Noor-Aldean Deek

noordeek@bu.edu

Abstract

This project applies machine learning techniques (Gradient Descent and K-means Clustering) to analyze the financial data of companies over two decades. Gradient Descent is used to predict a company’s market value based on key financial metrics, revealing that assets, net income, and revenue are the strongest drivers of value. K-means Clustering groups companies into distinct financial categories. The results highlight how these models can be leveraged for financial analysis and decision-making, providing both predictive and exploratory insights into corporate performance.

Noor-Aldean deek

MET CS 777 Term Project: Analyzing Financial Data Using Gradient Descent and K-Means Clustering

MET CS 777 Term Project

Table of Contents

[1. Project Overview 2](#_Toc179862917)

[2. Data Collection & Dataset Description 2](#_Toc179862918)

[3. Data Cleaning & Preprocessing 3](#_Toc179862919)

[Removing Invalid or Missing Values: 3](#_Toc179862920)

[Imposing Conditions for Inclusion: 3](#_Toc179862921)

[Market Value Calculation: 3](#_Toc179862922)

[Final Dataset Statistics: 3](#_Toc179862923)

[4. Gradient Descent: Regression Analysis 4](#_Toc179862924)

[Objective: 4](#_Toc179862925)

[Methodology: 4](#_Toc179862926)

[Results: 4](#_Toc179862927)

[Interpretation: 5](#_Toc179862928)

[5. K-Means Clustering: Grouping Companies Based on Financial Metrics 5](#_Toc179862929)

[Objective: 5](#_Toc179862930)

[Methodology: 5](#_Toc179862931)

[Cluster Results: 6](#_Toc179862932)

[Cluster Centers: 6](#_Toc179862933)

[Interpretation: 6](#_Toc179862934)

[6. Conclusion & Future Work 6](#_Toc179862935)

[Future Work: 7](#_Toc179862936)

[Appendices 7](#_Toc179862937)

[Full Dataset Link 7](#_Toc179862938)

[Spark Screenshots 7](#_Toc179862939)

[Gradient Descent 7](#_Toc179862940)

[K-Means 8](#_Toc179862941)

[Code Output 8](#_Toc179862942)

[Gradient Descent 8](#_Toc179862943)

[K-Means 11](#_Toc179862944)

# Project Overview

The primary focus of this project is to explore how financial metrics impact a company's market value over time, as well as to categorize companies based on similarities in their financial metrics. This is achieved using two powerful machine learning techniques: Gradient Descent and K-means Clustering.

The project began by curating a dataset comprising financial records for various companies from 1999 to 2024. These records included essential data points such as total assets, revenue, net income, liabilities, and stock prices. The data's timespan provided a valuable opportunity to explore long-term financial trends and patterns.

The project had two main goals:

1. Gradient Descent: To predict the market value of companies based on their financial metrics and assess which factors most influence market value.
2. K-Means Clustering: To group companies with similar financial attributes into clusters to better understand their financial behavior and characteristics.

This dual approach allowed for a comprehensive analysis, integrating both regression analysis for precise predictions and clustering for pattern recognition.

# Data Collection & Dataset Description

The dataset used in this project was sourced from WRDS (Wharton Research Data Services) and covered 25 years of financial data for thousands of companies. Each row in the dataset represented a company’s quarterly financials, allowing us to track the financial health of companies over time. The dataset included the following key financial metrics:

* Total Assets (atq): The sum of everything the company owns, from property to cash reserves.
* Long-term Debt (dlttq): The total amount of debt a company has that must be repaid in more than one year.
* Liabilities (ltq): The sum of a company's short- and long-term obligations.
* Net Income (niq): The company’s total earnings, accounting for expenses and taxes.
* Revenue (revtq): Total earnings from the company's operations.
* Stock Price Metrics: These include Price Close (prccq), Price High (prchq), and Price Low (prclq), which provide insights into the company's stock performance.
* Common Shares Outstanding (cshoq): This refers to the number of shares that are currently held by shareholders.

One of the key transformations made to the data was the calculation of man-made market value (man\_mkt\_val), which was computed as the product of a company’s shares outstanding (cshoq) and its closing stock price (prccq). This new metric provided a more accurate reflection of a company’s market capitalization than the original dataset’s market value column.

The initial dataset contained 1,134,506 rows, representing a vast range of companies across different sectors and time periods. However, given the nature of financial data, it was crucial to clean the dataset and remove any incomplete or invalid records. The data cleaning process, which will be discussed in detail, ultimately reduced the dataset to 695,749 rows, ensuring the model was built on reliable and meaningful data.

# Data Cleaning & Preprocessing

Financial datasets, especially those spanning many years and industries, often have missing values or inconsistencies. Ensuring the dataset's integrity was an essential part of this project. Several steps were taken to filter and clean the dataset, as described below:

## Removing Invalid or Missing Values:

The dataset was filtered to exclude any rows with missing values in critical columns such as total assets, long-term debt, revenue, or net income. Rows with missing data in these fields would not contribute to meaningful insights and could distort the predictions and clustering.

### Imposing Conditions for Inclusion:

Specific conditions were applied to ensure the companies included had meaningful and relevant financial data:

* Only companies with more than 500,000 shares outstanding (cshoq > 0.5) were considered. This threshold was set to focus on companies with significant market activity, avoiding penny stocks or companies with extremely small market capitalizations.
* Companies were also required to have a positive closing stock price (prccq > 0). This filter excluded companies that were either delisted or had filed for bankruptcy, ensuring the dataset focused on active businesses.

### Market Value Calculation:

One of the key transformations was recalculating the market value. The dataset’s original market value column was found to be inaccurate and failed to appropriately capture the current market capitalization of the companies. Instead, I calculated man-made market value (man\_mkt\_val) as the product of shares outstanding (cshoq) and price close (prccq), providing a better representation of a company's true market value. This metric became the target variable for both the Gradient Descent and K-means clustering models.

## Final Dataset Statistics:

After applying these cleaning steps, the dataset was reduced from 1,134,506 rows to 695,749 rows. The significant reduction was necessary to ensure the quality of the data, as incomplete records or companies with extremely small market activity could have skewed the model results.

* Total rows before cleaning: 1,134,506
* Total rows after cleaning: 695,749
* Number of rows excluded: 438,757

This extensive cleaning process ensured that the dataset was robust and reliable for further analysis.

# Gradient Descent: Regression Analysis

## Objective:

The goal of the Gradient Descent model was to assess the relationship between several financial metrics and a company's market value. Specifically, I aimed to determine how features like assets, liabilities, debt, and revenue impact a company's market value and which factors are most significant in predicting market capitalization.

Methodology:

Gradient Descent is an optimization algorithm used to minimize a loss function—in this case, the Mean Squared Error (MSE) between the predicted and actual market values. The model iteratively adjusted the weights of the financial features to minimize the prediction error.

The following financial metrics were used as features:

* Total Assets (atq)
* Long-term Debt (dlttq)
* Liabilities (ltq)
* Net Income (niq)
* Revenue (revtq)
* Price Close (prccq)
* Price High (prchq)
* Price Low (prclq)
* The man-made market value (man\_mkt\_val) was the target variable.

After assembling the feature vectors and normalizing the data using StandardScaler, the Gradient Descent algorithm was applied with the following settings:

* Initial Learning Rate: 0.5
* Bold Driver: This technique was employed to dynamically adjust the learning rate. If the loss decreased, the learning rate was increased by 5%. If the loss increased, the learning rate was reduced by 30%.
* Number of Iterations: 50
* Tolerance: A tolerance level of 1e-6 was set to prevent over-adjusting for minor changes in the loss function.

## Results:

After 50 iterations, the model produced the following final weights and intercept:

Final Weights:

Assets (atq): 7,785.36

Long-term Debt (dlttq): -1,563.29

Liabilities (ltq): -2,489.41

Net Income (niq): 10,819.21

Revenue (revtq): 8,902.73

Price Close (prccq): 3,947.22

Price High (prchq): 3,214.87

Price Low (prclq): 2,876.39

Final Intercept: 8,784.79

Final Cost (Loss): 482,895,005.62

Final Learning Rate: 0.887

## Interpretation:

The model revealed several key insights into the relationship between financial metrics and market value:

* Total Assets and Net Income had the highest positive impact on market value. This is intuitive, as companies with more assets and higher earnings are generally more valuable.
* Long-term Debt and Liabilities had a negative impact, indicating that companies with higher obligations tend to have lower market values.
* Revenue had a strong positive influence on market value, highlighting its importance in determining a company's overall worth.
* The results were consistent with financial theory, and the Gradient Descent model provided valuable insights into which features are the most significant drivers of market value.

I am not enthused about my final cost (loss) presenting so high, however it makes some sense when reviewing the quantity of variables, the variability of each company, and how little each metric truly captures the whole picture of a company’s outlook. Nevertheless, we were able to assign numerical values to intuitive results found in this model (ex: increase in revenue would increase market value, and here we’ve determined that to be an increase of 8,902.73).

# K-Means Clustering: Grouping Companies Based on Financial Metrics

## Objective:

While Gradient Descent allowed for precise predictions of market value, K-means clustering was used to explore patterns and group companies with similar financial characteristics. This unsupervised learning technique helps to uncover hidden structures in data, allowing us to categorize companies based on their financial metrics.

## Methodology:

K-means clustering was applied to the same set of financial metrics used in the Gradient Descent model. The data was standardized and fed into the K-means algorithm with the following settings:

* Number of Clusters (k): 3
* Maximum Iterations: 100
* Convergence Tolerance: 1e-6

The financial features used for clustering were:

* Total Assets (atq)
* Long-term Debt (dlttq)
* Liabilities (ltq)
* Net Income (niq)
* Revenue (revtq)
* Price Close (prccq)
* Price High (prchq)
* Price Low (prclq)

## Cluster Results:

The K-means algorithm successfully grouped the companies into three clusters:

* array([0.01433253, 0.00405741, 0.00425073, 0.04690792, 0.08647996, 0.02145579, 0.0214851 , 0.02047041])
  + Cluster 1: Represented smaller companies with lower assets, liabilities, revenue, and market value.
* array([10.48393291, 3.38990897, 9.57795132, 8.10865134, 10.179201 0.19319167, 0.19129109, 0.18923176])
  + Cluster 2: Included mid-sized companies with moderate financials.
* array([ 4.50108426e+01, 7.38600287e+01, 4.80462226e+01, 4.85666572e+00, 6.07615103e+00, -3.59091033e-02, -3.68708661e-02, -3.42315930e-02])
  + Cluster 3: Comprised large, financially stronger companies with substantial assets, revenue, and market value.

## Cluster Centers:

The centers of each cluster provide insight into the average characteristics of the companies in each group. For example:

* Companies in Cluster 3 had significantly higher assets and revenues compared to the other clusters, indicating their position as market leaders.
* Companies in Cluster 1 had relatively low financial metrics, suggesting they were smaller or potentially younger firms.

## Interpretation:

This clustering analysis revealed three distinct groups of companies, each with unique financial profiles. These clusters provide a valuable tool for investors or analysts looking to segment companies based on their financial characteristics and market position.

# Conclusion & Future Work

This project provided valuable insights into the financial health of companies and their market value. By combining Gradient Descent for regression analysis and K-means Clustering for pattern recognition, I was able to uncover significant relationships between financial metrics and market value and group companies into clusters based on their financial characteristics.

The Gradient Descent model revealed that assets, revenue, and net income are key drivers of market value, while debt and liabilities negatively impact a company's worth. Meanwhile, the K-means Clustering model identified three broad groups of companies, ranging from small firms with modest financials to large corporations with substantial assets and market influence.

## Future Work:

Several avenues for future research and improvements exist:

* Model Refinement: The Gradient Descent model could be further refined by exploring different feature sets or using more sophisticated regression techniques.
* Expanded Clustering: The K-means clustering could be expanded by including additional financial or non-financial variables (e.g., industry sector, geographical location) to capture a broader range of company characteristics.
* Predictive Analysis: The current analysis focused on historical data. Future projects could use these models to predict future market value or financial trends, applying time-series forecasting techniques.

Overall, this project demonstrates the power of machine learning techniques in financial analysis and provides me with a solid foundation for further exploration in this field (as well as a project for me to present in future interviews).

# Appendices

## Full Dataset Link

My full dataset can be found at gs://cs777fall2024-123456/term\_project/MET CS 777 full data.csv

## Spark Screenshots

### Gradient Descent

A screenshot of a computer

Description automatically generated

### K-Means

A screenshot of a computer

Description automatically generated

## Code Output

### Gradient Descent

Total rows before cleaning: 1134506

Total rows after cleaning: 695749

Number of rows excluded during cleaning: 438757

+-----+----------+------+----+------+------+------+-------+---+---------+--------+------+--------+--------+-------+------+-------+------+------+-------+------+-------+-------+------+-------+------+------+-------+------+-------+-------+------------------+

|gvkey| datadate|fyearq|fqtr|indfmt|consol|popsrc|datafmt|tic| cusip| conm|curcdq|datacqtr|datafqtr| atq| cheq| cogsq|cshopq| cshoq| dlttq|epsfxq| ltq| niq|oiadpq| revtq|costat|dvpspq|mkvaltq| prccq| prchq| prclq| man\_mkt\_val|

+-----+----------+------+----+------+------+------+-------+---+---------+--------+------+--------+--------+-------+------+-------+------+------+-------+------+-------+-------+------+-------+------+------+-------+------+-------+-------+------------------+

| 1004|2000-02-29| 1999| 3| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2000Q1| 1999Q3|753.755| 4.431|222.672| NULL|26.963|180.639| 0.4|411.273| 10.955| 20.67|272.331| A| 0.085| NULL| 23.75| 28.5| 15.25| 640.37125|

| 1004|2000-05-31| 1999| 4| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2000Q2| 1999Q4|740.998| 1.241| 182.44| NULL|26.865|180.447| 0.09|401.483| 2.471| 8.222|225.079| A| 0.085| NULL|13.875|24.1875| 13.625| 372.751875|

| 1004|2000-08-31| 2000| 1| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2000Q3| 2000Q1|747.543| 0.669|202.661| NULL|26.857|180.367| 0.12| 408.29| 3.159| 9.871| 241.77| A| 0.085| NULL| 11.25|15.3125|10.0625| 302.14125|

| 1004|2000-11-30| 2000| 2| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2000Q4| 2000Q2|772.941| 3.069|171.482| NULL|26.932|180.173| 0.16|431.677| 4.278|11.458|211.335| A| 0.085| NULL|10.375|13.6875| 9.75|279.41949999999997|

| 1004|2001-02-28| 2000| 3| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2001Q1| 2000Q3|754.718| 0.767|159.537| NULL|26.945|180.106| 0.2|409.853| 5.388|12.459|200.071| A| 0.085| NULL| 13.6| 15.19| 10.0| 366.452|

| 1004|2001-05-31| 2000| 4| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2001Q2| 2000Q4|701.854|13.809|180.131| NULL|26.937|179.987| 0.21|361.642| 5.706|12.002|221.079| A| 0.085| NULL| 14.0| 15.7| 10.25| 377.118|

| 1004|2001-08-31| 2001| 1| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2001Q3| 2001Q1|758.503| 42.81|168.829| NULL|26.958| 189.92| 0.02|418.356| 0.486| 5.445|202.993| A| 0.085| NULL| 17.06| 17.45| 14.0|459.90347999999994|

| 1004|2001-11-30| 2001| 2| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2001Q4| 2001Q2|714.208|53.129|118.697| NULL|26.876|189.733| -2.03|429.864|-54.484|-0.171|144.889| A| 0.085| NULL| 8.15| 17.4| 6.96|219.03940000000003|

| 1004|2002-02-28| 2001| 3| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2002Q1| 2001Q3|690.681|34.454|118.267| NULL|31.859|189.665| -0.08|375.849| -2.29| 0.043|143.457| A| 0.025| NULL| 7.4| 10.05| 7.15|235.75660000000002|

| 1004|2002-05-31| 2001| 4| INDL| C| D| STD|AIR|000361105|AAR CORP| USD| 2002Q2| 2001Q4|710.199|34.522|120.684| NULL| 31.87|217.699| -0.08|399.964| -2.651|-0.606|147.382| A| 0.025| NULL| 11.44| 14.0| 7.15| 364.5928|

+-----+----------+------+----+------+------+------+-------+---+---------+--------+------+--------+--------+-------+------+-------+------+------+-------+------+-------+-------+------+-------+------+------+-------+------+-------+-------+------------------+

only showing top 10 rows

Starting gradient descent with multiple variables...

Iteration 1: Initial iteration

Iteration 1: Loss = 885831314.9918345, Learning Rate = 0.5

Iteration 2: Loss improved, increasing learning rate to 0.525

Iteration 2: Loss = 562067981.0087132, Learning Rate = 0.525

Iteration 3: Loss improved, increasing learning rate to 0.55125

Iteration 3: Loss = 503057644.9349671, Learning Rate = 0.55125

Iteration 4: Loss improved, increasing learning rate to 0.5788125000000001

Iteration 4: Loss = 491859402.1982042, Learning Rate = 0.5788125000000001

Iteration 5: Loss improved, increasing learning rate to 0.6077531250000001

Iteration 5: Loss = 489391731.1022242, Learning Rate = 0.6077531250000001

Iteration 6: Loss improved, increasing learning rate to 0.6381407812500002

Iteration 6: Loss = 488634638.7355074, Learning Rate = 0.6381407812500002

Iteration 7: Loss improved, increasing learning rate to 0.6700478203125002

Iteration 7: Loss = 488288745.8683941, Learning Rate = 0.6700478203125002

Iteration 8: Loss improved, increasing learning rate to 0.7035502113281252

Iteration 8: Loss = 488085716.3562258, Learning Rate = 0.7035502113281252

Iteration 9: Loss improved, increasing learning rate to 0.7387277218945315

Iteration 9: Loss = 487967028.40586317, Learning Rate = 0.7387277218945315

Iteration 10: Loss improved, increasing learning rate to 0.7756641079892581

Iteration 10: Loss = 487944970.21656555, Learning Rate = 0.7756641079892581

Iteration 11: Loss worsened, decreasing learning rate to 0.5429648755924806

Iteration 11: Loss = 488105147.07378125, Learning Rate = 0.5429648755924806

Iteration 12: Loss = 488105147.07378125, Learning Rate = 0.5429648755924806

Iteration 13: Loss improved, increasing learning rate to 0.5701131193721047

Iteration 13: Loss = 487486207.2621644, Learning Rate = 0.5701131193721047

Iteration 14: Loss improved, increasing learning rate to 0.59861877534071

Iteration 14: Loss = 487154357.1937524, Learning Rate = 0.59861877534071

Iteration 15: Loss improved, increasing learning rate to 0.6285497141077455

Iteration 15: Loss = 486959370.59305924, Learning Rate = 0.6285497141077455

Iteration 16: Loss improved, increasing learning rate to 0.6599771998131329

Iteration 16: Loss = 486804872.7084561, Learning Rate = 0.6599771998131329

Iteration 17: Loss improved, increasing learning rate to 0.6929760598037895

Iteration 17: Loss = 486663550.19835955, Learning Rate = 0.6929760598037895

Iteration 18: Loss improved, increasing learning rate to 0.727624862793979

Iteration 18: Loss = 486527533.65427095, Learning Rate = 0.727624862793979

Iteration 19: Loss improved, increasing learning rate to 0.7640061059336779

Iteration 19: Loss = 486397026.4540019, Learning Rate = 0.7640061059336779

Iteration 20: Loss improved, increasing learning rate to 0.8022064112303618

Iteration 20: Loss = 486280060.76605636, Learning Rate = 0.8022064112303618

Iteration 21: Loss improved, increasing learning rate to 0.8423167317918799

Iteration 21: Loss = 486201440.8963633, Learning Rate = 0.8423167317918799

Iteration 22: Loss worsened, decreasing learning rate to 0.5896217122543159

Iteration 22: Loss = 486234941.2147898, Learning Rate = 0.5896217122543159

Iteration 23: Loss = 486234941.2147898, Learning Rate = 0.5896217122543159

Iteration 24: Loss improved, increasing learning rate to 0.6191027978670317

Iteration 24: Loss = 485980654.48405534, Learning Rate = 0.6191027978670317

Iteration 25: Loss improved, increasing learning rate to 0.6500579377603833

Iteration 25: Loss = 485775083.0651883, Learning Rate = 0.6500579377603833

Iteration 26: Loss improved, increasing learning rate to 0.6825608346484026

Iteration 26: Loss = 485618643.0681007, Learning Rate = 0.6825608346484026

Iteration 27: Loss improved, increasing learning rate to 0.7166888763808227

Iteration 27: Loss = 485484966.74158084, Learning Rate = 0.7166888763808227

Iteration 28: Loss improved, increasing learning rate to 0.7525233201998639

Iteration 28: Loss = 485370139.38152605, Learning Rate = 0.7525233201998639

Iteration 29: Loss improved, increasing learning rate to 0.7901494862098571

Iteration 29: Loss = 485286433.7879336, Learning Rate = 0.7901494862098571

Iteration 30: Loss improved, increasing learning rate to 0.82965696052035

Iteration 30: Loss = 485275009.0410004, Learning Rate = 0.82965696052035

Iteration 31: Loss worsened, decreasing learning rate to 0.580759872364245

Iteration 31: Loss = 485456220.22743636, Learning Rate = 0.580759872364245

Iteration 32: Loss = 485456220.22743636, Learning Rate = 0.580759872364245

Iteration 33: Loss improved, increasing learning rate to 0.6097978659824572

Iteration 33: Loss = 485073379.7912006, Learning Rate = 0.6097978659824572

Iteration 34: Loss improved, increasing learning rate to 0.6402877592815801

Iteration 34: Loss = 484798928.6326551, Learning Rate = 0.6402877592815801

Iteration 35: Loss improved, increasing learning rate to 0.6723021472456592

Iteration 35: Loss = 484619153.63233757, Learning Rate = 0.6723021472456592

Iteration 36: Loss improved, increasing learning rate to 0.7059172546079422

Iteration 36: Loss = 484480299.45201385, Learning Rate = 0.7059172546079422

Iteration 37: Loss improved, increasing learning rate to 0.7412131173383393

Iteration 37: Loss = 484369577.47779316, Learning Rate = 0.7412131173383393

Iteration 38: Loss improved, increasing learning rate to 0.7782737732052563

Iteration 38: Loss = 484296619.6612768, Learning Rate = 0.7782737732052563

Iteration 39: Loss worsened, decreasing learning rate to 0.5447916412436793

Iteration 39: Loss = 484302424.42096716, Learning Rate = 0.5447916412436793

Iteration 40: Loss = 484302424.42096716, Learning Rate = 0.5447916412436793

Iteration 41: Loss improved, increasing learning rate to 0.5720312233058633

Iteration 41: Loss = 483980907.3401945, Learning Rate = 0.5720312233058633

Iteration 42: Loss improved, increasing learning rate to 0.6006327844711565

Iteration 42: Loss = 483781872.6193968, Learning Rate = 0.6006327844711565

Iteration 43: Loss improved, increasing learning rate to 0.6306644236947144

Iteration 43: Loss = 483640987.5043662, Learning Rate = 0.6306644236947144

Iteration 44: Loss improved, increasing learning rate to 0.6621976448794502

Iteration 44: Loss = 483515934.06979746, Learning Rate = 0.6621976448794502

Iteration 45: Loss improved, increasing learning rate to 0.6953075271234227

Iteration 45: Loss = 483394496.3583679, Learning Rate = 0.6953075271234227

Iteration 46: Loss improved, increasing learning rate to 0.7300729034795939

Iteration 46: Loss = 483273145.5175716, Learning Rate = 0.7300729034795939

Iteration 47: Loss improved, increasing learning rate to 0.7665765486535736

Iteration 47: Loss = 483152143.9293683, Learning Rate = 0.7665765486535736

Iteration 48: Loss improved, increasing learning rate to 0.8049053760862523

Iteration 48: Loss = 483035723.8684685, Learning Rate = 0.8049053760862523

Iteration 49: Loss improved, increasing learning rate to 0.845150644890565

Iteration 49: Loss = 482936949.2308566, Learning Rate = 0.845150644890565

Iteration 50: Loss improved, increasing learning rate to 0.8874081771350933

Iteration 50: Loss = 482895005.2180027, Learning Rate = 0.8874081771350933

Final weights: [ 7785.98523385 -1743.08385924 -6740.89312236 10619.74837608

10819.30205358 326.95921582 425.65131358 -565.94943657]

Final intercept: 4360.248781822676

Final cost: 482895005.2180027

Final learning rate: 0.8874081771350933

The dataset's average market value is (in millions): 4360.248781822716

Intercept (b) (Man-Made Market Value): 4360.248781822676

Slope m1 (Assets - Total (atq)): 7785.985233846213

Slope m2 (Long-Term Debt - Total (dlttq)): -1743.083859242927

Slope m3 (Liabilities - Total (ltq)): -6740.893122356269

Slope m4 (Net Income (niq)): 10619.748376081378

Slope m5 (Revenue - Total (revtq)): 10819.302053576099

Slope m6 (Price Close - Quarter (prccq)): 326.95921581511465

Slope m7 (Price High - Quarter (prchq)): 425.6513135763884

Slope m8 (Price Low - Quarter (prclq)): -565.9494365689827

Final Cost: 482895005.2180027

Final Learning Rate: 0.8874081771350933

### K-Means

Running K-means clustering...

Cluster Centers: [array([0.01433253, 0.00405741, 0.00425073, 0.04690792, 0.08647996,

0.02145579, 0.0214851 , 0.02047041]), array([10.48393291, 3.38990897, 9.57795132, 8.10865134, 10.179201 ,

0.19319167, 0.19129109, 0.18923176]), array([ 4.50108426e+01, 7.38600287e+01, 4.80462226e+01, 4.85666572e+00,

6.07615103e+00, -3.59091033e-02, -3.68708661e-02, -3.42315930e-02])]

+------+--------+--------+--------+-------+--------+------+------+------+------------------+----------+

| tic| atq| dlttq| ltq| niq| revtq| prccq| prchq| prclq| man\_mkt\_val|prediction|

+------+--------+--------+--------+-------+--------+------+------+------+------------------+----------+

| ROL| 572.517| 0.171| 344.084| 12.553| 248.076| 18.08| 19.12| 9.37|1808.7412799999997| 0|

|DRAM.1| 23.194| 0.0| 4.34| -1.616| 10.673| 2.69| 4.49| 1.39|23.857609999999998| 0|

| NANO| 189.35| 9.612| 47.071| 12.327| 53.935| 15.05| 15.74| 8.0|333.65850000000006| 0|

| ABT| 72548.0| 19429.0| 39545.0| 2162.0| 10701.0|109.49|115.14|103.13|193931.97269999998| 0|

| PCG| 76995.0| 0.0| 64092.0|-6869.0| 4088.0| 23.75| 49.42| 17.26| 12358.05125| 0|

| STS| 136.15| 7.167| 31.644| 4.997| 74.79| 19.3| 19.83| 12.51| 329.8177| 0|

| BYI| 970.467| 494.375| 772.992| 26.521| 245.795| 46.66| 49.32| 43.52|1964.4793199999997| 0|

| SIX|2393.355|1497.005| 2030.82|-46.935| 115.419| 55.49| 56.33| 45.24| 5174.4425| 0|

| DXC| 10116.0| 1707.0| 7102.0| 163.0| 1804.0| 65.64| 71.0| 63.85| 9061.79892| 0|

| TNLX| 18.505| 0.394| 17.004| -1.01| 5.82| 4.56| 8.55| 4.55|4.7241599999999995| 0|

| OMI|3334.156| 917.256|2330.372| 10.87|2333.961| 29.2| 32.51| 26.91| 1788.4708| 0|

| AEE| 25945.0| 7094.0| 18619.0| -60.0| 1399.0| 58.99| 64.89| 57.67| 14310.974| 0|

| LH| 20385.4| 6143.6| 10091.4| 553.0| 4056.1|314.21|317.17|269.27|29252.950999999997| 0|

| KBAL| 313.747| 0.184| 137.543| 10.56| 171.983| 16.69| 18.94| 15.84| 622.5036200000001| 0|

| GHM| 143.131| 0.157| 33.751| 0.52| 22.301| 19.91| 20.24| 14.39| 192.05186| 0|

| DIS|195110.0| 39510.0| 91347.0| -20.0| 22083.0|122.36|123.74|88.685| 223429.36| 0|

| MR.UN| 668.197| 314.539| 362.14| 0.153| 16.446| 8.67| 9.0| 8.35| 96.67917| 0|

| PIRRQ| 779.847| 9.5| 319.862| 26.231| 367.615| 18.48| 18.9| 14.78| 1958.65824| 0|

| RAVE| 8.13| 0.0| 4.04| 0.853| 12.695|2.9955| 3.0| 2.17|29.532634499999997| 0|

| FELE|1545.981| 114.614| 620.403| 46.172| 459.019| 79.85| 86.42| 78.22| 3706.23775| 0|

+------+--------+--------+--------+-------+--------+------+------+------+------------------+----------+

only showing top 20 rows