Machine Learning Based Classification of Financially Strong Firms Using Financial Fundamental Metrics

-Chandan Maka

MSDS – 692

March 2026

Introduction

- This project aims to classify the selected company as financially strong or weak on the basis of the extracted financial statements and fundamental financial ratios from its annual report of the company i.e 10-k report with the help of machine learning.
- It can really be helpful for financial decison-making to an entry-level investor and financial analyst to decide whether the company is financially stable or not based on its annual report and extracted financial information.

Problem Statement

- For any investor or financial analyst, deciding to choose company for investment. The first thing they think about is **whether the company is financially stable or not.**
- This can be manually done by looking at the corresponding company's annual financial report but it can really be challenging and sometimes they might **overlook or misclassify** some of the financial information from the report.
- There should be a **systematic approach** to classify the selected company as financially stable or weak just by providing the annual report(10-k form).
- **Hence, this project uses machine learning to classify the selected company using extracted info from annual report of that company.**

Data Source

- Used API from the Financial Modeling Prep to collect the financial statement from annual report(10-k form) of publicly traded company in US stock market
- It provides income statements, balance sheets and cashflow statements
- Can access all the fundamental metrics from those data that are necessary for the project
- Collected latest financial statements of **17513 companies**
- Link : https://site.financialmodelingprep.com/datasets

Data Collection

- Used Financial Modeling Prep API to collect
  - Income Statements
  - Balance Sheet Statements
  - Cashflow Statements

Data Preparation & Cleaning

- Handling null values (total of **561** records had null values which was significantly less so, I dropped those rows hence, total rows after dropping null values was **16952** records)
- There were **132** columns after merging the **3 three statements** but I only need those columns that are important for my classification model
- **8 different category metrics will be used for generating financial ratios in feature engineering phase so, I only need major columns that fulfill those requirement metrics**

Data Preparation & Cleaning

- 8 Financial categorical metrics are

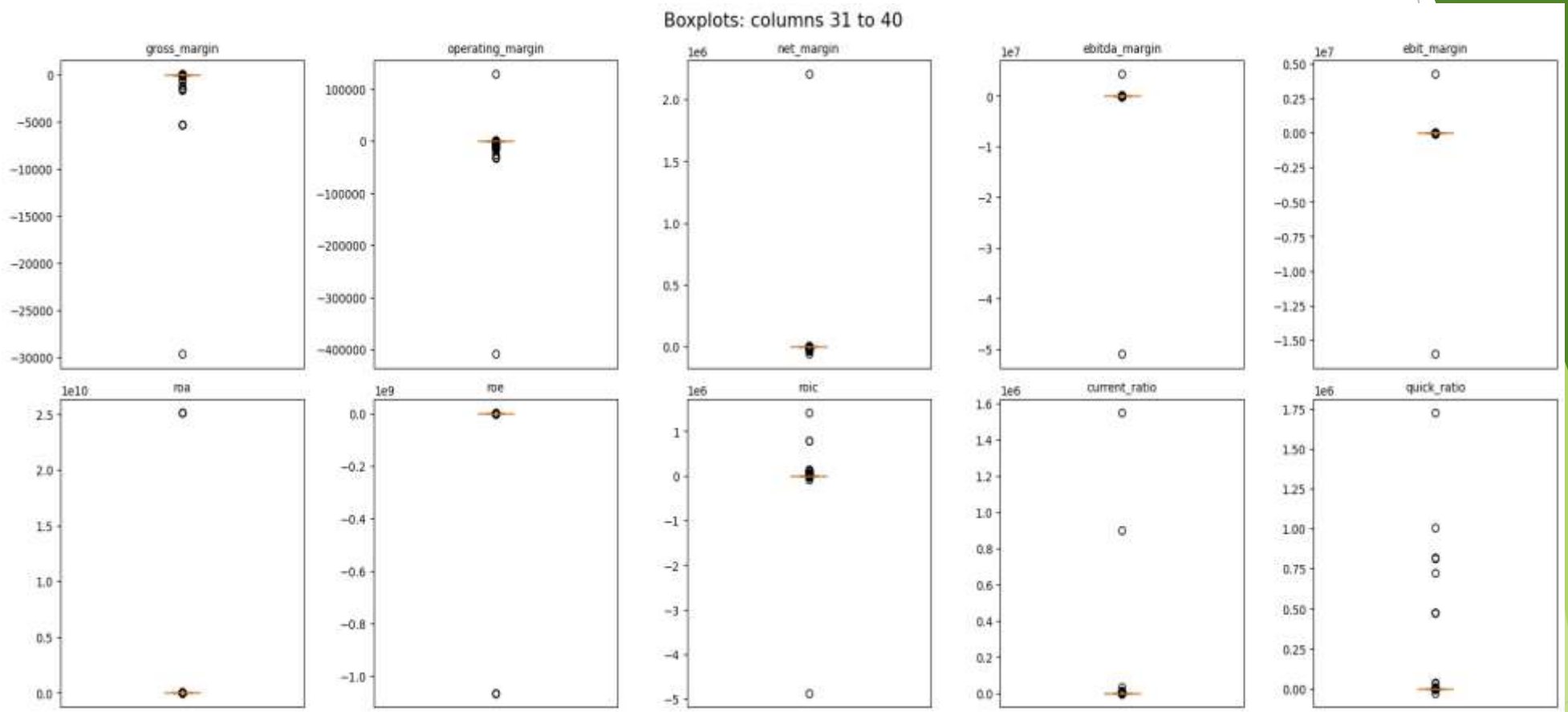
Profitability	Liquidity	Leverage	CashFlow Quality
Effeciency	Capital Investment	Cost Structure	ShareHolder value

- Total of 32 columns were used for further processing
- Some of the columns name are "filingDate","symbol","revenue","costOfRevenue","grossProfit","operatingIncome","netIncome","ebitda","ebit","totalAssets","totalStockholdersEquity" and so on……

Metric Category	Features
Profitability Metrics	gross_margin, operating_margin, net_margin, ebit_margin, ebitda_margin, roa, roe, roic
Liquidity Metrics	current_ratio, quick_ratio, cash_ratio, working_capital
Leverage / Solvency Metrics	debt_to_equity, debt_to_assets, equity_ratio, net_debt_to_ebitda, interest_coverage
Efficiency / Activity Metrics	asset_turnover, inventory_turnover, receivables_turnover
Cash Flow Metrics	ocf_ratio, ocf_to_revenue, fcf_margin, cf_to_net_income, capex_to_revenue, capex_to_ocf
Per-Share Metrics	fcf_per_share, book_value_per_share
Cost Structure Metrics	opex_ratio

Feature Engineering

- 29 financial ratios were computed with the help of existing features under 8 categorical financial metrics
- **Notable problem** that was faced was division by zero, instead of keeping value infinite I decided to keep **null**
- Eg : gross_margin = grossProfit / revenue

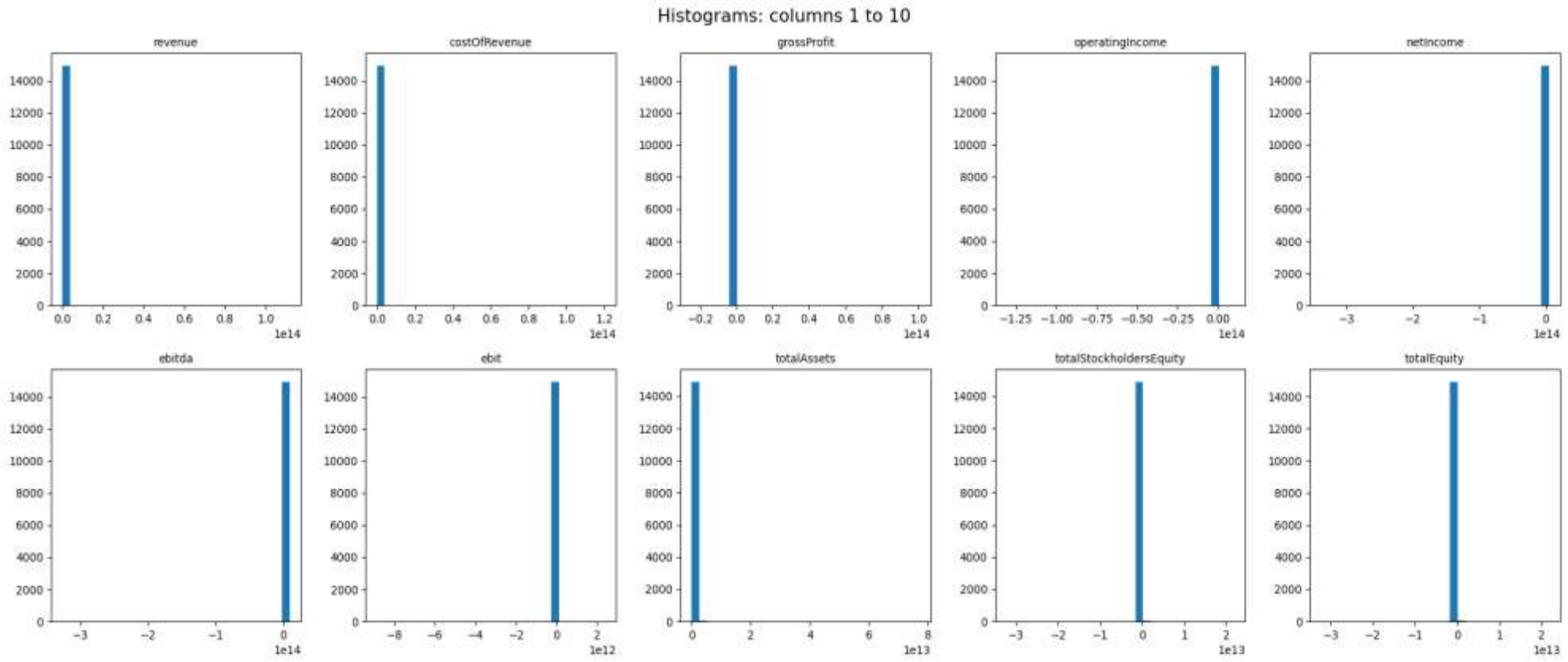


EDA & Visualization

Boxplot

EDA & Visualization

- Instead of treating the extreme values as outliers and removing them they are important data
- Some companies revenue can be in billions of dollar these are extreme values but not outliers they are valuable information so instead of removing them, we use one procedure for clipping those values in range that process is **Winsorization**
- I applied winsorization by limiting each selected feature to the 1st and 99th percentile range



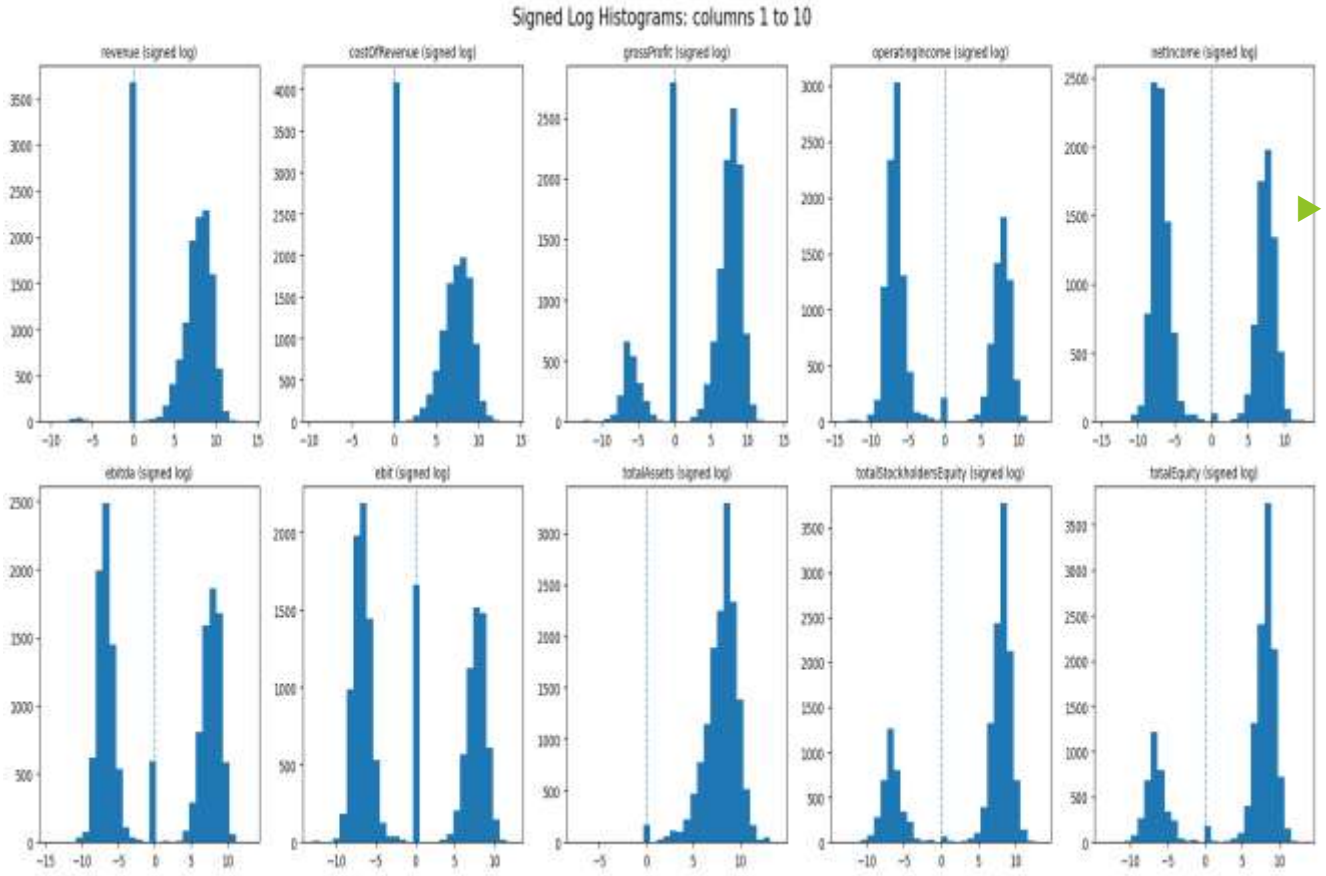
EDA & Visualization

Histogram

EDA & Visualization

- The problem that histogram is showing in single bar is due to extremely large range and highly skewed data.
- most companies have relatively small values while a few very large companies have values that are many times larger
- Solution is **Signed log transformed Histogram**

EDA & Visualization



- Signed Log Transformation

KNN Imputation

- During feature engineering, ratios with null values were also created
- Ratios that had denominator zero gives infinite value so we replaced with null
- Eg : gross_margin = grossProfit /revenue
- We could have done mean or median imputation method to fill the null values but the distribution of features was highly skewed and bimodal so, the best solution was to use **KNN imputation**

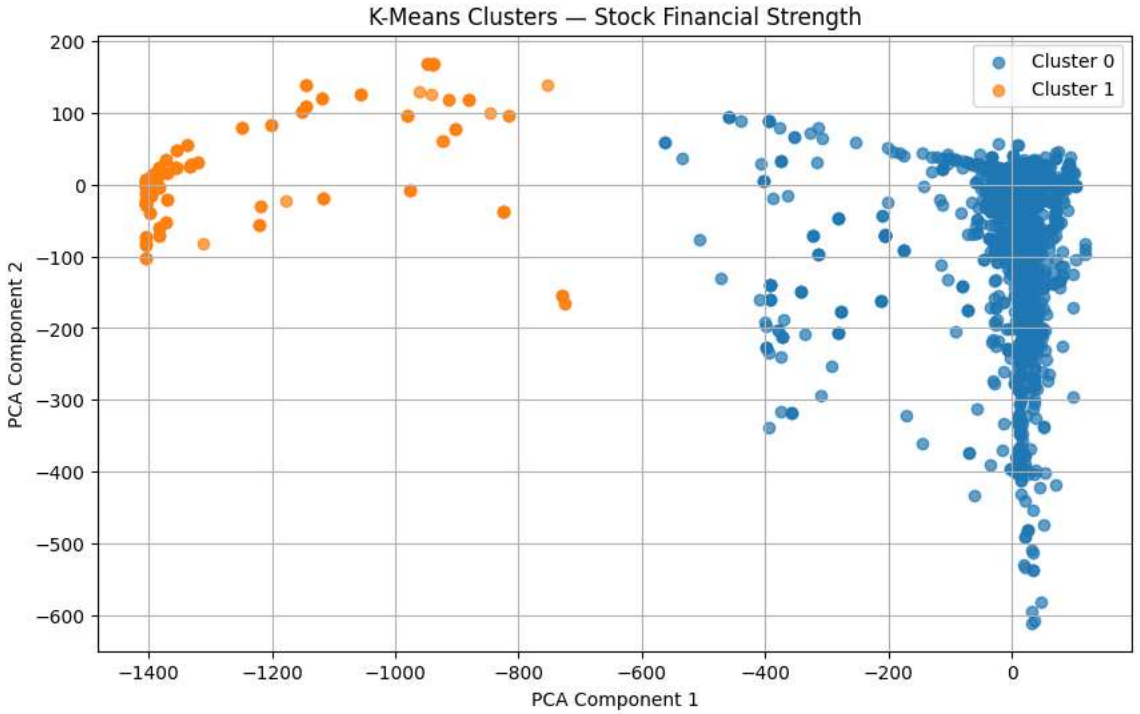
Data Labeling

- For training the model I need to have labeled data where the strong financial company should be labeled as 1 / “strong” and the other should have label as 0 / “weak”
- Tried Method :
  - K-means clustering (Didn’t work)
  - Binary Quantile method (Worked)

Data Labeling

- Used unsupervised clustering method with k = 2 so that the data separates into 2 clusters but the **result was not good and highly biased**

Cluster 0	Cluster 1
14784	180



Data Labeling (Binary Quantile Method)

Features having higher value that shows company as strong	gross_margin','operating_margin','net_margin', 'ebitda_margin','ebit_margin',etc
Features having higher value that shows company as weak	'debt_to_equity','debt_to_assets', 'net_debt_to_ebitda', 'capex_to_revenue',etc

- What it basically does is calculates the financial score for each selected feature as 0 or 1 , 0 means weak and 1 means strong
- The value 0 or 1 is calculated for each selected feature based on the rule
- def **binary_quantile_label**(*feature*, *higher_is_better*=True):

```
median = feature.quantile(0.50)
if higher_is_better:
    return feat_score = feature >= median ? 1: 0
else:
    return feat_score = feature >= median ? 0 :
1
```

- Now we get the financial_score for each record by averaging the feat_score
- We label "strong" if financial_score >= financial_score.median else "weak"

Data labeling

- Example
  - Gross_margin.median = 0.5
  - Debt_to_equity.median = 0.5
  - Financial_score.median = 0.5
    - For company A

Output from Binary Quantile Method	
Stock Label	
Strong	7574
Weak	7390

		Feat_score	Financial_score
Gross_margin,higher_is_nice	0.5	1	(1+0)/2 = 0.5
Debt_to_equity,lower_is_nice	1	0	

Label
Financial_score(0.5) >= financial_score.median so, “strong”

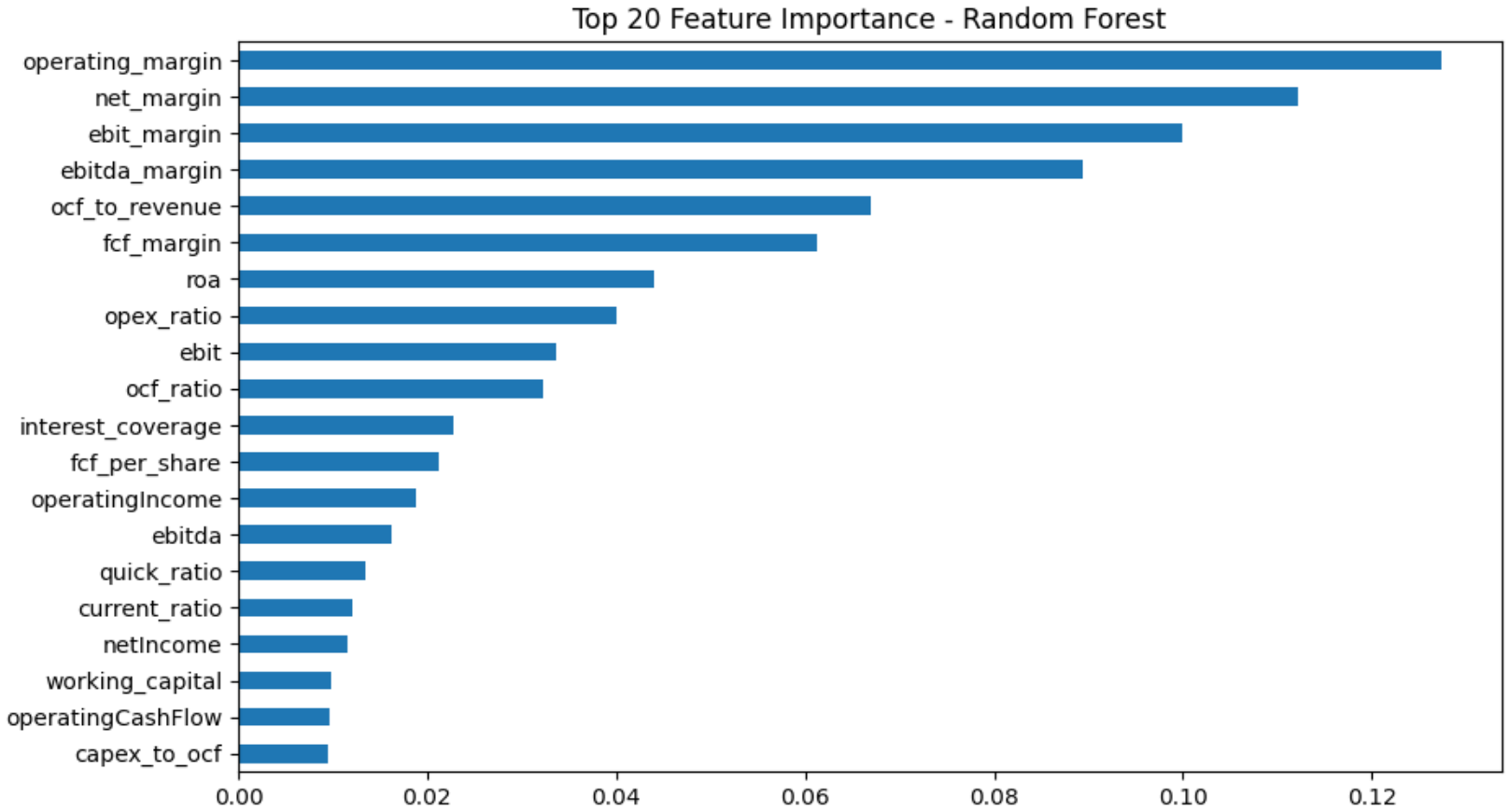
Model Training and Interpretation

- Used Random Forest model to train the data
- Accuracy: 0.9702639492148346
- Classification Report

	precison	recall
0	0.96	0.98
1	0.98	0.96

Model Training & Interpretation

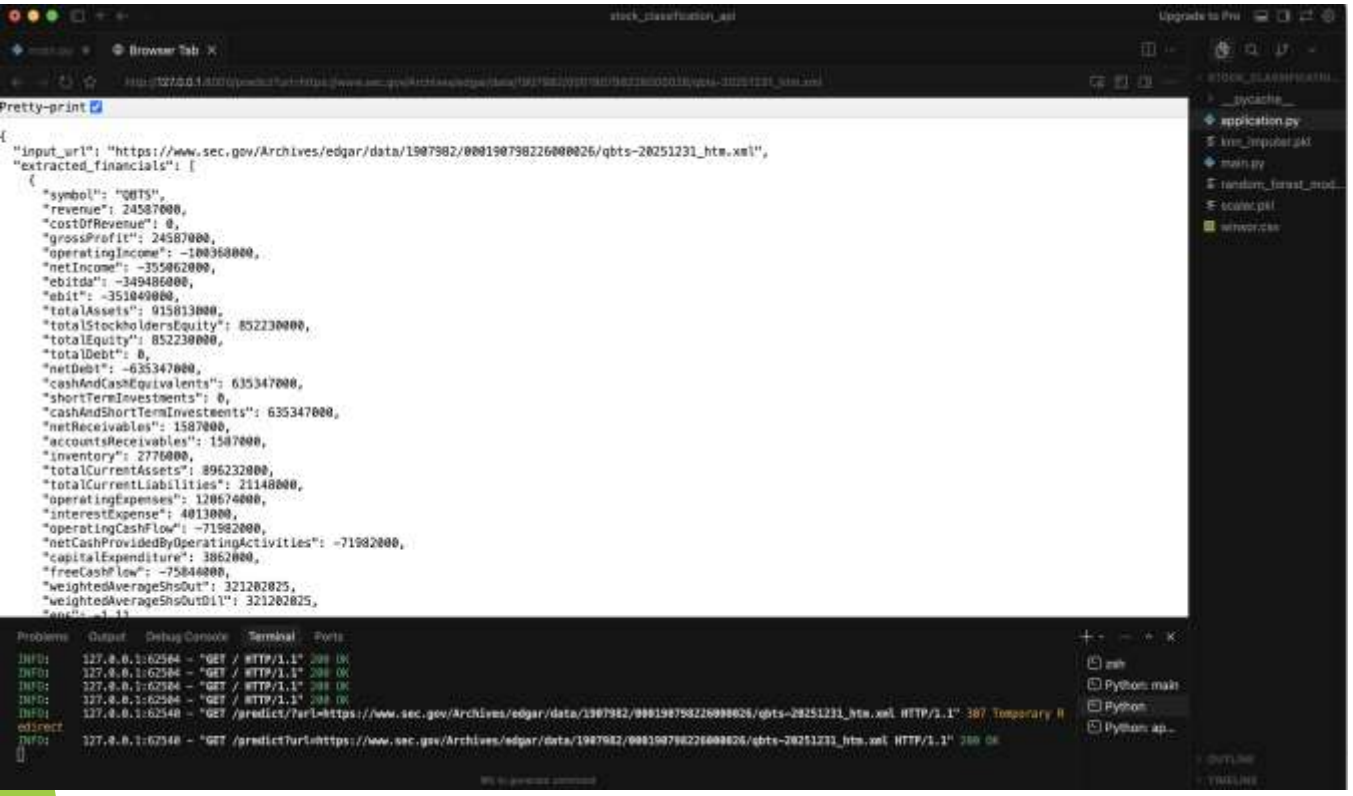
- Feature Importance



Annual Report (10-K form) Parser

- The parser takes the interactive xml file of annual report and parses the related information
- It extracts all the required financial statements from the xml file and returns the required data that can be used for classification of that selected firm.

Output API



- Company : D-wave quantum Inc.
- "prediction": [0],
- "predict_proba": [[0.959521480693368, 0.0404785193066 32]]

Thank You!!!