Erdos Institute Winter 2024

April 26, 2024

Predicting Stock Prices after Earnings Calls

Team: Yearning for Earnings

Members: Shabarish Chenakkod, Jasper Liang, Tejaswi Tripathi, Shravan Patankar

Overview

- Earnings calls often lead to **volatility in stock prices.**
- We want to create **machine learning models** that **predict percentage changes** in stock prices surrounding earnings calls based on
    - **sentiment analysis** of earnings call transcripts
    - earnings and revenue data
    - stock prices and volume before the earnings
- Detect factors which influence stock price the most

NVDA earnings call working in favor of its stock price

META earnings call negatively impacts tech sector

Datasets

We selected **98 companies** from S&P 500 with a **6-year** earning period. In total we have **over 2000** data points. Sample companies include:

Data were gathered using Seeking Alpha API from [rapidapi.com]() and Yahoo Finance.

Earnings Call Transcripts

Web Scraping using Seeking Alpha API obtained from Rapid API

+

Earnings and Revenue Data

Web Scraping using Rapid API

1. Earnings per share (EPS)
2. Earning and revenue surprises

+

Stock Prices and Volume

Yahoo finance python package

1. Stock prices several time points before and after earnings
2. Average volume 50 days before each earning

Data Processing

- Instead of looking at **absolute changes**, we look at **percentage changes**.
- Key features include: **average_volume_50_days, % Change Revenue, % Change EPS Normalized** and several sentiment scores:
  - **Financial_performance_score**
  - **Market_position_score**
  - **Strategic_direction_score**
  - **Operational_aspects_score**
  - **Financial_indicators_score**
  - **Risks_challenges_score**
  - **Economic_factors_score**
- Target Feature: **perc_change_next_prev** (percentage of stock price changes next day to previous day of earnings call)

We label the data using
Symbol + Year + Quarter

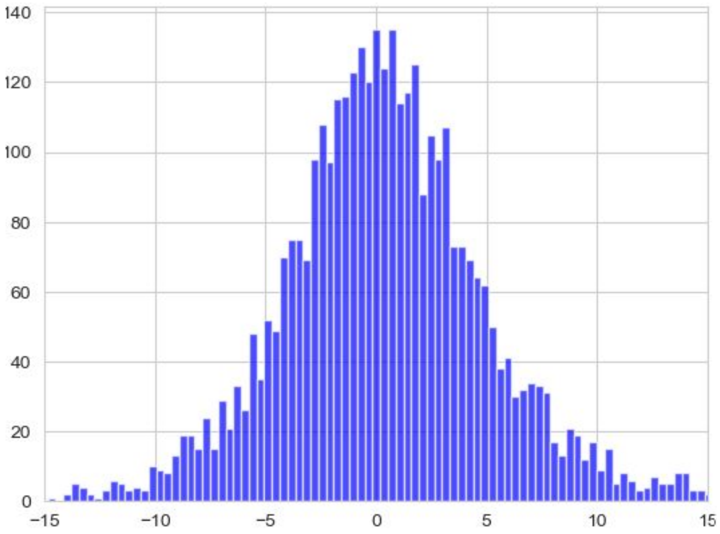
AAPL2019Q1
AAPL2020Q1
AAPL2021Q1
AAPL2022Q1
AAPL2023Q1
...
XOM2019Q3
XOM2020Q3
XOM2021Q3
XOM2022Q3
XOM2023Q3

Exploratory Data Analysis (EDA)

Scatter plot of percent change in **Revenue** vs **Stock Price** on earnings

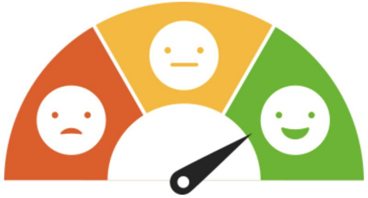
x = Stock Price Change

Histogram of **percentage change in stock price** after earnings



x = Percent Stock Price Change, 84 bins

Extracting Features from Earning Call Transcripts



Sentiment Scores

1. **Organized keywords into seven categories.**

Example:

financial_performance_keywords = {revenue, profit, loss, earnings, sales, expense, cost,....}

market_position_keywords = {market, share, grow, growth, decline, competitive, demand,....}

risks_challenges_keywords = {risk, challenge, uncertainty, regulation, legal, compliance, issue,....}

2. **For each category, we extracted sentences (with context) and computed average sentiment scores using VADER.**

spaCy

NLTK

Models

- **Baseline Model:** Predicts no change from previous day.
- **Linear Regression**

- **XGBoost** (parameters tuned using GridSearchCV)
- **Neural Network**

- Logistic Regression
- Neural Network (Classification)

Linear Regression

Training - Test Split: 0.8 - 0.2

Stratified by **symbol**

⬇

Run 1000 times, after scaling features

⬇

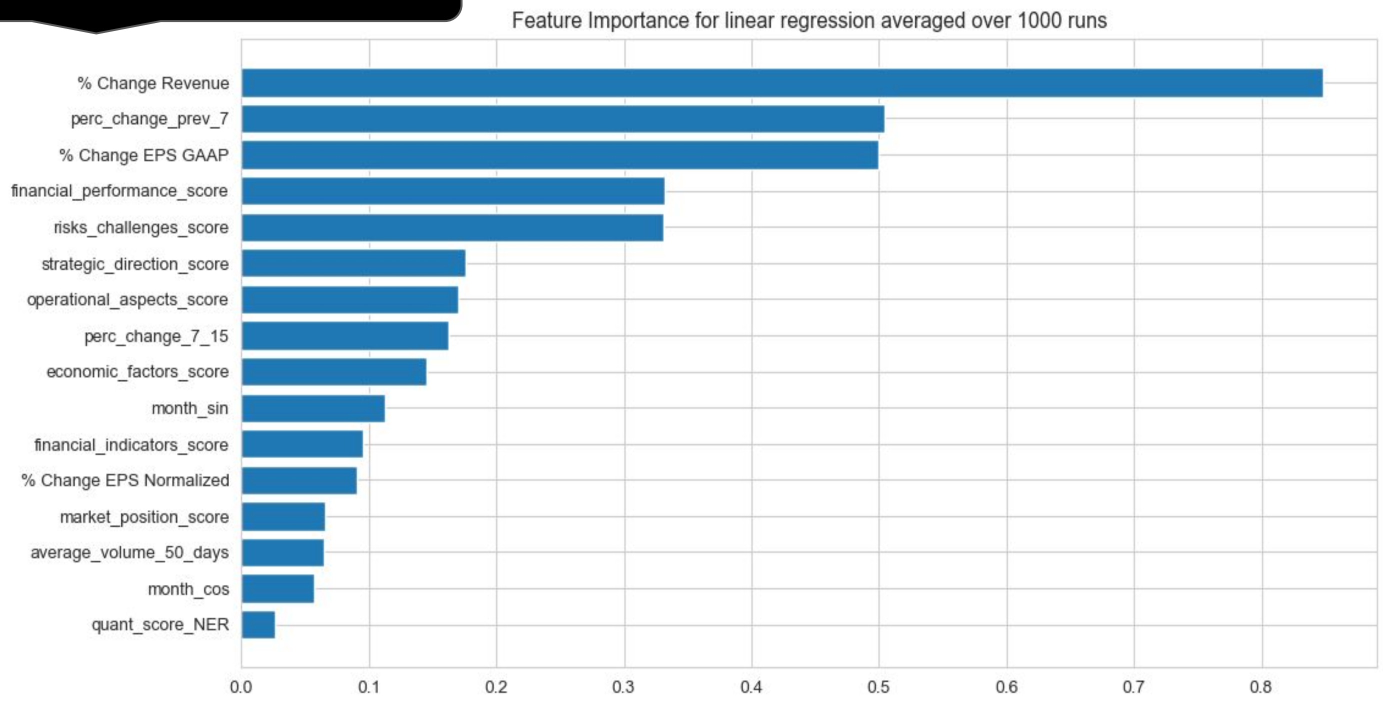
Record Actual and Predicted Values

Remark:

Technically, we should only be doing a time based split. However, we found that the performance is similar by doing that.

Linear Regression

Feature Importance



XGBoost

Fine tune parameters using **GridSearchCV**

↓

Run 1000 times

↓

Record Actual and Predicted Values

Parameters tuned:

- alpha (L1 regularization)
- lambda (L2 regularization)
- n_estimators (number of trees used)
- max_depth of a tree
- learning_rate

Results

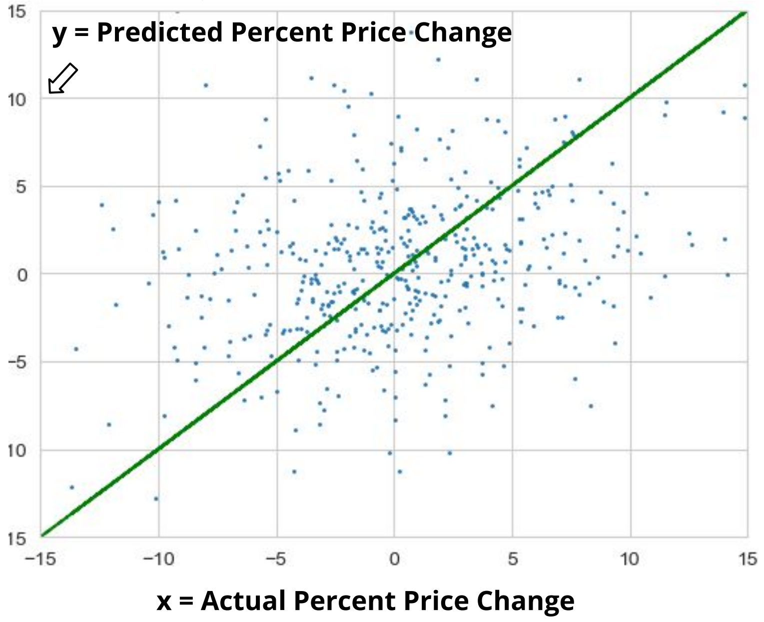
The numbers are **means** of multiple runs

	MSE	Correlation
Baseline	31.16	N/A
Linear Regression	24.16	0.18
XGBoost	23.85	0.22
Neural Network	35.62	0.31

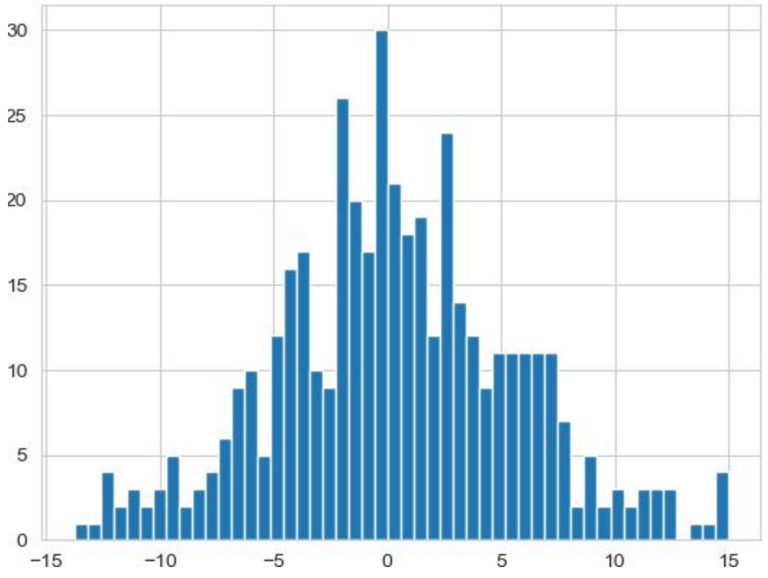
Results Visualization

Both are based on Neural Network

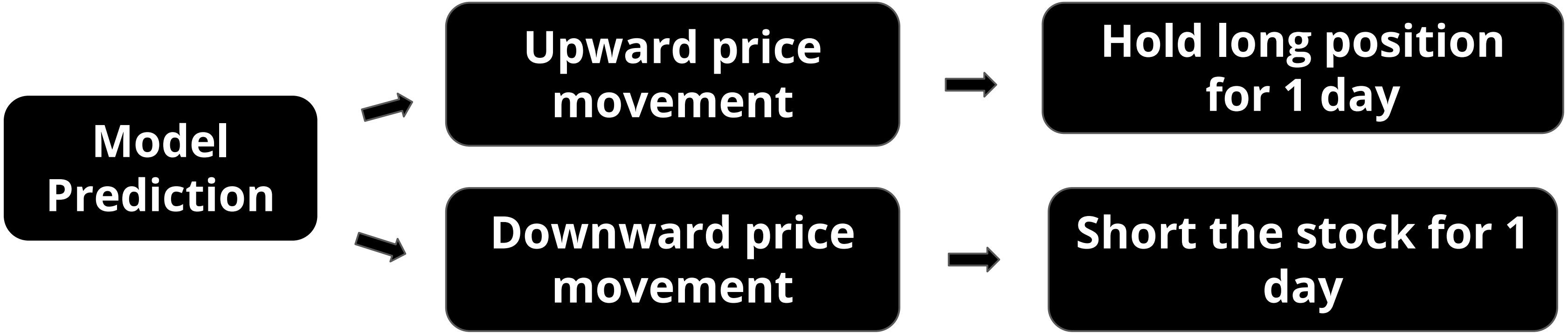
Scatter plot **Predicted percent price changes** vs **Actual**



Histogram of **Residuals**



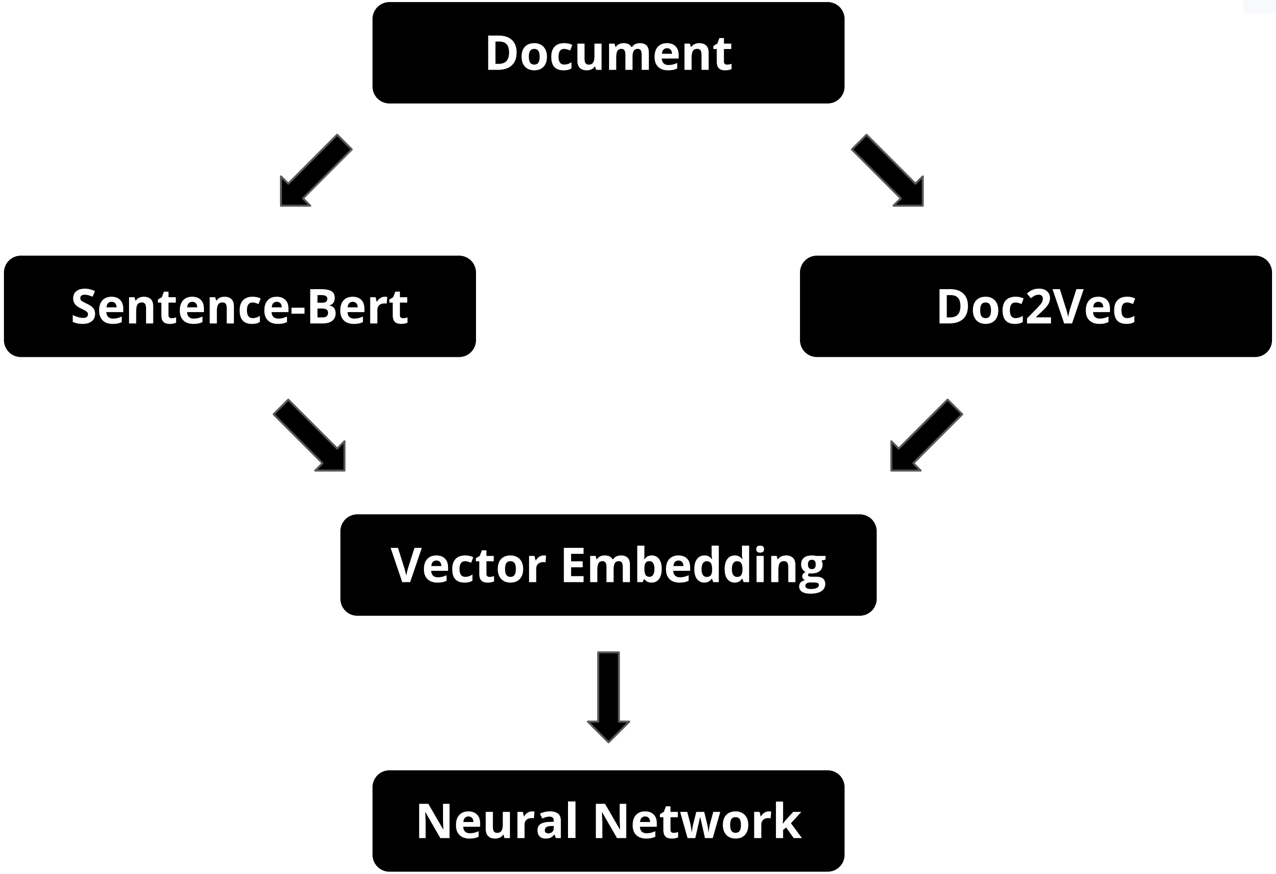
A Simple Trading Strategy

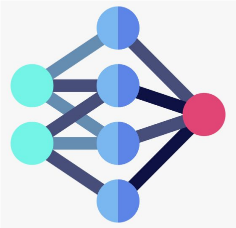


Results:

- We obtain **returns of 4.8%** over trades around 4 earnings call days.
- In comparison, blindly buying the stock on the day of earnings call and selling it the next day gives **a return of 2%.**

Future Research





Acknowledgement

Thanks to Roman Holowinsky, Steven Gubkin, Alec Clott and the Erdös Institute for their support throughout the Winter 2024 boot camp.

Thanks to Dyas Utomo and Andrew McMillan for the mentorship.

Special thanks to Sridhar Venkatesh for his help during the beginning stage of the project.

UNIVERSITY OF MICHIGAN

UIC