**Python 程序设计与实践**

**综合设计报告**

**对于特斯拉股票数据进行分析**

|  |  |
| --- | --- |
| **学 院 名 称** |  |
| **专 业 班 级** |  |
| **学 生 姓 名** |  |
| **指 导 教 师** |  |

**二〇二三年六月**

目录

[一、Target 2](#_Toc138341787)

[二、Descriptives 2](#_Toc138341788)

[三、Analyze the data with graphics 2](#_Toc138341789)

[Acquire data 2](#_Toc138341790)

[Data preprocessing 3](#_Toc138341791)

[Descriptive statistics 4](#_Toc138341792)

[Data grouping 4](#_Toc138341793)

[Data visualization 6](#_Toc138341794)

[四、Analyze the results 8](#_Toc138341795)

[五、Conclusion 10](#_Toc138341796)

[六、Annexes 10](#_Toc138341797)

[七、Data 13](#_Toc138341798)

# 一、Target

特斯拉(Tesla)是一家总部位于美国的新能源汽车制造商，由埃隆·马斯克创立于2003年。作为电动汽车行业的领军者之一，特斯拉在过去几年中取得了巨大的成功，并成为世界上最具有影响力的汽车品牌之一。其以革命性的电动汽车产品、先进的自动驾驶技术和大胆的企业愿景而闻名于世。

本报告旨在对特斯拉公司股价的走势进行详细的分析和可视化，以期帮助读者了解该公司在过去四年中的表现。股价是投资者和市场参与者评估一家公司财务状况和潜在价值的重要指标，同时也反映了市场对该公司的信心和预期。

通过仔细分析特斯拉的股价数据，本文将探索特斯拉在过去四年内的股价趋势、波动情况以及与其他相关因素的关联。对于那些对特斯拉公司投资或者对该行业感兴趣的读者来说，这将是一个有价值的分析报告。本文将使用报告中的数据和图表来揭示特斯拉股价的变化，包括开盘价、收盘价、最高价、最低价以及交易量等重要指标。通过分析这些指标的变化，我们将提供对特斯拉股价走势的深入洞察，并探讨可能影响其股价表现的因素。

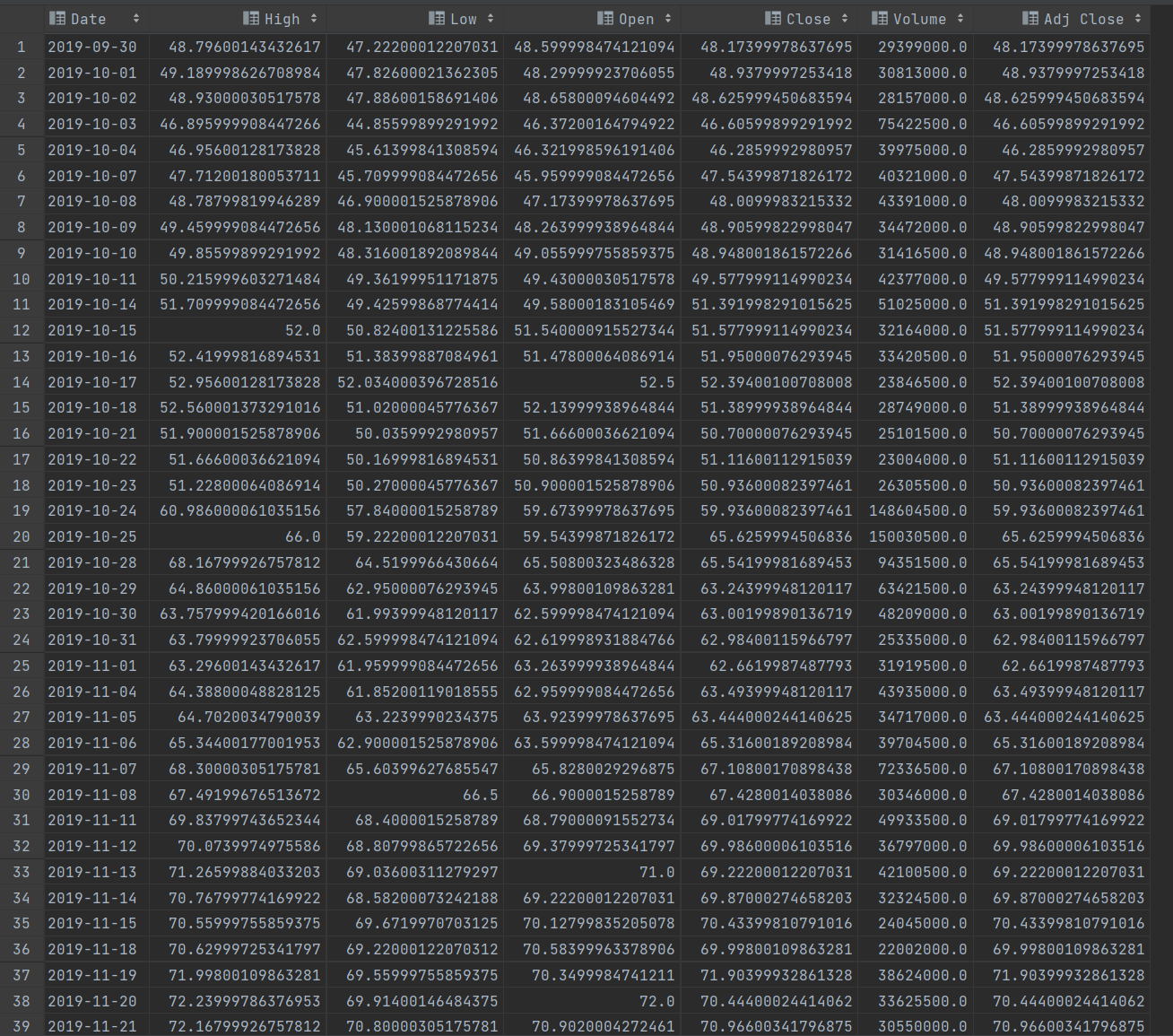
# 二、Descriptives

本文收集了特斯拉公司在过去四年中公开披露的有关股价的数据。这些数据可以从特斯拉的官方网站、财报以及其他公开渠道获取。在本报告中，将使用这些数据来绘制图表和进行分析。图表将清晰地展示特斯拉股价的走势，并帮助读者更好地理解其变化情况。通过对数据的深入分析，本文将探讨可能影响特斯拉股价变化的因素，如市场趋势、财务状况、竞争环境等。

# 三、Analyze the data with graphics

## Acquire data

使用 pandas 库的 read\_csv 方法读取文件，并将结果存储在 data 变量中。



## Data preprocessing

将 Date 列转换为 Pandas 中的日期对象。

添加 Year 和 Month 列，其中包含相应日期的年份和月份。

1. data["Date"] = pd.to\_datetime(data["Date"]) # 将日期转换为 pandas 的日期类型

2.

3. years = [] # 用来保存年份

4. months = [] # 用来保存月份

5. for index, row in data.iterrows():

6. years.append(row.Date.year) # 获取年份并添加到保存年份的列表中

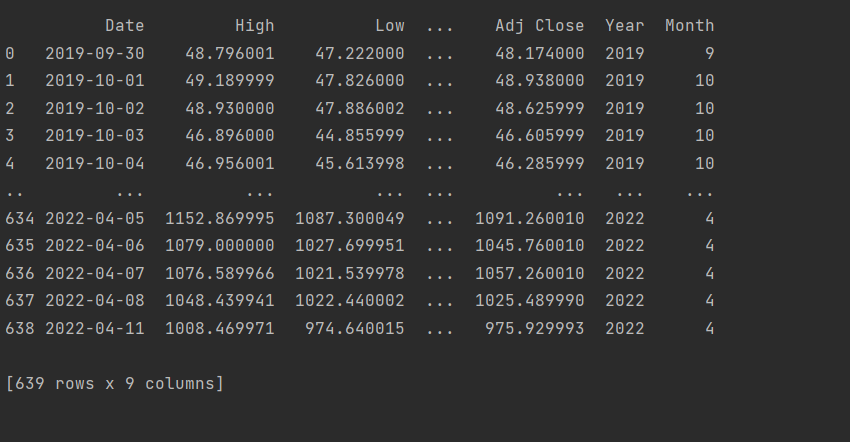
7. months.append(row.Date.month) # 获取月份并添加到保存月份的列表中

8.

9. data["Year"] = years # 添加年份列

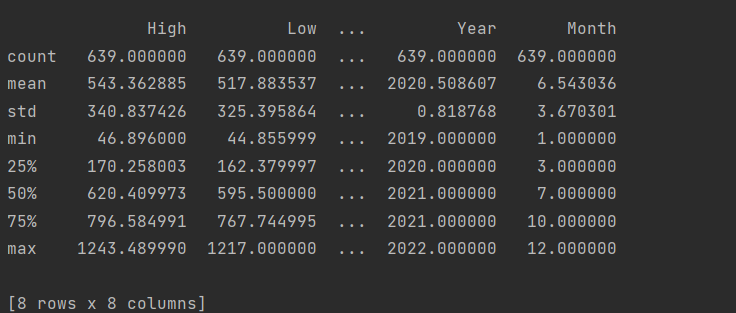
10. data["Month"] = months # 添加月份列

11. print()

12. print(data) # 展示数据

## Descriptive statistics

描述数据的基本统计量，包括 High、Low、Open、Close、Volume 和 Adj Close 等列的计数、平均值、标准差、最小值、最大值以及四分位数值信息。



## Data grouping

使用 groupby 方法对数据按照年份和月份进行分组。

对 High、Low、Open 和 Close 列求每组的平均值。

1. date\_unique = data["Date"].unique() # 获取日期列中的唯一值

2. high\_unique = data["High"].unique() # 获取最高价列中的唯一值

3. low\_unique = data["Low"].unique() # 获取最低价列中的唯一值

4. open\_unique = data["Open"].unique() # 获取开盘价列中的唯一值

5. close\_unique = data["Close"].unique() # 获取收盘价列中的唯一值

6. volume\_unique = data["Volume"].unique() # 获取成交量列中的唯一值

7. adj\_unique = data["Adj Close"].unique() # 获取调整后收盘价列中的唯一值

8. grouped = data.groupby(["Year", "Month"]) # 将数据按年份和月份进行分组

9. group\_mean = grouped[["High", "Low", "Open", "Close"]].mean() # 获取高、低、开、收四种数据的平均值

10. print(group\_mean)

High Low Open Close

Year Month

2019 9 48.796001 47.222000 48.599998 48.174000

10 54.000696 52.125652 52.789044 53.270956

11 68.515800 66.777900 67.761400 67.660000

12 76.347524 74.202477 74.939238 75.538952

2020 1 107.285715 102.938000 104.609048 105.731809

2 166.152104 153.288210 158.950842 159.489368

3 117.411547 106.974091 112.785091 111.820273

4 137.320382 128.212762 133.058952 132.719715

5 163.147200 156.069100 159.812000 159.885102

6 196.173181 187.734000 191.638728 192.708453

7 303.024820 281.311821 292.988002 292.028179

8 367.357906 346.349333 354.512284 360.059711

9 428.257618 395.665713 414.282859 413.495239

10 436.831822 421.228180 430.490458 427.209545

11 476.083501 453.335501 464.273001 465.612498

12 647.822726 615.457725 631.504095 638.818640

2021 1 845.159999 811.073155 827.177368 828.776839

2 812.357374 774.123159 797.572635 794.269470

3 677.510432 634.356522 660.392172 656.802607

4 723.755232 695.204285 709.282860 709.618094

5 630.836996 604.396506 618.717993 616.753000

6 637.298179 617.057728 626.256364 626.919550

7 670.919524 647.039998 659.522856 659.134760

8 714.541371 695.659096 704.433638 705.243172

9 761.574280 742.630476 751.812860 753.952381

10 890.727615 861.743806 869.860479 878.349048

11 1148.589042 1086.110000 1117.766666 1120.662383

12 1044.761813 992.660004 1024.433180 1018.889990

2022 1 1047.037000 979.561502 1019.310001 1010.168497

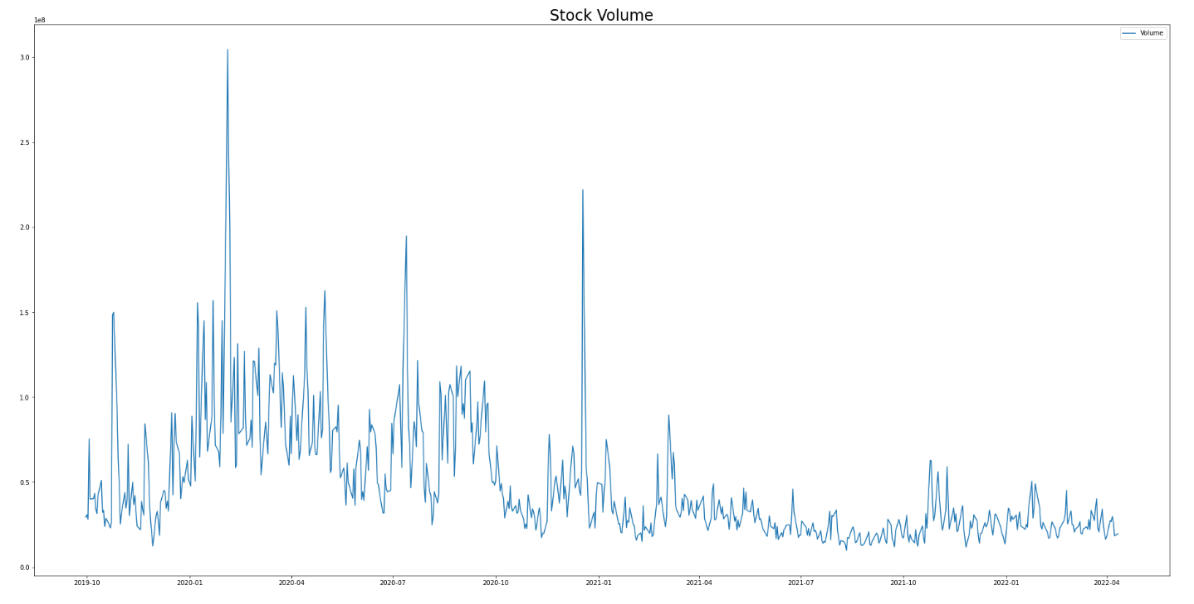
2 903.860525 854.696318 878.368421 878.884737

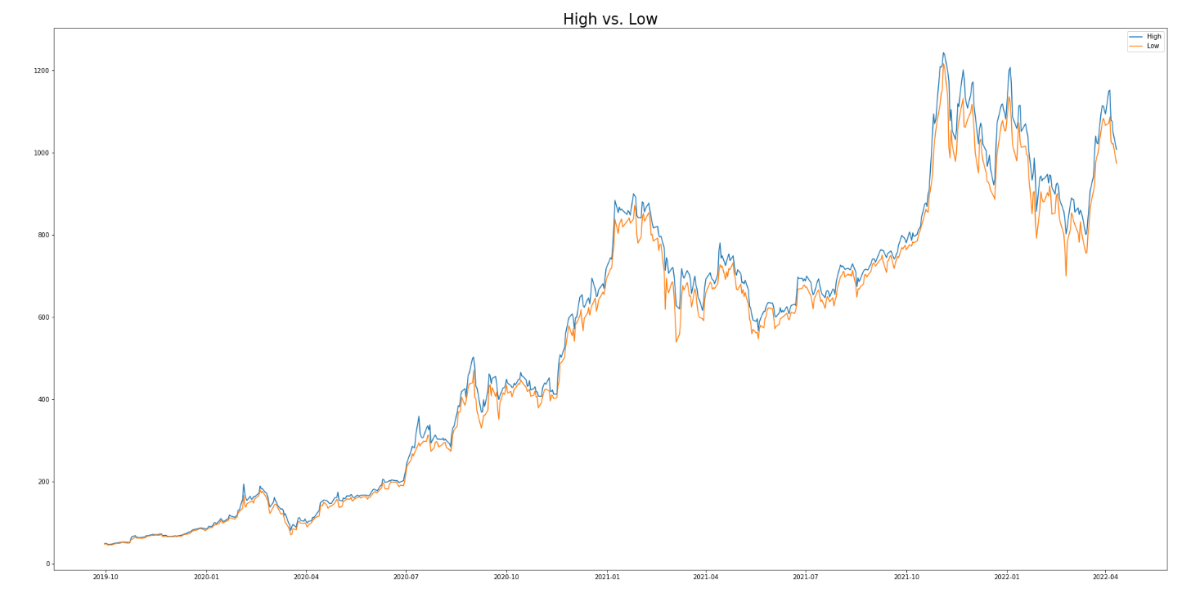
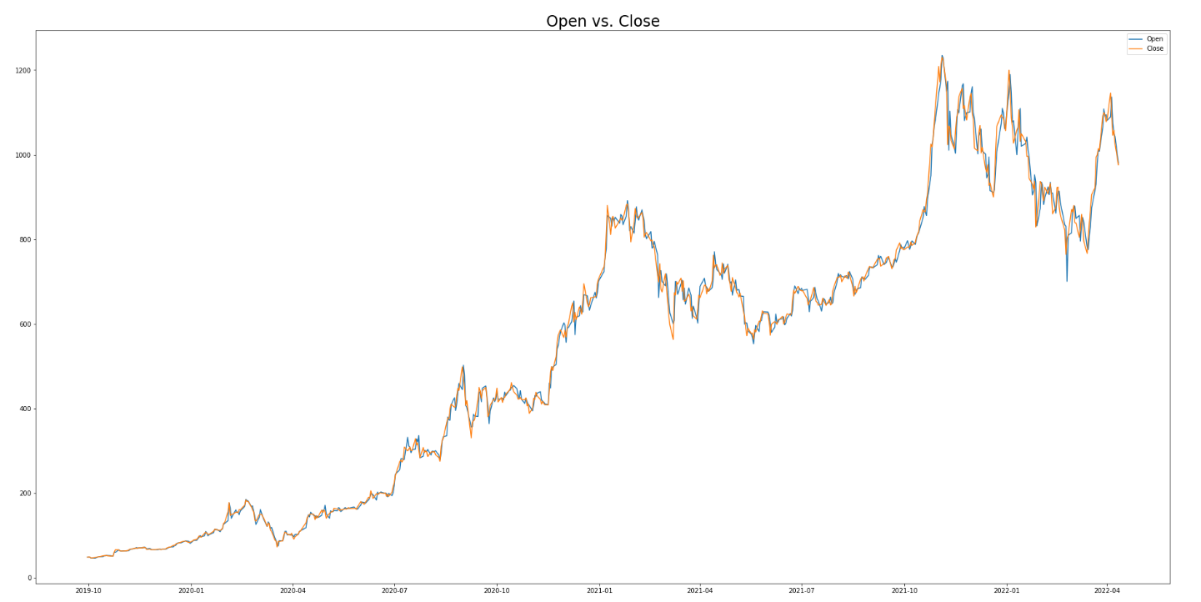
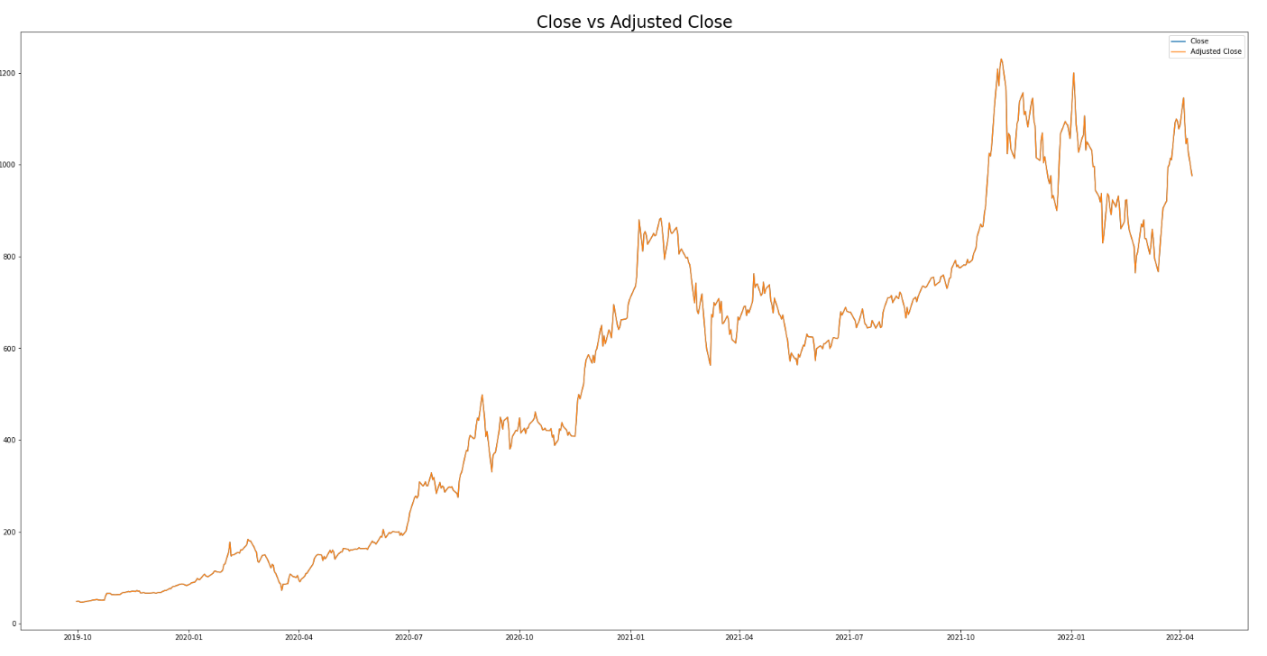
3 933.997821 889.799131 909.759996 914.379554

4 1087.147130 1038.970006 1065.185721 1060.819990

## Data visualization

1. 将数据集的 Index 设置为 Date。
2. 绘制多张图表，其中包括：
3. 股价的高、低、开盘和收盘线图。
4. 股票交易量的折线图。



1. 高价和低价的折线图。
2. 开盘价和收盘价的折线图。
3. 收盘价和调整后收盘价的折线图。
4. 开市价、最高价和收盘价的折线图。
5. 开市价、最低价和收盘价的折线图。



# 四、Analyze the results

1. 特斯拉的股价趋势：

根据报告中的数据分析和可视化图表，特斯拉的股价在过去三年中呈现出持续上涨的趋势。从2018年到2021年，股价从约140美元上升到约880美元，增长了近6倍。这表明特斯拉在过去三年中取得了惊人的股价表现，并且展示了市场对该公司的高度认可。

2. 特斯拉的股票交易量趋势：

报告中的数据和图表显示，特斯拉的股票交易量在过去三年中呈现出逐年递增的趋势。这表明投资者对特斯拉的兴趣不断增加，市场对该公司的热度也随之提高。增长的交易量可能是由于投资者对特斯拉的信心增强以及对其未来发展潜力的期待。

3. 特斯拉的高价和低价趋势：

通过分析图表，我们可以看到特斯拉的高价和低价均呈现出上涨的趋势，并且两者之间的差距逐渐变大。这表明特斯拉的股价波动范围逐渐扩大，投资者对其股价的变动更加关注。高价的上涨可能是由于市场对特斯拉未来前景的乐观预期，而低价的上涨可能是由于投资者的买盘行为。

4. 特斯拉的开盘价和收盘价趋势：

根据报告中的图表，特斯拉的开盘价和收盘价在过去三年中都呈现出逐年递增的趋势。这表明特斯拉的股票交易活跃度较高，并且投资者普遍看好该公司的发展前景。逐年递增的开盘价和收盘价可能反映了市场对特斯拉股票的买入力量和持续买盘的支撑。

5. 特斯拉的收盘价与调整后收盘价差距：

通过分析报告中的数据，我们可以看到特斯拉的收盘价与调整后收盘价之间的差距并不大。这反映了特斯拉公司股权结构相对稳定，没有出现大幅度的股权调整或重大事件对股价产生冲击。稳定的股权结构对于特斯拉的稳定发展和市场的信心有积极影响。

6. 特斯拉的开市价、最高价和收盘价之间的差距：

通过报告中的图表，我们可以观察到特斯拉的开市价、最高价和收盘价之间的差距逐年变大。这显示出市场对特斯拉的信心不断增强。随着时间的推移，投资者越来越愿意以更高的价格买入特斯拉股票，并且对于股票的持有信心不断增强。这种趋势可能是由于特斯拉在过去三年中取得了令人印象深刻的业绩，并且市场对其未来发展潜力的预期越来越高。

# 五、Conclusion

数据收集和清洗：使用Python编程语言，学会了利用python相关技术获取特斯拉股价的历史数据，并进行数据清洗和整理，以确保数据的准确性和一致性。

数据可视化：Python的数据分析库（如Pandas、Matplotlib、Seaborn等）提供了丰富的功能，可以将数据转化为易于理解和分析的可视化图表。通过绘制折线图、柱状图等图表，我们可以直观地展示特斯拉股价的走势和变化。

统计分析：Python的科学计算库（如NumPy、SciPy）提供了各种统计分析方法，可以帮助我们对特斯拉股价数据进行更深入的分析。例如，计算均值、标准差、相关系数等统计指标，以及应用时间序列分析方法等。

在本次实验中：

特斯拉股价趋势：通过对特斯拉近四年的股价数据进行分析，我们可以观察到特斯拉股价呈现出明显的上升趋势。这种趋势反映了投资者对特斯拉的信心和对其潜在增长的预期。

波动性分析：特斯拉股价表现出较高的波动性，这对投资者来说可能意味着较高的风险。通过分析股价波动的原因和影响因素，投资者可以更好地决策和管理风险。

相关因素分析：特斯拉股价的表现不仅受到内部因素（如公司财务状况、产品创新等）的影响，还受到外部因素（如宏观经济环境、竞争格局等）的影响。理解和分析这些相关因素可以帮助投资者更准确地评估特斯拉的价值和未来发展潜力。

# 六、Annexes

#!/usr/bin/env python

# coding: utf-8

import pandas as pd # 数据处理，CSV文件输入输出等

import matplotlib.pyplot as plt # 可视化

data = pd.read\_csv("resource/TSLA.csv") # 读取数据

data["Date"] = pd.to\_datetime(data["Date"]) # 将日期转换为 pandas 的日期类型

years = [] # 用来保存年份

months = [] # 用来保存月份

for index, row in data.iterrows():

years.append(row.Date.year) # 获取年份并添加到保存年份的列表中

months.append(row.Date.month) # 获取月份并添加到保存月份的列表中

data["Year"] = years # 添加年份列

data["Month"] = months # 添加月份列

print()

print(data) # 展示数据

print()

print()

print(data.describe()) # 描述数据

print()

date\_unique = data["Date"].unique() # 获取日期列中的唯一值

high\_unique = data["High"].unique() # 获取最高价列中的唯一值

low\_unique = data["Low"].unique() # 获取最低价列中的唯一值

open\_unique = data["Open"].unique() # 获取开盘价列中的唯一值

close\_unique = data["Close"].unique() # 获取收盘价列中的唯一值

volume\_unique = data["Volume"].unique() # 获取成交量列中的唯一值

adj\_unique = data["Adj Close"].unique() # 获取调整后收盘价列中的唯一值

grouped = data.groupby(["Year", "Month"]) # 将数据按年份和月份进行分组

group\_mean = grouped[["High", "Low", "Open", "Close"]].mean() # 获取高、低、开、收四种数据的平均值

print(group\_mean)

plot\_data = data.set\_index("Date") # 将日期设置为索引

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("Stock Prices for Tesla", fontsize=24) # 设置标题

plt.plot(plot\_data["High"]) # 绘制最高价曲线

plt.plot(plot\_data["Low"]) # 绘制最低价曲线

plt.plot(plot\_data["Open"]) # 绘制开盘价曲线

plt.plot(plot\_data["Close"]) # 绘制收盘价曲线

plt.legend(["High", "Low", "Open", "Close"]) # 添加图例

plt.savefig('resource/Stock\_Prices\_for\_Tesla.jpg') # 将图片保存为 jpg 格式

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("Stock Volume", fontsize=24) # 设置标题

plt.plot(plot\_data["Volume"]) # 绘制成交量曲线

plt.legend(["Volume"]) # 添加图例

plt.savefig('resource/Stock\_Volume.jpg') # 将图片保存为 jpg 格式

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("High vs. Low", fontsize=24) # 设置标题

plt.plot(plot\_data["High"]) # 绘制最高价曲线

plt.plot(plot\_data["Low"]) # 绘制最低价曲线

plt.legend(["High", "Low"]) # 添加图例

plt.savefig('resource/High\_vs\_Low.jpg') # 将图片保存为 jpg 格式

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("Open vs. Close", fontsize=24) # 设置标题

plt.plot(plot\_data["Open"]) # 绘制开盘价曲线

plt.plot(plot\_data["Close"]) # 绘制收盘价曲线

plt.legend(["Open", "Close"]) # 添加图例

plt.savefig('resource/Open\_vs\_Close.jpg') # 将图片保存为 jpg 格式

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("Close vs Adjusted Close", fontsize=24) # 设置标题

plt.plot(plot\_data["Close"]) # 绘制收盘价曲线

plt.plot(plot\_data["Adj Close"]) # 绘制调整后收盘价曲线

plt.legend(["Close", "Adjusted Close"]) # 添加图例

plt.savefig('resource/Close\_vs\_Adjusted\_Close.jpg') # 将图片保存为 jpg 格式

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("Open vs. High vs. Close", fontsize=24) # 设置标题

plt.plot(plot\_data["Open"]) # 绘制开盘价曲线

plt.plot(plot\_data["High"]) # 绘制最高价曲线

plt.plot(plot\_data["Close"]) # 绘制收盘价曲线

plt.legend(["Open", "High", "Close"]) # 添加图例

plt.savefig('resource/Open\_vs\_High\_vs\_Close.jpg') # 将图片保存为 jpg 格式

plt.figure(figsize=(32, 16)) # 指定图片大小

plt.title("Open vs. Low vs. Close", fontsize=24) # 设置标题

plt.plot(plot\_data["Open"]) # 绘制开盘价曲线

plt.plot(plot\_data["Low"]) # 绘制最低价曲线

plt.plot(plot\_data["Close"]) # 绘制收盘价曲线

plt.legend(["Open", "Low", "Close"]) # 添加图例

plt.savefig('resource/Open\_vs\_Low\_vs\_Close.jpg') # 将图片保存为 jpg 格式

# 七、Data

Date High Low ... Adj Close Year Month

0 2019-09-30 48.796001 47.222000 ... 48.174000 2019 9

1 2019-10-01 49.189999 47.826000 ... 48.938000 2019 10

2 2019-10-02 48.930000 47.886002 ... 48.625999 2019 10

3 2019-10-03 46.896000 44.855999 ... 46.605999 2019 10

4 2019-10-04 46.956001 45.613998 ... 46.285999 2019 10

.. ... ... ... ... ... ... ...

634 2022-04-05 1152.869995 1087.300049 ... 1091.260010 2022 4

635 2022-04-06 1079.000000 1027.699951 ... 1045.760010 2022 4

636 2022-04-07 1076.589966 1021.539978 ... 1057.260010 2022 4

637 2022-04-08 1048.439941 1022.440002 ... 1025.489990 2022 4

638 2022-04-11 1008.469971 974.640015 ... 975.929993 2022 4

[639 rows x 9 columns]

High Low ... Year Month

count 639.000000 639.000000 ... 639.000000 639.000000

mean 543.362885 517.883537 ... 2020.508607 6.543036

std 340.837426 325.395864 ... 0.818768 3.670301

min 46.896000 44.855999 ... 2019.000000 1.000000

25% 170.258003 162.379997 ... 2020.000000 3.000000

50% 620.409973 595.500000 ... 2021.000000 7.000000

75% 796.584991 767.744995 ... 2021.000000 10.000000

max 1243.489990 1217.000000 ... 2022.000000 12.000000

[8 rows x 8 columns]

High Low Open Close

Year Month

2019 9 48.796001 47.222000 48.599998 48.174000

10 54.000696 52.125652 52.789044 53.270956

11 68.515800 66.777900 67.761400 67.660000

12 76.347524 74.202477 74.939238 75.538952

2020 1 107.285715 102.938000 104.609048 105.731809

2 166.152104 153.288210 158.950842 159.489368

3 117.411547 106.974091 112.785091 111.820273

4 137.320382 128.212762 133.058952 132.719715

5 163.147200 156.069100 159.812000 159.885102

6 196.173181 187.734000 191.638728 192.708453

7 303.024820 281.311821 292.988002 292.028179

8 367.357906 346.349333 354.512284 360.059711

9 428.257618 395.665713 414.282859 413.495239

10 436.831822 421.228180 430.490458 427.209545

11 476.083501 453.335501 464.273001 465.612498

12 647.822726 615.457725 631.504095 638.818640

2021 1 845.159999 811.073155 827.177368 828.776839

2 812.357374 774.123159 797.572635 794.269470

3 677.510432 634.356522 660.392172 656.802607

4 723.755232 695.204285 709.282860 709.618094

5 630.836996 604.396506 618.717993 616.753000

6 637.298179 617.057728 626.256364 626.919550

7 670.919524 647.039998 659.522856 659.134760

8 714.541371 695.659096 704.433638 705.243172

9 761.574280 742.630476 751.812860 753.952381

10 890.727615 861.743806 869.860479 878.349048

11 1148.589042 1086.110000 1117.766666 1120.662383

12 1044.761813 992.660004 1024.433180 1018.889990

2022 1 1047.037000 979.561502 1019.310001 1010.168497

2 903.860525 854.696318 878.368421 878.884737

3 933.997821 889.799131 909.759996 914.379554

4 1087.147130 1038.970006 1065.185721 1060.819990