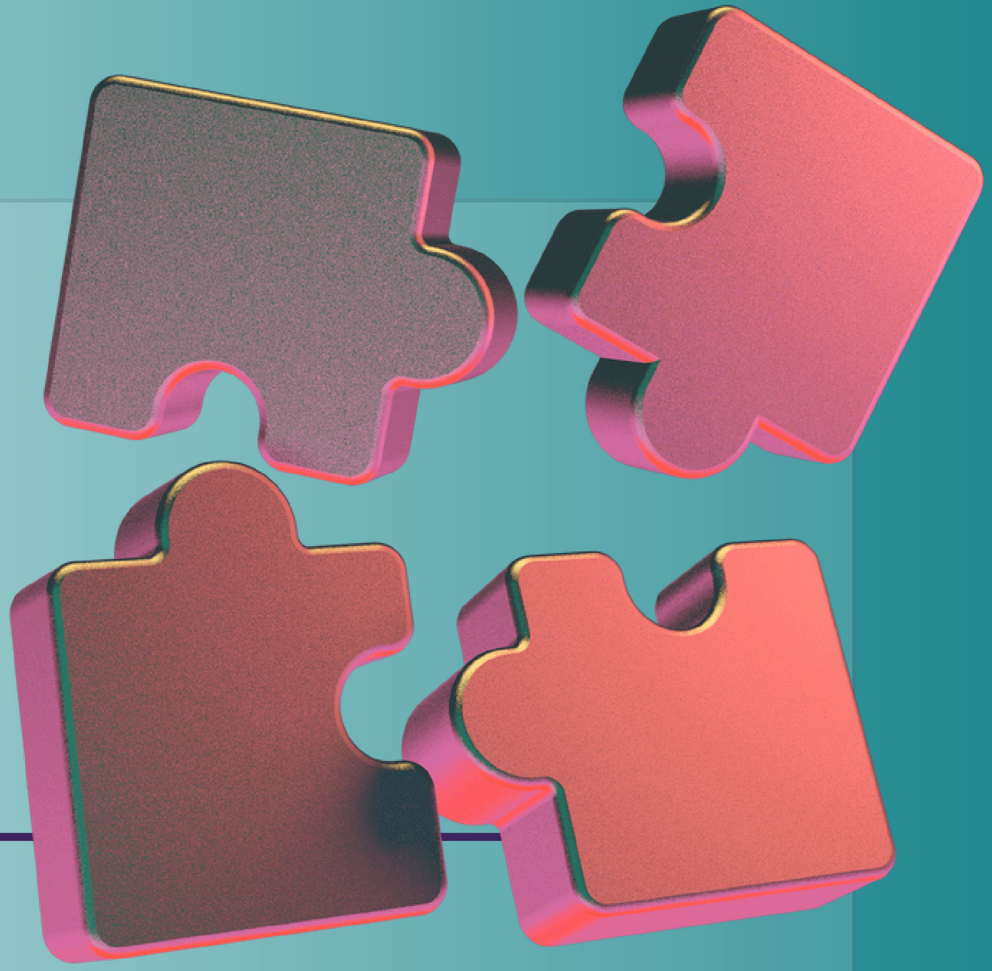
- **Mean Absolute Error (MAE): Measures the average of the absolute errors between predicted and actual values. A lower MAE indicates closer predictions to actual stock price growth. Random Forest again outperformed XGBoost in this metric.**

-

- **R-squared (R^2): Represents the proportion of the variance in the dependent variable (stock growth) that can be explained by the independent variables. A higher R^2 value indicates better model fit. Random Forest had a higher R^2, suggesting it captured more of the market's underlying dynamics.**

For regression problems, the goal is to predict continuous numerical values

Conclusion



While sentiment analysis of board meetings provides valuable insights into the company's internal perspective, it should not be the sole basis for predicting stock performance or volatility. Rather, it can be considered one piece of the puzzle in a more comprehensive investment strategy that incorporates financial metrics, market trends, and economic data.
For investors and analysts, understanding the tone and sentiment within board meetings could offer some indication of company confidence or concerns, which may influence investor sentiment and behavior. However, it should be paired with other fundamental and technical analysis tools.

- The Random Forest model's success demonstrates that using multiple features and a strong machine learning approach can improve stock prediction accuracy.

Lessons Learned & Next Steps

Challenges faced

- Difficulty in finding free APIs that provide precise and accurate financial information without access limitation.
- Complexity in integrating multiple datasets & APIs seamlessly.
- Most models struggle with analyzing financial transcripts due to their inherently neutral tone.

Future Improvements

- Incorporate more data: We can expand the dataset to include more features like social media sentiment, news sentiment, historical price trends, or macroeconomic factors.
- Model Optimization: Further hyperparameter tuning and feature engineering can be done to improve model accuracy.
- Real-Time Data: Integrate real-time stock price and earnings call transcript data for up-to-date predictions.

DEMO

We created an interactive dashboard that allows users to analyze stock performance based on earnings data, sentiment, and growth trends over time. The visual outputs (charts) make it easier for users to interpret the stock's behavior and make informed decisions.
Key Features of the Code:

- Real-time earnings data retrieval from Alpha Vantage.
- Sentiment analysis using OpenAI’s GPT model.
- Stock growth calculation with yfinance.
- Visual chart generation (Stock price, growth, and EPS comparison).
- Gradio interface for easy user interaction.

This system can be very useful for traders or analysts who want to understand stock behavior based on earnings calls and sentiment analysis.