999786

June 12, 2023

1 Investigating periods of increasing interest rates for the S&P 1500

1.0.1 by Luca Reichelt, 999786

1.1 1. Outline

This project aims to investigate periods of increasing interest rates by the federal reserve based on the constituents of the S&P 1500. As an end result we attempt to implement our findings into a portfolio strategy.

The first month of observation is defined as the month of the first hike. The last date of observation is defined as the last month before a decreasing or stagnant interest rate (no change). Company data is based on the latest data available which have been published before the first day. We do not adjust for changes in the SP1500, but will include all the data available for its constituents as of the first month of the respective period.

All of the data that isn't retrieved while executing the code has been exported from Bloomberg Terminal or investing.com and is available within the folder "data_raw" in xlsx format. The data which is retrieved during the code execution is retrieved from Yahoo Finance or the FRED API, which is a API provided by the Federal Reserve of St.Lewis (https://fred.stlouisfed.org/docs/api/fred/).

You should be able to run this notebook after creating a local environment using anconda/miniconda and executing following command while referring to the provided yaml file:

`conda env create --file=env.yaml`

1.2 2. Data

[ ]:
```
# importing the main packages
from datetime import datetime
# import dateutil.parser
import os
import numpy as np
import pandas as pd

pd.options.mode.chained_assignment = None
import yfinance as yf
import pandas_datareader.data as web
import pickle as pkl
```

```
import dataframe_image as dfi
```

1.2.1 2.1 Data Retrieval

```
[ ]: # getting the last periods of increasing interest rates (Federal Funds␣
     ↪Effective MONTHLY Rate) from 1965 onwards
     fed_rates = web.DataReader("FEDFUNDS", "fred", 1965)
     fed_rates.set_index(pd.to_datetime(fed_rates.index.date), inplace=True)

     # test for empty values
     print(fed_rates.index.isna().sum())
```

0

```
[ ]: def period_df(start, duration, fed_rates=fed_rates):
         fed_rates = fed_rates[fed_rates.index >= start]

         df = pd.DataFrame(columns=["Name", "Start", "Last"])
         df.loc[df.shape[0]] = [None, None, None]

         df.is_copy = False

         period = 0
         j = 0

         for i in range(0, len(fed_rates) - 1):
             if (fed_rates.iloc[i + 1]["FEDFUNDS"] <= fed_rates.iloc[i]["FEDFUNDS"])␣
     ↪and (
                 i - j >= duration
             ):
                 df.loc[period, "Last"] = datetime.strftime(fed_rates.index[i],␣
     ↪"%Y-%m-%d")
                 period += 1
                 if (fed_rates.index[-1] - fed_rates.index[i]).days >= 365:
                     df.loc[len(df)] = [None, None, None]
                     j = i
             elif (fed_rates.iloc[i + 1]["FEDFUNDS"] <= fed_rates.
     ↪iloc[i]["FEDFUNDS"]) and (
                 i - j < duration
             ):
                 df.loc[period, "Name"] = "Period " + str(period + 1)
                 df.loc[period, "Start"] = datetime.strftime(fed_rates.index[i],␣
     ↪"%Y-%m-%d")
                 j = i

         # add last date
         df.loc[period, "Last"] = "2023-04-01"
```

```
    # add duration column
    df["Duration"] = (
        round(
            (pd.to_datetime(df["Last"]) - pd.to_datetime(df["Start"]))
            / np.timedelta64(1, "M")
        )
    ).astype(int)

    # export df as image
    dfi.export(
        df.style.set_properties(
            **{"background-color": "white", "color": "black", "border-color":␣
 ↪"#948b8b"}
        ),
        "periods" + start + ".png",
    )
    # df["Start"] = pd.to_datetime(df["Start"])
    # df["Last"] = pd.to_datetime(df["Last"])
    return df


period_df(start="1965", duration=12)
# unfortunately lack of data regarding constituents for this period from␣
 ↪bloomberg

periods = period_df(start="1995", duration=9)
```

[ ]:
```
def web_import(rate, start, end):
    df = web.DataReader(rate, "fred", start, end)
    df.index = pd.to_datetime(df.index)
    df.sort_index(inplace=True)
    return df
```

[ ]:
```
# def yf_import(ticker, start, end):
#     data = yf.download(ticker, start, end)
#     monthly = data.groupby(pd.PeriodIndex(data.index, freq="M"))["Close"].
 ↪mean()
#     print(monthly)
#     if data.size > 0:
#         if (
#             not data.empty == True
#             and len(monthly.values)
#             == round((data.index[-1] - data.index[0]) / np.timedelta64(1,␣
 ↪"M"))
#             and len(monthly.values) >= 9
#         ):
```

```
#             df = pd.DataFrame(
#                 index=pd.date_range(start, end, freq="MS"),
#                 data=np.append(monthly.values, data.iloc[-1]["Close"]),
#             )
#             df.rename(columns={df.columns[0]: ticker}, inplace=True)
#             df.drop(columns=df.columns.difference([ticker]), inplace=True)
#             df.index = pd.to_datetime(df.index)
#             return df
#         elif len(monthly.values) < 9:
#             empty = np.empty((1, 11 - (len(monthly))))
#             empty[:] = np.nan
#             df_data = np.append(empty, monthly.values)
#             df = pd.DataFrame(index=pd.date_range(start, end, freq="MS"),␣
↪data=df_data)
#             df.rename(columns={df.columns[0]: ticker}, inplace=True)
#             df.drop(columns=df.columns.difference([ticker]), inplace=True)
#             df.index = pd.to_datetime(df.index)
#             return df
#     else:
#         df = pd.DataFrame(index=pd.date_range(start, end, freq="MS"),␣
↪columns=[ticker])
#         df.index = pd.to_datetime(df.index)
#         return df

# above is overkill

def yf_import(ticker, start, end):
    data = yf.download(ticker, start, end)
    monthly = data.groupby(pd.PeriodIndex(data.index, freq="M"))["Close"].mean()
    monthly.index = monthly.index.to_timestamp()
    df = pd.DataFrame(index=pd.date_range(start, end, freq="MS"))
    df.index = pd.to_datetime(df.index)
    df.insert(0, ticker, monthly)
    return df
```

[ ]:
```
raw = "data_raw/"

gold = pd.read_csv(raw + "Gold_Futures.csv").set_index("Date")
gold["Gold"] = gold["Price"].str.replace(",", "").astype(float)
gold.drop(columns=gold.columns.difference(["Gold"]), inplace=True)
gold.index = pd.to_datetime(gold.index)
gold.sort_index(inplace=True)
```

1.2.2 2.2 Introductory Visualizations

[ ]:
```python
import plotly.express as px
import plotly.graph_objects as go
from plotly.subplots import make_subplots

tickers = ["^SP1500", "^IXIC"]
rates = ["FEDFUNDS", "CORESTICKM159SFRBATL", "UNRATE"]


def comparison(tickers, start, end):
    df = gold.loc[start:end]
    for ticker in tickers:
        df = pd.concat([df, yf_import(ticker, start, end)], axis=1)
    df = pd.concat([df, web.DataReader("WTISPLC", "fred", start, end)], axis=1)
    df_chg = df.pct_change() * 100

    legend = {
        "Gold": "Gold",
        "^SP1500": "S&P 1500",
        "^IXIC": "Nasdaq",
        "WTISPLC": "Spot Crude Oil Price WTI",
        "FEDFUNDS": "FED Rate",
        "CORESTICKM159SFRBATL": "CPI",
        "UNRATE": "Rate of Unemployment",
    }

    fig = px.line(
        df_chg,
        # ["Gold", "^SP1500", "^IXIC", "WTISPLC"],
        labels=legend,
        title="Indices/Assets monthly change in %",
    )
    fig.update_yaxes(title="Change in % compared to month before")
    fig.update_xaxes(title="Date")
    fig.for_each_trace(
        lambda t: t.update(
            name=legend[t.name],
            legendgroup=legend[t.name],
            hovertemplate=t.hovertemplate.replace(t.name, legend[t.name]),
        )
    )

    for index, row in periods.iterrows():
        fig.add_vline(
            x=row["Start"], line_width=2, line_dash="dash", line_color="green"
        )
```

```python
        fig.add_vline(x=row["Last"], line_width=2, line_dash="dash",␣
↪line_color="red")

    fig.update_layout(legend_title="Legend", autosize=False, width=1200,␣
↪height=600)
    fig.show()

    df_perf = df.apply(lambda x: x.div(x.iloc[0]).subtract(1).mul(100))

    for rate in rates:
        df_perf = pd.concat([df_perf, web.DataReader(rate, "fred", start,␣
↪end)], axis=1)

    fig = make_subplots(specs=[[{"secondary_y": True}]])

    x = df_perf.index

    for ticker in tickers:
        fig.add_trace(
            go.Scatter(x=x, y=df_perf[ticker], name=legend[ticker]),
            secondary_y=False,
        )

    fig.add_trace(
        go.Scatter(x=x, y=df_perf["Gold"], name="Gold"),
        secondary_y=False,
    )

    fig.add_trace(
        go.Scatter(x=x, y=df_perf["WTISPLC"], name="Spot Crude Oil Price WTI"),
        secondary_y=False,
    )

    for rate in rates:
        fig.add_trace(
            go.Scatter(x=x, y=df_perf[rate], name=legend[rate]),
            secondary_y=True,
        )

    for index, row in periods.iterrows():
        fig.add_vline(
            x=row["Start"], line_width=2, line_dash="dash", line_color="green"
        )
        fig.add_vline(x=row["Last"], line_width=2, line_dash="dash",␣
↪line_color="red")
```

```
    fig.update_yaxes(title_text="Total Indices/Asset change in %",␣
↪secondary_y=False)
    fig.update_yaxes(title_text="Rates for FED/CPI/Unemployment",␣
↪secondary_y=True)

    fig.update_xaxes(title="Date")

    fig.update_layout(
        title="Total performance to FED/CPI/Unemployment Rate",
        legend_title="Legend",
        autosize=False,
        width=1200,
        height=600,
    )
    fig.show()
```

[ ]:
```
comparison(
    tickers,
    periods.iloc[0]["Start"],
    periods.iloc[-1]["Last"],
)
```

```
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
```

1.2.3 2.3 Pre-Processing the features data

This data has been retrieved beforehand via the Bloomberg Excel Add-ins and is imported from the files in data-raw

[ ]:
```
# the periods refer to the four timeframes from the outline

period_data = [
    "SPR_Period_1.xlsx",
    "SPR_Period_2.xlsx",
    "SPR_Period_3.xlsx",
    "SPR_Period_4.xlsx",
]
```

[ ]:
```
def scatterplots(df):
    fig = px.scatter(
        df,
        x="Beta:M-1",
        y="Market Cap_perf",
        size="Market Cap",
        title="Risk/Compared Volatility to SP1500 compared to Return/
↪Performance for single companies",
        color="GICS Sector",
        hover_name="Name",
```

```
        log_x=True,
        size_max=100,
        width=1200,
        height=600,
    )

    fig.show()

    grouped = df.drop(columns=["Name"]).groupby("GICS Sector").mean()

    fig = px.scatter(
        grouped,
        x="Beta:M-1",
        y="Market Cap_perf",
        size="Market Cap",
        title="Risk/Compared Volatility to SP1500 compared to Return/
 ↪Performance by GICS Sector",
        color="Market Cap_perf",
        hover_name=grouped.index,
        log_x=True,
        size_max=100,
        width=1200,
        height=600,
    )

    fig.show()
```

[ ]:
```
import matplotlib.pyplot as plt
import seaborn as sns

def heatmap(df):
    df = df.corr(numeric_only=True)

    f, ax = plt.subplots(figsize=(18, 18))
    sns.heatmap(df, annot=True, linewidths=0.5, fmt=".1f", ax=ax)
    sns.set(font_scale=2)
    plt.xticks(rotation=45)
    plt.yticks(rotation=0)
    plt.show()
```

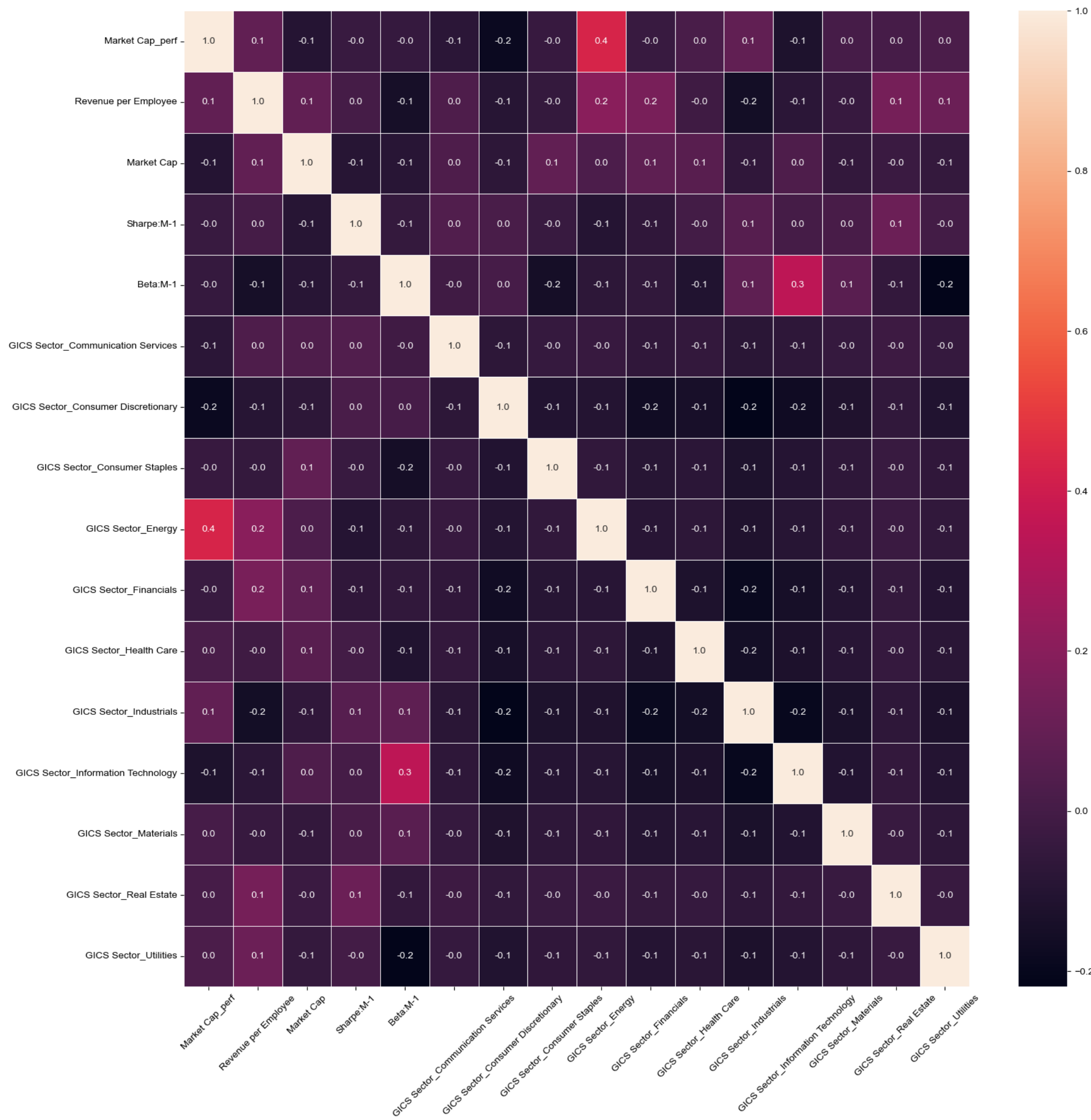
[ ]:
```
# this function cleans the datasets and provides first tabular and visual␣
 ↪descriptions of the data
# its also calls the heatmap visualization function above to give us an idea on␣
 ↪how to structure the model


def df(filename):
```

```
    df = pd.read_excel(open(raw + filename, "rb"))

    string_cols = ["Ticker", "Name", "GICS Sector"]

    for col in set(df.columns) - set(string_cols):
        df.loc[:, col] = pd.to_numeric(df.loc[:, col], errors="coerce")

    df.dropna(inplace=True)

    sector_dummies = pd.get_dummies(df[["GICS Sector"]])
    df = pd.concat([df, sector_dummies], axis=1)

    df["Revenue per Employee"] = (
        df["Revenue T12M"] / df["Number of Employees:Y"]
    ).astype(float)

    df["Market Cap_perf"] = (
        (df["Market Cap_last"] - df["Market Cap"]) / df["Market Cap"]
    ).astype(float)

    df.drop(["Market Cap_last", "Price"], axis=1, inplace=True)

    df.set_index("Ticker", inplace=True)
    df.sort_values("GICS Sector", inplace=True)

    print("\nData decription for cleaned " + filename)
    perf = df.pop("Market Cap_perf")
    df.insert(0, "Market Cap_perf", perf)

    rpe = df.pop("Revenue per Employee")
    df.insert(1, "Revenue per Employee", rpe)

    print(df["GICS Sector"].value_counts(ascending=False))
    print(df.iloc[:, :12].describe())

    scatterplots(df)

    heatmap(df)

    return df
```

[ ]:
```
P1 = df("SPR_Period_1.xlsx")
P1.to_pickle("data/P1.pkl")
P2 = df("SPR_Period_2.xlsx")
P2.to_pickle("data/P2.pkl")
P3 = df("SPR_Period_3.xlsx")
P3.to_pickle("data/P3.pkl")
```

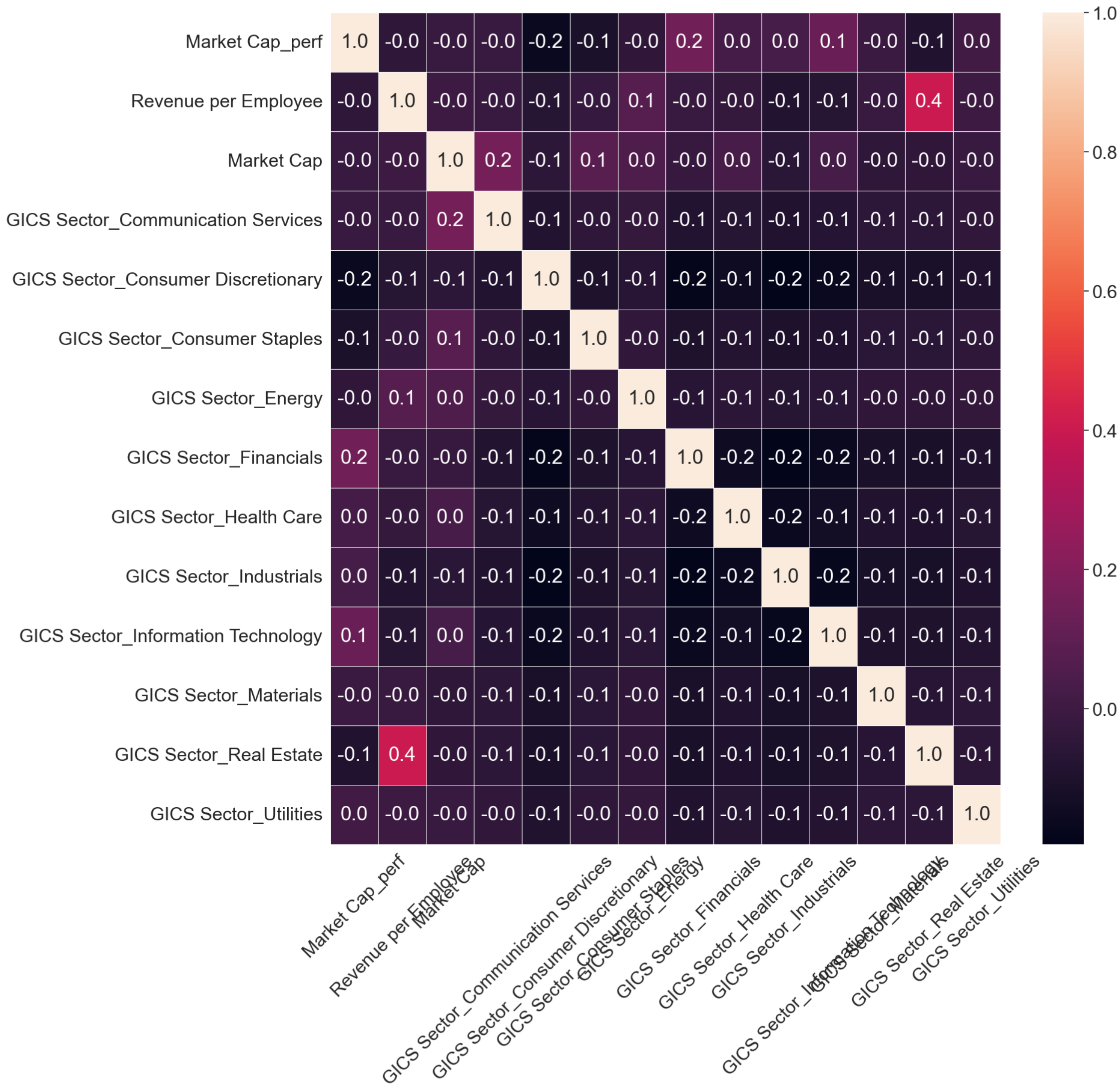
```python
CrP = df("SPR_Period_4.xlsx")
CrP.to_pickle("data/CrP.pkl")
```

```
Data decription for cleaned SPR_Period_1.xlsx
GICS Sector
Industrials                 145
Consumer Discretionary      120
Information Technology       98
Financials                   94
Health Care                  69
Materials                    51
Utilities                    47
Consumer Staples             44
Energy                       40
Communication Services       22
Real Estate                  14
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee     Market Cap  Sharpe:M-1  \
count       744.000000          7.440000e+02     744.000000  744.000000
mean          0.294032          4.468768e+05   12065.701248    0.715024
std           0.658582          6.624057e+05   32176.136903    3.805139
min          -0.916199          2.557610e+04     146.250700   -4.577547
25%          -0.113207          1.789160e+05    1152.156775   -1.530829
50%           0.179333          2.688665e+05    2761.541750   -0.361485
75%           0.503834          4.660680e+05    9059.699000    1.725719
max           5.033335          1.063660e+07  308608.843100   36.750693

         Beta:M-1
count  744.000000
mean     1.079571
std      0.353574
min      0.274396
25%      0.829210
50%      1.030910
75%      1.278337
max      2.533833
```



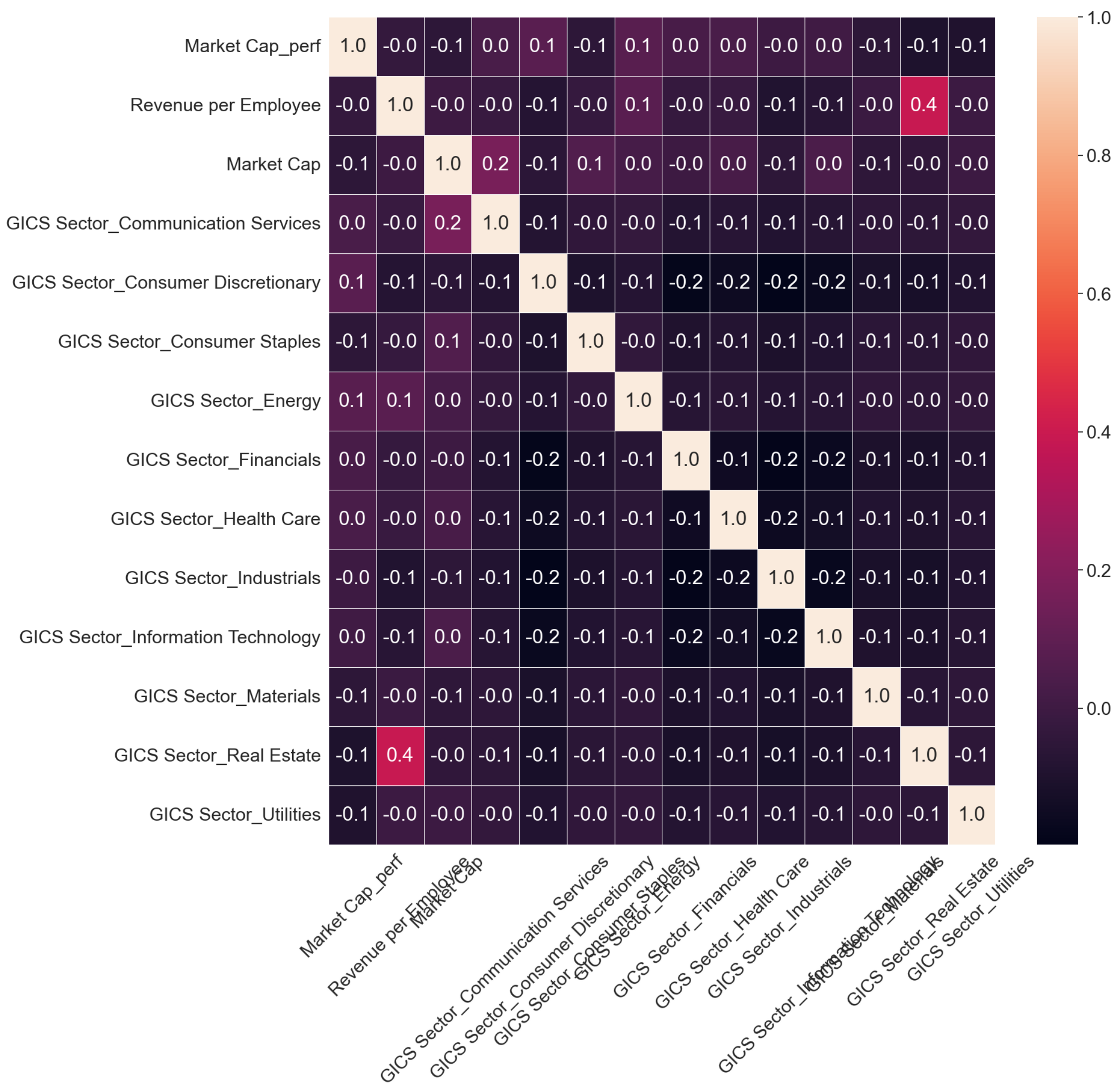

```
Data decription for cleaned SPR_Period_2.xlsx
GICS Sector
Industrials               216
Financials                198
Consumer Discretionary    196
Information Technology    162
Health Care               139
Real Estate                82
Materials                  81
Consumer Staples           65
Utilities                  52
Communication Services     47
```

```
Energy                        31
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee     Market Cap
count      1269.000000          1.269000e+03    1269.000000
mean          0.145752          9.019770e+05   16865.722069
std           0.310371          2.643064e+06   47038.230613
min          -0.839416          2.457104e+04     144.281000
25%          -0.031057          2.482764e+05    1402.683200
50%           0.138096          3.864179e+05    3528.513100
75%           0.286780          7.353425e+05   12265.221500
max           2.616234          5.176925e+07  575108.507600
```



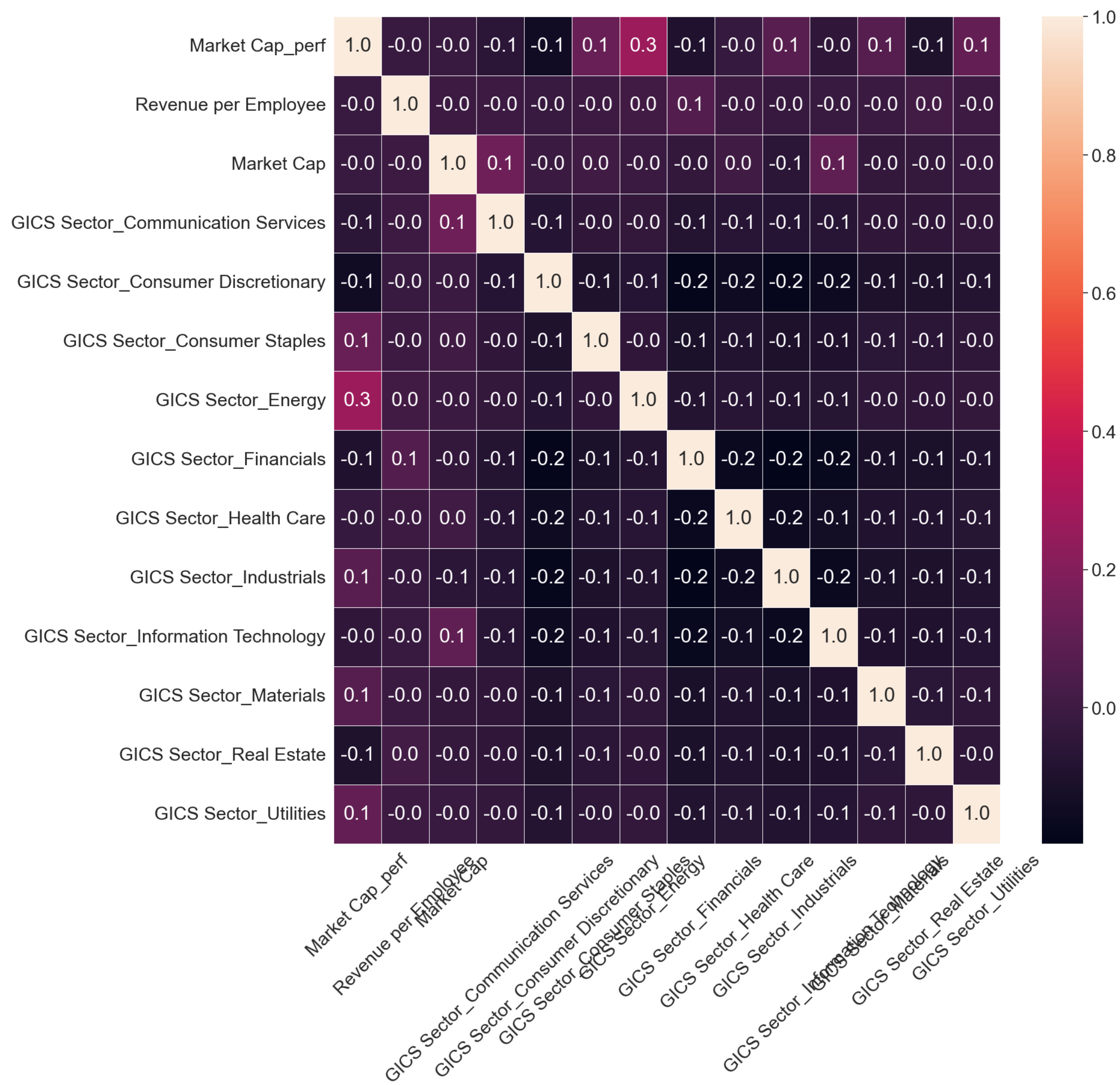
Data decription for cleaned SPR_Period_3.xlsx

```
GICS Sector
Industrials               221
Consumer Discretionary    207
Financials                196
Information Technology    164
Health Care               143
Real Estate                91
Materials                  77
Consumer Staples           65
Utilities                  48
Communication Services     46
Energy                     39
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee     Market Cap
count      1297.000000          1.297000e+03    1297.000000
mean          0.150803          9.279740e+05   18790.798789
std           0.314666          2.660655e+06   55517.775553
min          -0.882032          2.241154e+04     105.860000
25%          -0.042641          2.495037e+05    1526.499100
50%           0.102934          4.043673e+05    3917.466100
75%           0.284772          7.870889e+05   13333.739800
max           2.378570          4.657489e+07  847355.653400
```




```
Data decription for cleaned SPR_Period_4.xlsx
GICS Sector
Financials                  214
Industrials                 204
Consumer Discretionary      187
Information Technology      163
Health Care                 147
Materials                    77
Consumer Staples             73
Real Estate                  73
Utilities                    48
Energy                       43
Communication Services       41
```

```
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee    Market Cap
count      1270.000000          1.270000e+03  1.270000e+03
mean         -0.148952          1.681369e+06  3.716714e+04
std           0.285485          2.446662e+07  1.588635e+05
min          -0.925602          0.000000e+00  2.625325e+02
25%          -0.325974          2.873606e+05  2.475682e+03
50%          -0.156679          4.597162e+05  6.106569e+03
75%           0.023768          9.459195e+05  2.262599e+04
max           1.598834          8.679870e+08  2.986128e+06
```

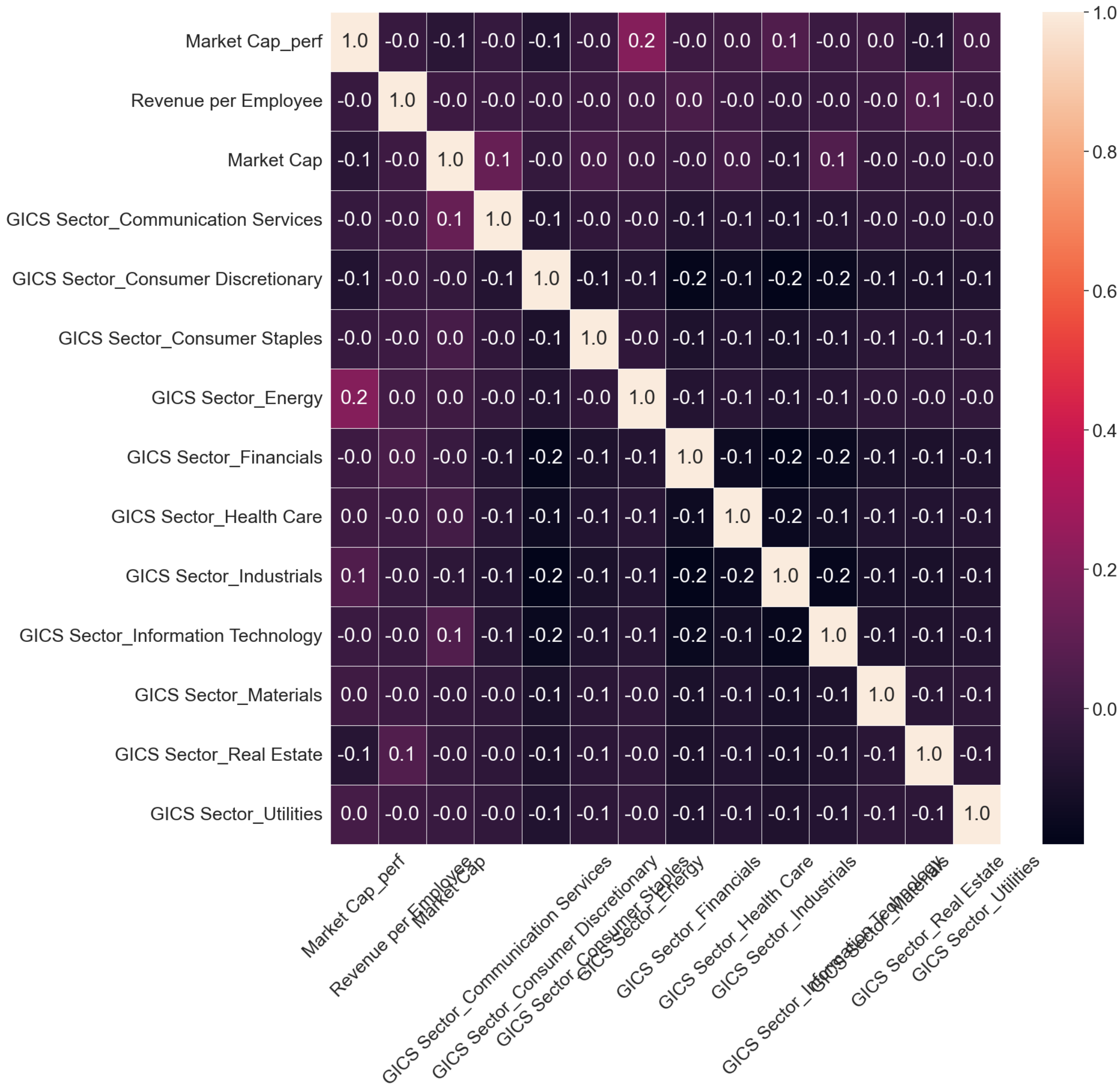



```
[ ]: # saving the pre-processed dfs for easy access

     P1 = pd.read_pickle("data/P1.pkl")
```

```
P2 = pd.read_pickle("data/P2.pkl")
P3 = pd.read_pickle("data/P3.pkl")
CrP = pd.read_pickle("data/CrP.pkl")
```

[ ]:
```
# creating a dataframe consisting of all data across periods
all_dfs = [P1, P2, P3, CrP]
all_df = pd.concat(all_dfs)

print("\nData description for all periods (including current period)")
perf = all_df.pop("Market Cap_perf")
all_df.insert(0, "Market Cap_perf", perf)

rpe = all_df.pop("Revenue per Employee")
all_df.insert(1, "Revenue per Employee", rpe)

print(all_df["GICS Sector"].value_counts(ascending=False))
print(all_df.iloc[:, :12].describe())

scatterplots(all_df)

heatmap(all_df)
```

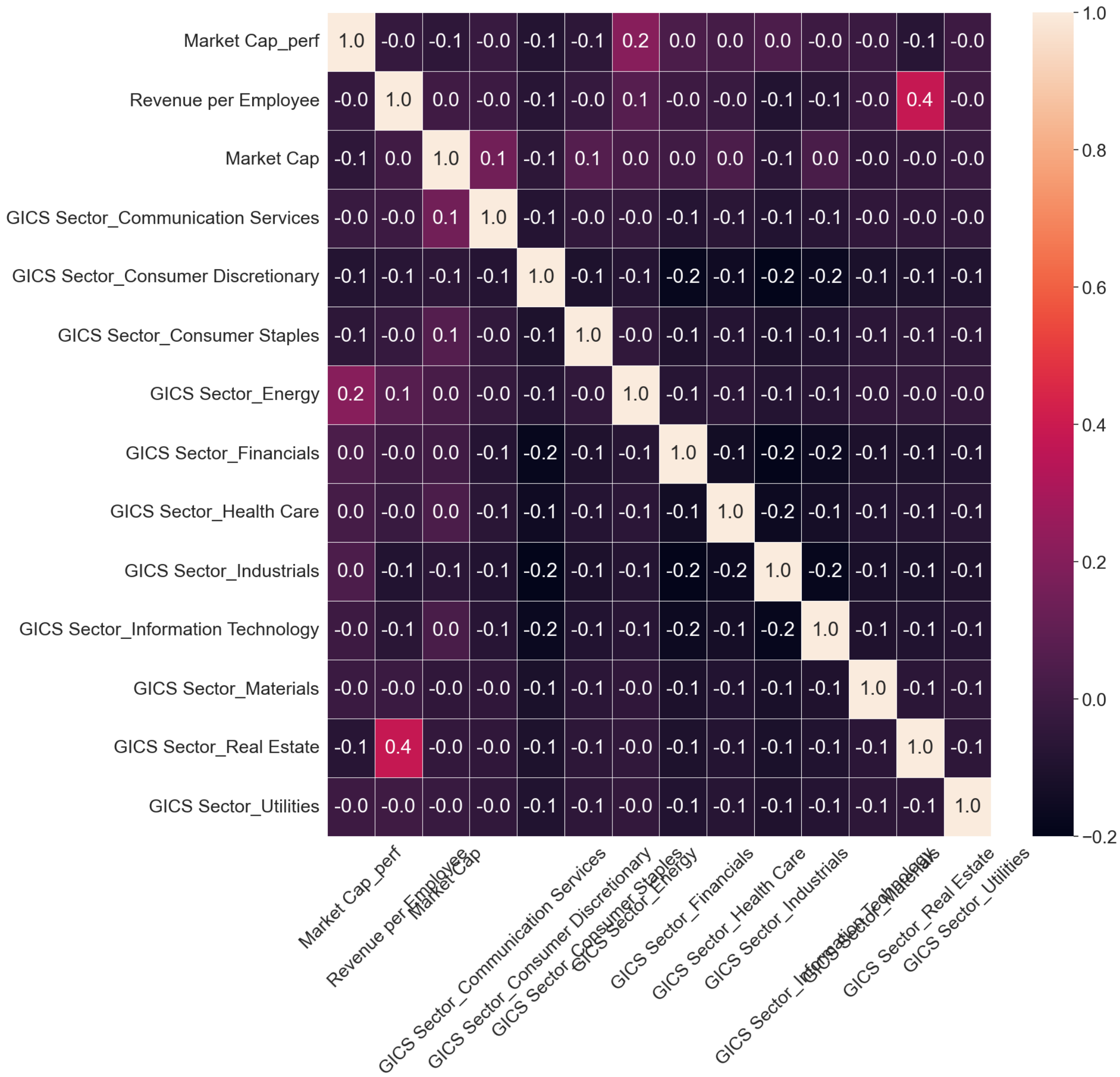
```
Data description for all periods (including current period)
GICS Sector
Industrials               786
Consumer Discretionary    710
Financials                702
Information Technology    587
Health Care               498
Materials                 286
Real Estate               260
Consumer Staples          247
Utilities                 195
Communication Services    156
Energy                    153
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee    Market Cap
count      4580.000000          4.580000e+03  4.580000e+03
mean          0.089550          1.051530e+06  2.226057e+04
std           0.414938          1.304172e+07  9.347337e+04
min          -0.925602          0.000000e+00  1.058600e+02
25%          -0.143823          2.424445e+05  1.559422e+03
50%           0.049928          3.851413e+05  4.083165e+03
75%           0.251214          7.485898e+05  1.406112e+04
max           5.033335          8.679870e+08  2.986128e+06
```




```
[ ]: # creating a df consisting only of concluded periods

     concluded_dfs = [P1, P2, P3]
     concluded_df = pd.concat(concluded_dfs)

     print("\nData decription for only concluded periods")
     perf = concluded_df.pop("Market Cap_perf")
     concluded_df.insert(0, "Market Cap_perf", perf)

     rpe = concluded_df.pop("Revenue per Employee")
     concluded_df.insert(1, "Revenue per Employee", rpe)

     print(concluded_df["GICS Sector"].value_counts(ascending=False))
```

```python
print(concluded_df.iloc[:, :12].describe())

scatterplots(concluded_df)

heatmap(concluded_df)
```

```
Data decription for only concluded periods
GICS Sector
Industrials               582
Consumer Discretionary    523
Financials                488
Information Technology    424
Health Care               351
Materials                 209
Real Estate               187
Consumer Staples          174
Utilities                 147
Communication Services    115
Energy                    110
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee     Market Cap
count      3310.000000          3.310000e+03    3310.000000
mean          0.181060          8.098693e+05   16541.132647
std           0.420463          2.363555e+06   47896.088008
min          -0.916199          2.241154e+04     105.860000
25%          -0.045230          2.300026e+05    1401.645550
50%           0.130913          3.599069e+05    3471.415900
75%           0.315884          6.920711e+05   11867.860650
max           5.033335          5.176925e+07  847355.653400
```

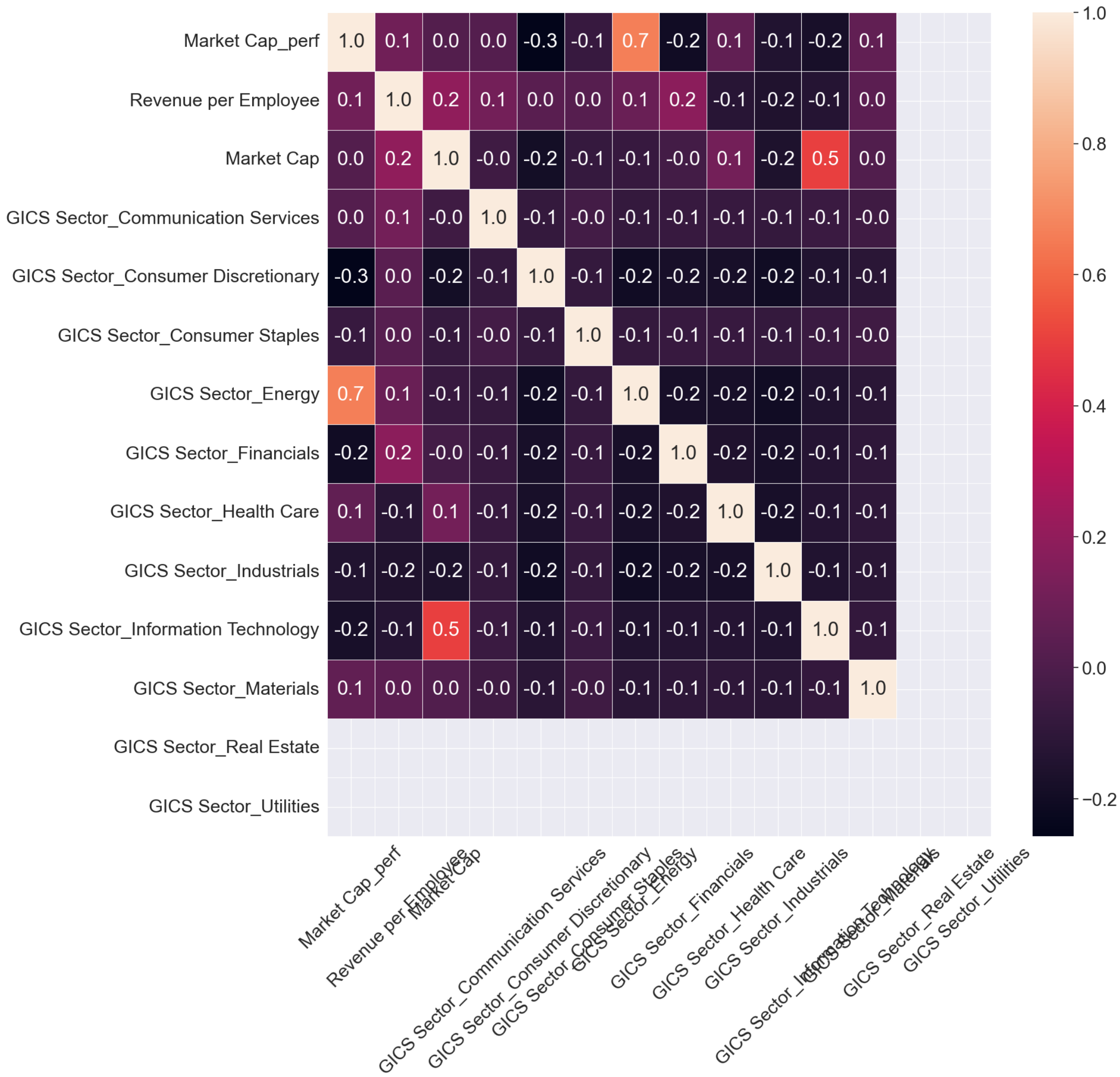


[ ]:

```
# insight into the characteristics of the best performing observations

top10 = pd.DataFrame()

for df in concluded_dfs:
    top10 = pd.concat(
        [top10, df.sort_values("Market Cap_perf", ascending=False).head(10)]
    )

perf = top10.pop("Market Cap_perf")
top10.insert(0, "Market Cap_perf", perf)

rpe = top10.pop("Revenue per Employee")
top10.insert(1, "Revenue per Employee", rpe)
```

```python
print(top10["GICS Sector"].value_counts(ascending=False))
print(top10.iloc[:, :12].describe())

top10.sort_values("Market Cap_perf", ascending=False)

heatmap(top10)
```

```
GICS Sector
Energy                    5
Industrials               5
Consumer Discretionary    5
Health Care               4
Financials                4
Information Technology    3
Materials                 2
Communication Services    1
Consumer Staples          1
Name: count, dtype: int64
       Market Cap_perf  Revenue per Employee    Market Cap
count        30.000000          3.000000e+01     30.000000
mean          2.280571          5.911873e+05   3358.930623
std           0.977822          4.769266e+05   6469.283667
min           1.308081          1.006522e+05    245.865200
25%           1.500476          2.704801e+05    473.586450
50%           1.992901          4.660659e+05   1135.774550
75%           2.539989          8.024246e+05   2422.908625
max           5.033335          2.353037e+06  33785.250000
```



1.3 3. Machine learning

1.3.1 Basic Decision Tree

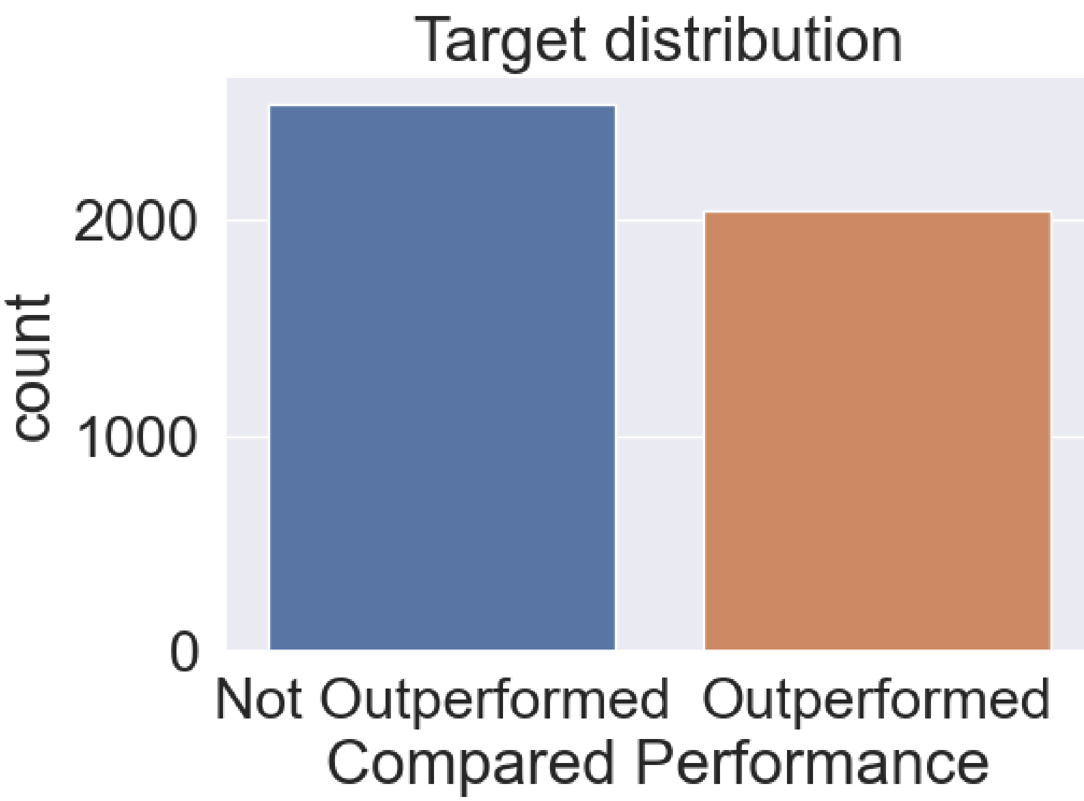
[ ]:
```python
from sklearn.metrics import confusion_matrix, accuracy_score

# function to quickly retrieve minimal evaluation data for the classifier


def evaluate(clf, X_train, X_test, y_train, y_test):
    print("Train Accuracy :", accuracy_score(y_train, clf.predict(X_train)))
    print("Train Confusion Matrix:")
    print(confusion_matrix(y_train, clf.predict(X_train)))
    print("Test Accuracy :", accuracy_score(y_test, clf.predict(X_test)))
```

```python
    print("Test Confusion Matrix:")
    print(confusion_matrix(y_test, clf.predict(X_test)))
```

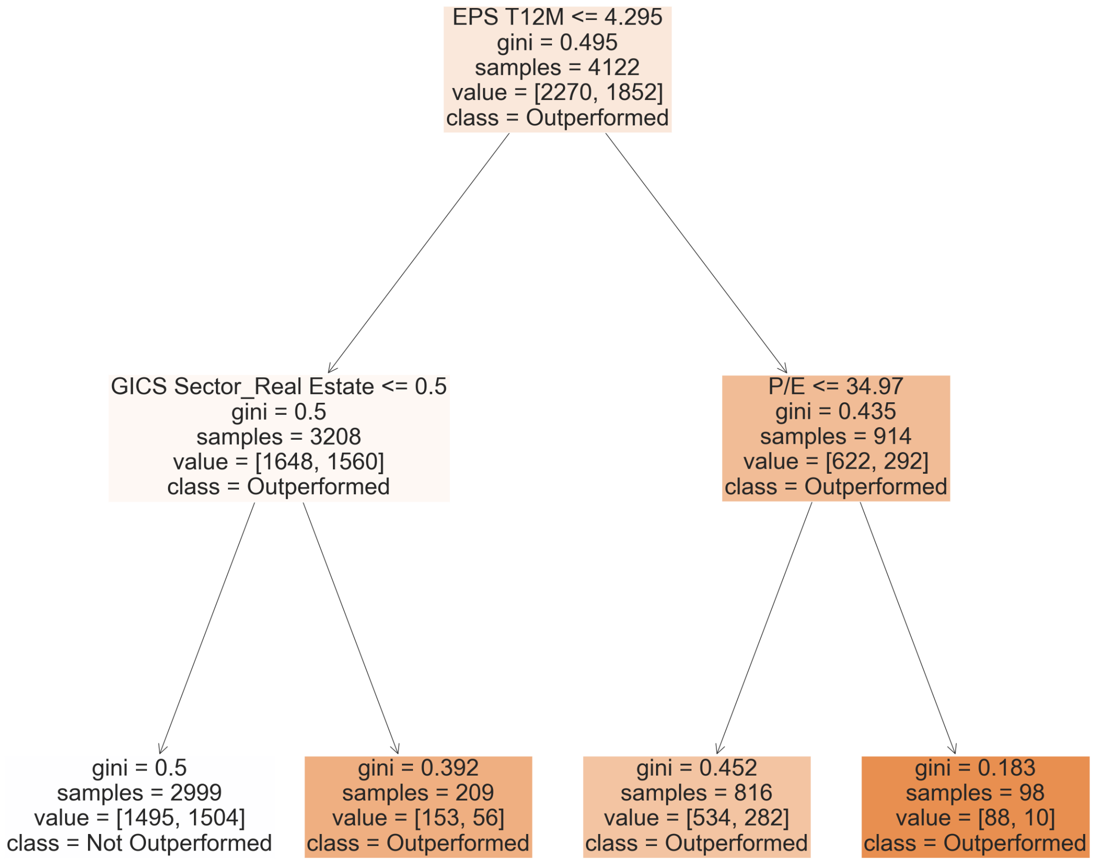
[ ]:
```python
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.tree import plot_tree


def decision_tree(df, title):
    df["Compared Performance"] = np.where(
        (df["Market Cap_perf"] > df["Market Cap_perf"].mean()),
        "Outperformed",
        "Not Outperformed",
    )
    y = df["Compared Performance"]
    X = df.drop(
        columns=["Compared Performance",
                 "Market Cap_perf", "Name", "GICS Sector"]
    )

    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.1, random_state=222
    )

    fig, ax = plt.subplots(figsize=(6, 4))
    sns.countplot(x="Compared Performance", data=df)
    plt.title("Target distribution")
    plt.show()

    print(f"X_train : {X_train.shape}")
    print(f"y_train : {y_train.shape}")
    print(f"X_test : {X_test.shape}")
    print(f"y_test : {y_test.shape}")

    print(title)
    dt = DecisionTreeClassifier(max_depth=2)

    dt.fit(X_train, y_train)

    fig, ax = plt.subplots(figsize=(30, 30))
    plot_tree(
        dt,
        feature_names=X.columns,
        class_names=["Outperformed", "Not Outperformed"],
        filled=True,
        proportion=False,
        fontsize=35,
```

```
    )

    evaluate(dt, X_train, X_test, y_train, y_test)
```

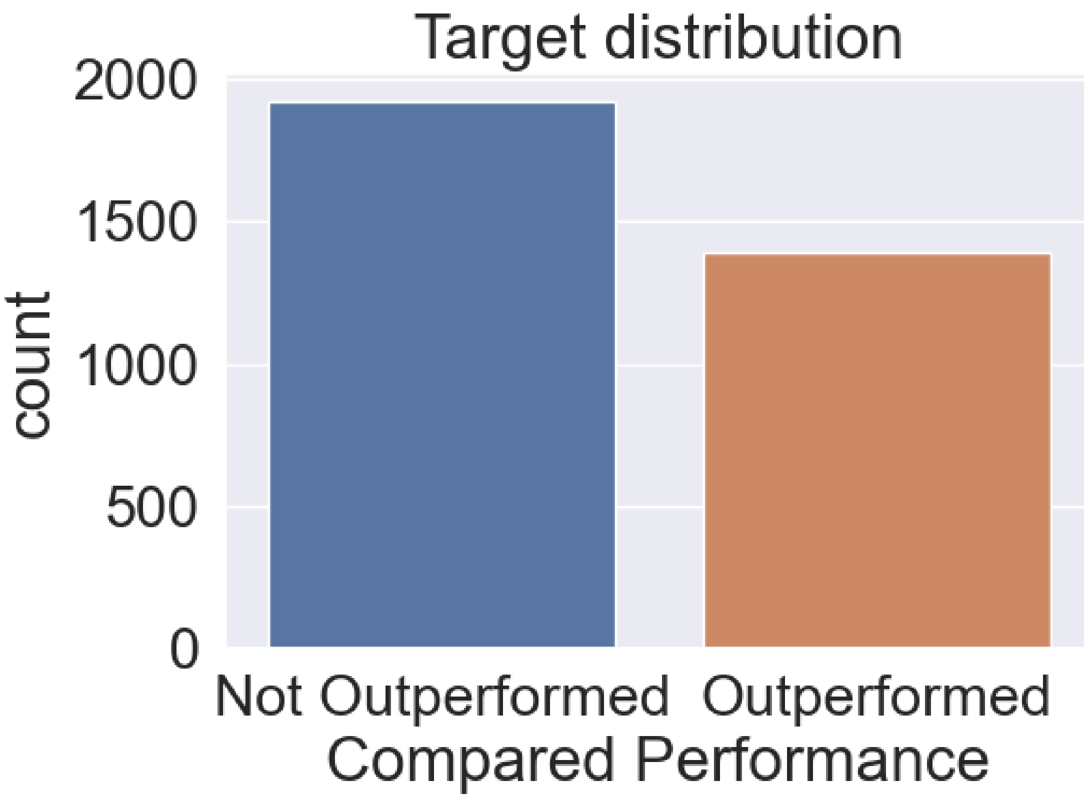
[ ]:
```
decision_tree(
    all_df, "Using the data of all periods (including the current, ongoing one)"
)
```



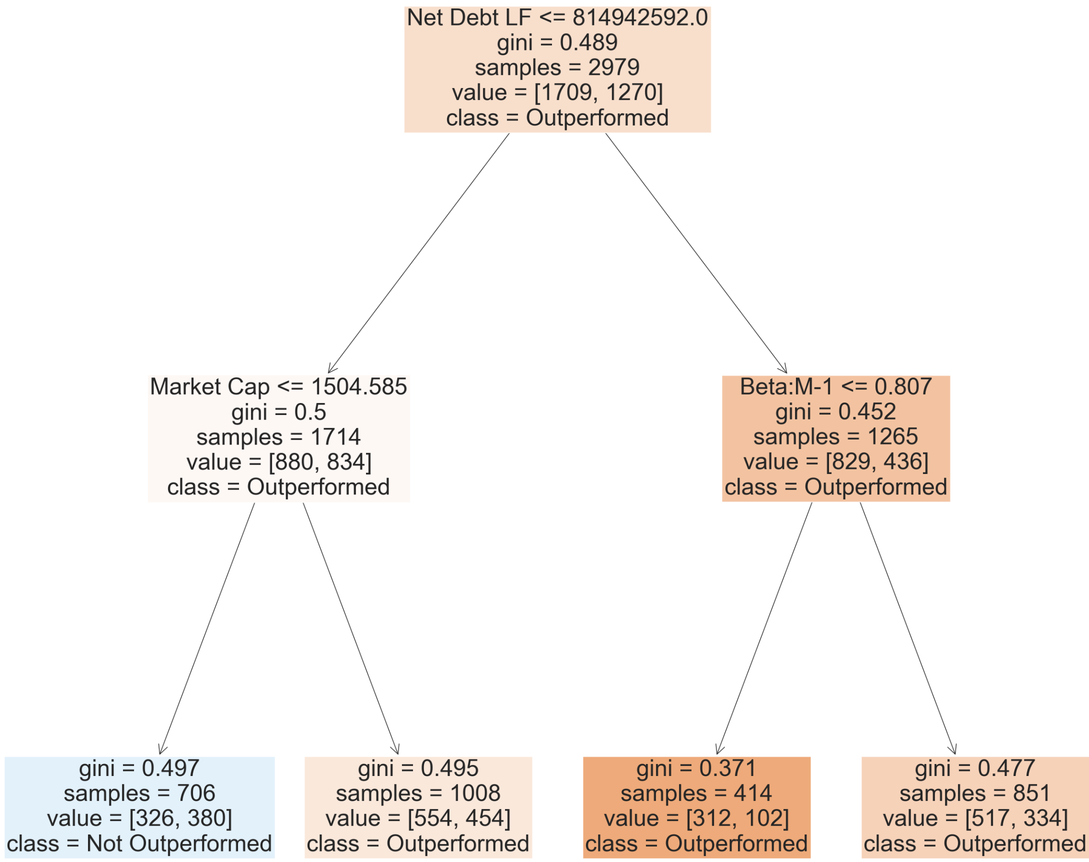

```
X_train : (4122, 20)
y_train : (4122,)
X_test : (458, 20)
y_test : (458,)
Using the data of all periods (including the current, ongoing one)
Train Accuracy : 0.5528869480834546
Train Confusion Matrix:
[[ 775 1495]
 [ 348 1504]]
Test Accuracy : 0.5
Test Confusion Matrix:
[[ 80 185]
 [ 44 149]]
```



[ ]:
```
decision_tree(concluded_df, "Using the data of only concluded periods")
```

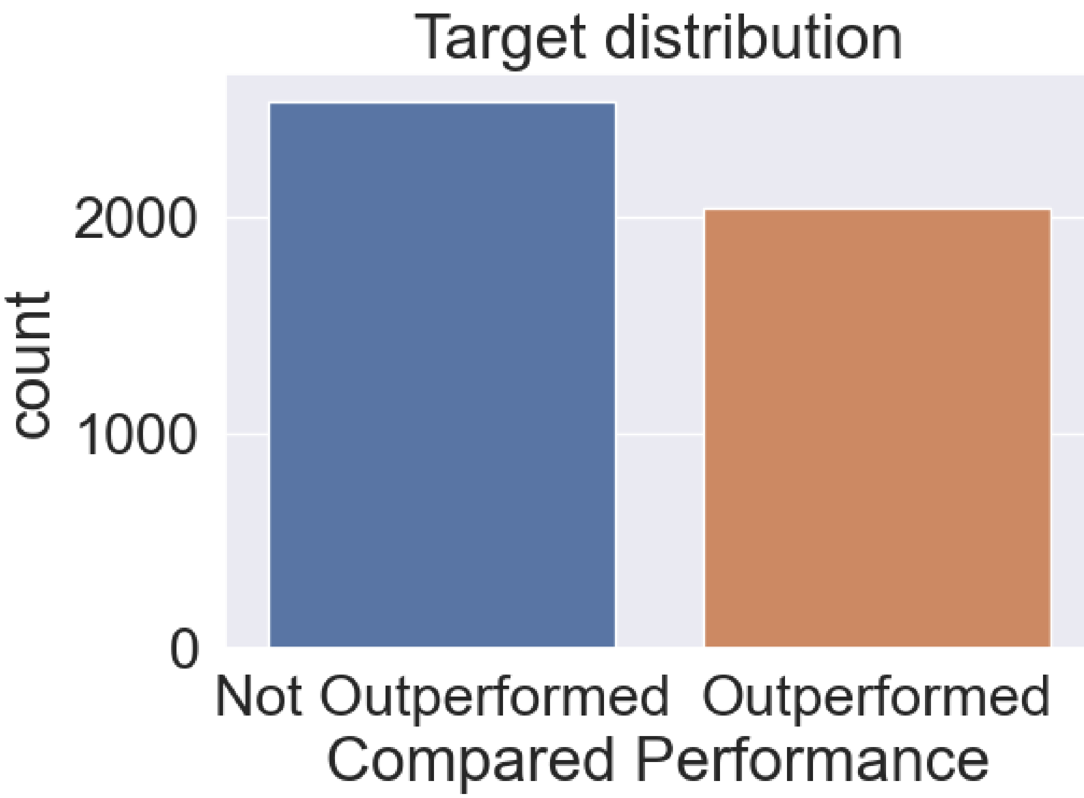



```
X_train : (2979, 20)
y_train : (2979,)
X_test : (331, 20)
y_test : (331,)
Using the data of only concluded periods
Train Accuracy : 0.5918093319906008
Train Confusion Matrix:
[[1383  326]
 [ 890  380]]
Test Accuracy : 0.5981873111782477
Test Confusion Matrix:
[[164  48]
 [ 85  34]]
```

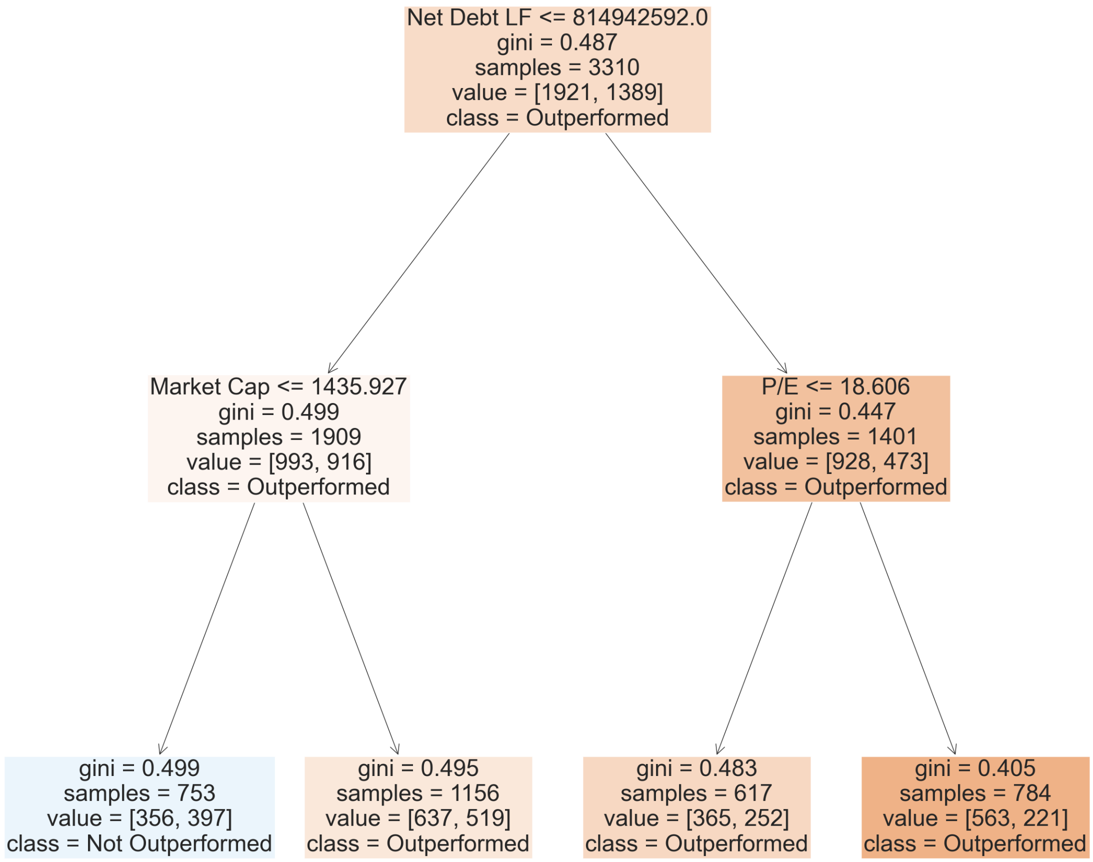


[ ]:
```
concluded_df["Compared Performance"] = np.where(
    (concluded_df["Market Cap_perf"] > concluded_df["Market Cap_perf"].mean()),
    "Outperformed",
    "Not Outperformed",
)
y_train = concluded_df["Compared Performance"]
X_train = concluded_df.drop(
    columns=["Compared Performance", "Market Cap_perf", "Name", "GICS Sector"]
)

CrP["Compared Performance"] = np.where(
    (CrP["Market Cap_perf"] > CrP["Market Cap_perf"].mean()),
    "Outperformed",
```

```
    "Not Outperformed",
)
y_test = CrP["Compared Performance"]
X_test = CrP.drop(
    columns=["Compared Performance", "Market Cap_perf", "Name", "GICS Sector"]
)

fig, ax = plt.subplots(figsize=(6, 4))
sns.countplot(x="Compared Performance", data=all_df)
plt.title("Target distribution")
plt.show()

print(f"X_train : {X_train.shape}")
print(f"y_train : {y_train.shape}")
print(f"X_test : {X_test.shape}")
print(f"y_test : {y_test.shape}")

print(
    "Using the data of concluded periods as training data and the current,␣
 ↪ongoing period data, as test data"
)
dt = DecisionTreeClassifier(max_depth=2)

dt.fit(X_train, y_train)

fig, ax = plt.subplots(figsize=(30, 30))
plot_tree(
    dt,
    feature_names=X_train.columns,
    class_names=["Outperformed", "Not Outperformed"],
    filled=True,
    proportion=False,
    fontsize=35,
)

evaluate(dt, X_train, X_test, y_train, y_test)
```




```
X_train : (3310, 20)
y_train : (3310,)
X_test : (1270, 20)
y_test : (1270,)
Using the data of concluded periods as training data and the current, ongoing
period data, as test data
Train Accuracy : 0.592749244712991
Train Confusion Matrix:
[[1565  356]
 [ 992  397]]
Test Accuracy : 0.5062992125984253
Test Confusion Matrix:
[[571  85]
 [542  72]]
```



1.3.2 3.2 Random Forest with Hyperparametertuning

[ ]:
```
n_estimators = np.arange(50, 250, 50)
max_features = ["auto", "sqrt"]
max_depth = np.arange(2, 20, 1)
min_samples_leaf = [1, 5, 25, 50]
min_samples_split = [2, 5, 25, 50]
max_leaf_nodes = [50, 100, 250, 500]
bootstrap = [True, False]

params_arr = {
    "n_estimators": n_estimators,
    "max_features": max_features,
```

```
    "max_depth": max_depth,
    "min_samples_leaf": min_samples_leaf,
    "min_samples_split": min_samples_split,
    "max_leaf_nodes": max_leaf_nodes,
    "bootstrap": bootstrap,
}
```

[ ]:
```
from treeinterpreter import treeinterpreter as ti
import os
from sklearn.tree import export_graphviz
import six
import pydot
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

rf = RandomForestClassifier()


def visualize_best(gridname, grid, X_test, X_train, y_test, y_train):
    params = {
        "n_estimators": [grid.best_params_["n_estimators"]],
        "max_features": [grid.best_params_["max_features"]],
        "max_depth": [grid.best_params_["max_depth"]],
        "min_samples_leaf": [grid.best_params_["min_samples_leaf"]],
        "min_samples_split": [grid.best_params_["min_samples_split"]],
        "max_leaf_nodes": [grid.best_params_["max_leaf_nodes"]],
        "bootstrap": [grid.best_params_["bootstrap"]],
    }

    clf = RandomForestClassifier(
        n_estimators=params["n_estimators"][0],
        max_features=params["max_features"][0],
        max_depth=params["max_depth"][0],
        min_samples_leaf=params["min_samples_leaf"][0],
        min_samples_split=params["min_samples_split"][0],
        max_leaf_nodes=params["max_leaf_nodes"][0],
        bootstrap=params["bootstrap"][0],
    )

    clf.fit(X_train.values, y_train.values)

    prediction, bias, contributions = ti.predict(clf, X_test.values)

    N = len(X_test.columns)

    outperformed = []
    not_outperformed = []
```

```
    for j in range(2):
        list_ = [outperformed, not_outperformed]
        for i in range(N - 1):
            val = contributions[0, i, j]
            list_[j].append(val)

    outperformed.append(prediction[0, 0] / N)
    not_outperformed.append(prediction[0, 1] / N)

    fig, ax = plt.subplots()
    ind = np.arange(N)

    width = 0.5

    p1 = ax.bar(ind, outperformed, width, color="green", bottom=0)
    p2 = ax.bar(ind + width, not_outperformed, width, color="red", bottom=0)

    ax.set_title("Feature importance for performance result")
    ax.set_xticks(ind + width / 2)
    ax.set_xticklabels(X_train.columns, rotation=90)
    ax.legend((p1[0], p2[0]), ("Outperformed",
              "Not Outperformed"), loc="upper left")
    ax.autoscale_view()

    fig.set_figwidth(15)
    plt.show()

    dotfile = six.StringIO()

    i = 0
    for tree_in_forest in clf.estimators_:
        export_graphviz(
            tree_in_forest,
            out_file="trees/" + gridname + "tree.dot",
            feature_names=X_train.columns,
            filled=True,
        )
        (graph,) = pydot.graph_from_dot_file("trees/" + gridname + "tree.dot")
        name = gridname + "tree_" + str(i)
        graph.write_png("trees/" + name + ".png")
        os.system("dot -Tpng tree.dot -o tree.png")
        i += 1
```

[ ]:
```
def params(X_train, y_train):
    rf_Grid = GridSearchCV(
        estimator=rf, param_grid=params_arr, cv=4, verbose=3, n_jobs=-1
```

```
    )
    rf_Grid.fit(X_train.values, y_train.values)

    params = {
        "n_estimators": [rf_Grid.best_params_["n_estimators"]],
        "max_features": [rf_Grid.best_params_["max_features"]],
        "max_depth": [rf_Grid.best_params_["max_depth"]],
        "min_samples_leaf": [rf_Grid.best_params_["min_samples_leaf"]],
        "min_samples_split": [rf_Grid.best_params_["min_samples_split"]],
        "max_leaf_nodes": [rf_Grid.best_params_["max_leaf_nodes"]],
        "bootstrap": [rf_Grid.best_params_["bootstrap"]],
    }

    print(params)

    return params
```

[ ]:
```
# current is test
current_is_test_params = params(X_train, y_train)

current_test_Grid = GridSearchCV(
    estimator=rf, param_grid=current_is_test_params, cv=4, verbose=3, n_jobs=-1
)
current_test_Grid.fit(X_train.values, y_train.values)

evaluate(
    current_test_Grid, X_train.values, X_test.values, y_train.values, y_test.
  ↪values
)

visualize_best("current_test_", current_test_Grid,
               X_train, X_test, y_train, y_test)

CrP.drop(["Compared Performance"], axis=1, inplace=True)

current_test_Grid
```

```
Fitting 4 folds for each of 18432 candidates, totalling 73728 fits
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```

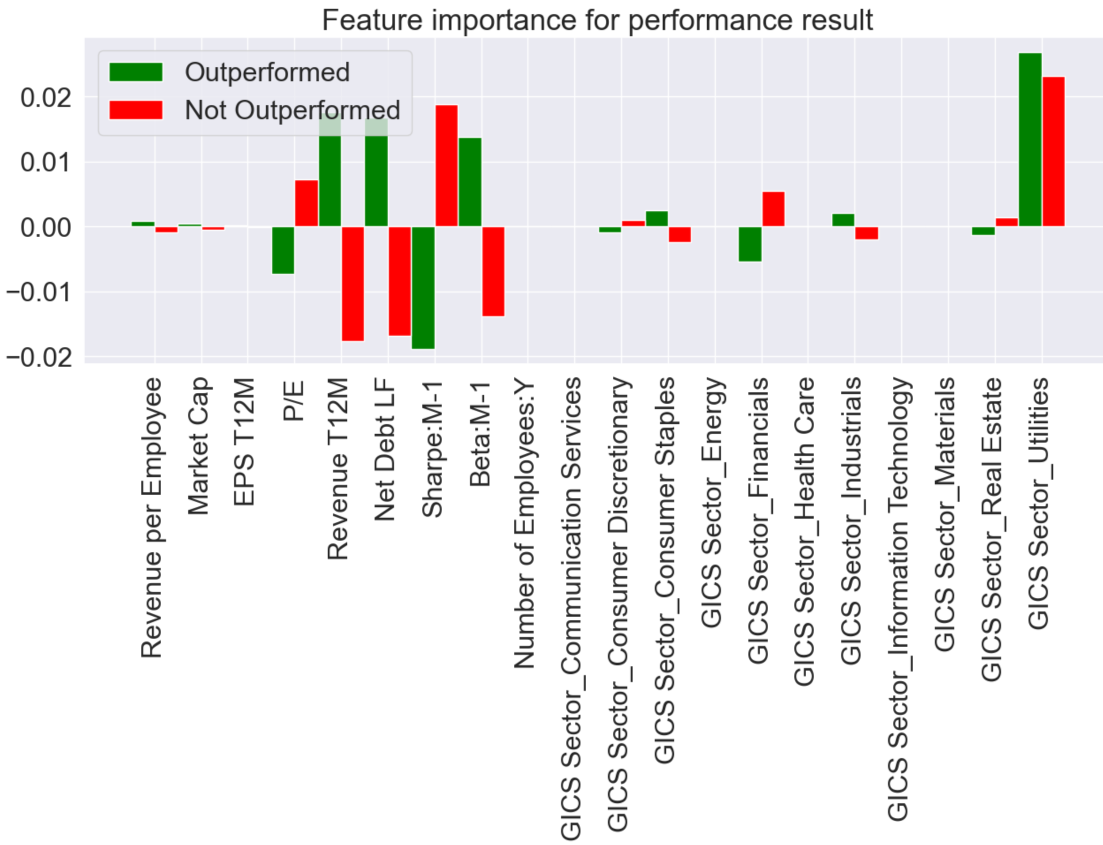
```
{'n_estimators': [50], 'max_features': ['auto'], 'max_depth': [2],
```

```
'min_samples_leaf': [50], 'min_samples_split': [25], 'max_leaf_nodes': [500],
'bootstrap': [False]}
Fitting 4 folds for each of 1 candidates, totalling 4 fits
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```

```
Train Accuracy : 0.5806646525679758
Train Confusion Matrix:
[[1917    4]
 [1384    5]]
Test Accuracy : 0.5149606299212598
Test Confusion Matrix:
[[651   5]
 [611   3]]
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```



[ ]:
```
GridSearchCV(cv=4, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'bootstrap': [False], 'max_depth': [2],
                         'max_features': ['auto'], 'max_leaf_nodes': [500],
                         'min_samples_leaf': [50], 'min_samples_split': [25],
                         'n_estimators': [50]},
             verbose=3)
```

[ ]:
```
# all_df
y = all_df["Compared Performance"]
X = all_df.drop(
    columns=["Compared Performance", "Market Cap_perf", "Name", "GICS Sector"]
)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.1, random_state=222
)

all_params = params(X_train, y_train)

all_Grid = GridSearchCV(
```

```
    estimator=rf, param_grid=all_params, cv=4, verbose=3, n_jobs=-1)
all_Grid.fit(X_train.values, y_train.values)

evaluate(all_Grid, X_train.values, X_test.values,
         y_train.values, y_test.values)

visualize_best("all_", all_Grid, X_train, X_test, y_train, y_test)

all_Grid
```

```
Fitting 4 folds for each of 18432 candidates, totalling 73728 fits
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```

```
{'n_estimators': [50], 'max_features': ['auto'], 'max_depth': [9],
'min_samples_leaf': [5], 'min_samples_split': [5], 'max_leaf_nodes': [500],
'bootstrap': [False]}
Fitting 4 folds for each of 1 candidates, totalling 4 fits
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```

```
Train Accuracy : 0.8027656477438136
Train Confusion Matrix:
[[1898  372]
 [ 441 1411]]
Test Accuracy : 0.6179039301310044
Test Confusion Matrix:
[[185  80]
 [ 95  98]]
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
```

ExtraTreesClassifiers.

[ ]:
```
GridSearchCV(cv=4, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'bootstrap': [False], 'max_depth': [9],
                         'max_features': ['auto'], 'max_leaf_nodes': [500],
                         'min_samples_leaf': [5], 'min_samples_split': [5],
                         'n_estimators': [50]},
             verbose=3)
```

[ ]:
```python
# concluded_df
y = concluded_df["Compared Performance"]
X = concluded_df.drop(
    columns=["Compared Performance", "Market Cap_perf", "Name", "GICS Sector"]
)

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.1, random_state=222
)

concluded_params = params(X_train, y_train)
```

```python
concluded_Grid = GridSearchCV(
    estimator=rf, param_grid=concluded_params, cv=4, verbose=3, n_jobs=-1
)
concluded_Grid.fit(X_train.values, y_train.values)

evaluate(concluded_Grid, X_train.values,
         X_test.values, y_train.values, y_test.values)

visualize_best("concluded_", concluded_Grid, X_train, X_test, y_train, y_test)

concluded_Grid
```

```
Fitting 4 folds for each of 18432 candidates, totalling 73728 fits
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```

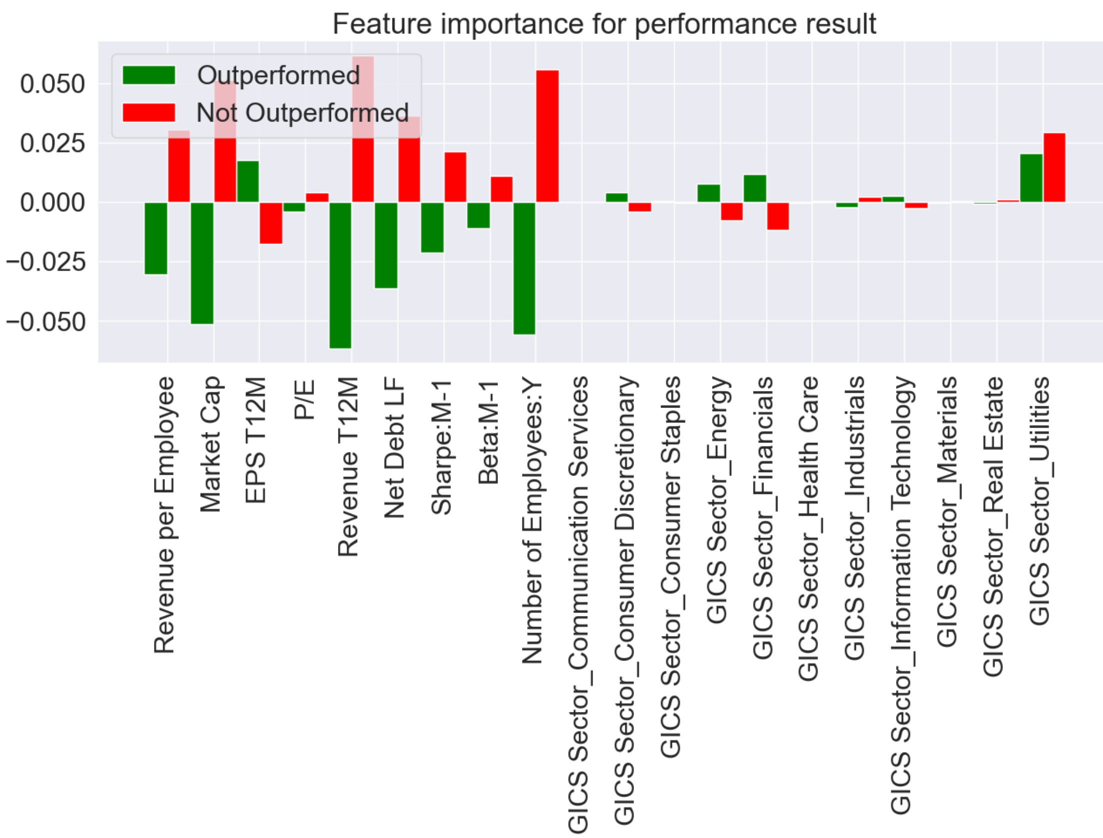
```
{'n_estimators': [50], 'max_features': ['auto'], 'max_depth': [17],
'min_samples_leaf': [5], 'min_samples_split': [5], 'max_leaf_nodes': [500],
'bootstrap': [True]}
Fitting 4 folds for each of 1 candidates, totalling 4 fits
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:

`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```

```
Train Accuracy : 0.9315206445115811
Train Confusion Matrix:
[[1676   33]
 [ 171 1099]]
Test Accuracy : 0.6706948640483383
Test Confusion Matrix:
[[165  47]
 [ 62  57]]
```

```
c:\Users\lucar\anaconda3\envs\stuffed\lib\site-
packages\sklearn\ensemble\_forest.py:424: FutureWarning:
```

```
`max_features='auto'` has been deprecated in 1.1 and will be removed in 1.3. To
keep the past behaviour, explicitly set `max_features='sqrt'` or remove this
parameter as it is also the default value for RandomForestClassifiers and
ExtraTreesClassifiers.
```



[ ]:
```
GridSearchCV(cv=4, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'bootstrap': [True], 'max_depth': [17],
                         'max_features': ['auto'], 'max_leaf_nodes': [500],
                         'min_samples_leaf': [5], 'min_samples_split': [5],
                         'n_estimators': [50]},
             verbose=3)
```

1.4 3.3 Predictor implementation

COMPARED TO INVESTING INTO THE SP1500 ONLY

Using the findings from above we can determine the companies predicted to outperform

[ ]:
```
# we add all companies that have been identified as "Outperformed" and assign␣
↪equal portfolio weights
all_classifier_performance = []
```

```
concluded_classifier_performance = []
current_test_classifier_performance = []

# calculating the portfolio performances
for df in all_dfs:
    current_test_outperformed = []
    concluded_outperformed = []
    all_outperformed = []

    compare_df = df.drop(["Market Cap_perf", "Name", "GICS Sector"], axis=1)
    for i in range(df.shape[0]):
        if current_test_Grid.predict([compare_df.iloc[i].values]) ==␣
↪"Outperformed":
            current_test_outperformed.append(df.iloc[i, 0])
        if all_Grid.predict([compare_df.iloc[i].values]) == "Outperformed":
            all_outperformed.append(df.iloc[i, 0])
        if concluded_Grid.predict([compare_df.iloc[i].values]) ==␣
↪"Outperformed":
            concluded_outperformed.append(df.iloc[i, 0])

    print(current_test_outperformed)
    print(all_outperformed)
    print(concluded_outperformed)

    current_test_classifier_performance.append(1 + np.
↪mean(current_test_outperformed))
    all_classifier_performance.append(1 + np.mean(all_outperformed))
    concluded_classifier_performance.append(1 + np.mean(concluded_outperformed))
```

```
[0.5021601011182174]
[-0.5414174630173221, 0.11610251740509339, 0.1506381395146691,
-0.045177658784108496, 0.13700589878376465, -0.06245373392666296,
-0.5242943917359884, 1.6368671160494397, -0.4445325980698409,
0.14080354187486238, 0.2482193468083046, 0.1137348283116921,
-0.17786349692068212, -0.23285434772813982, 0.4479053497444085,
2.3444866610105337, 0.5759779856664411, 1.4922006594926192, 0.5088552126082243,
0.20225914491572725, -0.3223220473617356, 0.2976992602286255,
0.1978238936817814, -0.4434762489388159, -0.6872113904478114,
0.9834763778207347, 0.9182991379097157, 0.21209757523786998,
-0.2578818568641189, -0.05147067528865498, 0.9248547469124445,
0.7341436150696985, -0.514181007110329, -0.4027423995610414,
-0.3249247733423064, -0.42707816479168115, 0.19536196783858029,
0.203147001042724, 0.2491877142308661, 0.7057514429205799, -0.3807508746053108,
0.6019448310473624, 0.2370513660564978, 0.5669679119190111, -0.7237308955544142,
-0.5333750873364139, 0.23123840715236918, 0.4143972575514973,
-0.7047114874702913, -0.26853905849409826, -0.4828057866331675,
0.6699160429822983, 0.20362131299398276, 0.10644115423489538,
```

0.35222304256179277, -0.16781792735057977, -0.002016233125294143,
0.5313786447707309, 0.5096353161359051, -0.5927521880363326, 0.4593864216253106,
0.3047741278341176, -0.23584900336189668, 1.3343218350781538,
-0.2674793896950261, -0.20402050091772464, 0.4817158111041038,
0.2744342253038181, -0.14870519323819356, 0.35578126886547917,
0.3097539462601414, 0.21813485371393798, 0.2668428505267832, -0.14663948297163,
0.437259288652291, 0.47529690528769214, 0.4980521377429851, 0.3692611096434666,
0.6845449200288876, 0.3526857842022358, 0.313933261241152, 0.13456949508157962,
1.6328155178376118, 0.42859607173399017, 0.03845268921796314,
0.46880869351717624, -0.2231227232899568, 0.5627500101391667, 2.328803875457947,
1.9125441762295279, 1.4112458411344866, 0.7386051748885498, 0.6846614024722847,
0.8780720953131759, 1.577699780884338, 1.171268687356031, 3.9755592638263932,
1.4247178469448563, 1.0331157067443635, 2.2484648072132054, 4.482588648024972,
1.607282737570318, 0.6878971577048497, 1.929871068982061, 0.5580502860267379,
1.2299164762902146, 1.240087893941281, 1.9506786632200588, 0.5150900481317011,
0.9853032287900783, 0.7012275440137224, 0.5606254133855006, 2.56939055847459,
1.649743629591535, 0.8181926171756307, 1.4035993967813265, 1.6331133321107316,
1.630704708744513, 1.7914608195688249, 5.033334675048552, 1.1918975481792347,
0.5910403071472786, -0.039170287617133266, 2.2690422643468855,
0.6693130838020735, 0.4842791259651729, 2.4517854057780357, 0.2707763754604936,
0.5960846767494155, -0.0017150483471995367, -0.11506629828185003,
0.4461452977237966, -0.056014115129762586, -0.011151134212528328,
0.14986805022495894, 0.11965256948712025, 0.12537098001707728,
0.18242542632691047, 0.23555304739227123, 0.23476835597612522,
-0.010988768906962747, 0.47275264764535563, -0.4837011046760249,
0.2527864262699388, 0.4990026487685499, 0.45942393458787634,
0.20250781281158137, 0.18435770315972413, 0.4925135856387481,
0.0035445587963678364, 0.15993535454701044, 0.6074695321589924,
-0.192926061347743, 0.09418176882553055, -0.0780169506554931,
-0.023946189149467566, 0.1743170930318638, 2.3655006496405506,
0.10454716902692711, 0.46268214354440607, -0.1676576707699076,
-0.011863974836106848, 1.585011502450826, 0.6904919008653627,
0.5813475269444487, 0.1907100999397022, 0.030255450652868533,
0.2164722713575743, 0.9419675318319924, 0.1495371710875427, 0.2359958563178308,
0.3474582825896819, 0.24009115363557731, 0.6030039528648313,
0.17049663711891272, -0.11630472329439923, 0.3265817324426125,
0.19565885543808328, -0.5506577360264467, 0.2075940886297987,
0.41101233905778445, 0.3496527870256465, 0.6903997083418368, 0.0995008254721791,
0.9527768751456592, 0.48896477874041744, 2.1674881669050547, 0.6730462337345446,
0.44076937712067366, 0.17952750730687822, 0.4654293484298403,
0.37260682490968133, 0.7312994197073449, 0.3886534977024654,
-0.17906427477925685, 0.8268229903994855, -0.27930623325049403,
0.23842888562588915, -0.3078622673958172, 0.5357483950821902,
-0.009463028200875091, 0.05476326289640393, 1.326331751112263,
-0.3974655758564839, 3.897444003344262, 0.40914695851773564,
0.20958158833906718, -0.34589213588849393, -0.0321358836114437,
0.13262036199126992, 0.6234010862308914, 0.888892932424372, 0.310016129930639,
0.021383145119204788, -0.24172811710173686, 0.10024274262994674,

0.8872677638624891, 1.2982712306127173, 0.9838712394107774,
-0.30043518996761065, 0.2859662464707668, 0.9403801666563228,
-0.15315404556373474, 0.5162339878762836, -0.46692601615166,
0.16743164853740242, 0.2648585987745585, 0.14978518526085124,
0.23784799224426967, 0.18361298268681656, -0.5292546334868099,
2.050244221312007, 0.08422926197702116, 0.47416604054425376,
-0.12185752544316358, 0.28308875852976195, 0.8509924476194859,
1.7939813335099408, 0.3470207195003873, 1.311319577028758, 0.14502186495896724,
0.1393745188926572, -0.6699033935908512, 0.21801764127450884,
-0.39772165195453785, 0.30402692073934984, 0.7887146474815644,
0.3400372999505674, 0.35271213430719506, 0.17335056550672623,
0.29934105954940676, 0.3892970336590765, 0.6932683426724564, 0.748724821769922,
0.13035061153036684, 0.4954609343705863, -0.4534882888078016,
0.1372195356169825, 1.0722216375606994, 0.7659645590773566, 0.9509649560331026,
1.9533584497714929, 0.3660034675186326, 0.04599610112552069, 0.6462720662925779,
0.201229389971671, -0.24729489063263013, 0.9920996522252675, 0.2732854693527646,
0.650127187530345, 0.788618594990843, -0.04645871968358828, 0.2748472903999895,
0.000638196783216478, 0.9951403913843766, 1.1570561090303968,
0.3987221928991689, 1.1266092224873776, 1.0145878487227833, -0.2568327291280308,
0.27276077189514886, 0.016358233396982457, 0.6798727991349135,
0.2990192113517311, 0.6979117515146395, 0.31280121829424296, 0.3208527490570134,
0.4087014478782557, 0.5529874571670721, 0.059801827722662945,
-0.2990393940574302, 1.8259048533040854, 0.13943856816023856, 3.364282541815597,
0.5739982282490672, 0.5425079203934924, 0.13224350162839824, 1.3430358770699073,
0.5904029855346464, 0.4359512517793129, 0.23180555343485834, 1.501619041920574,
1.3993908779206312, 0.8998784760454481, 0.9021792865441817, -0.5912863298348868,
1.6545187605990022, 0.589809240277653, 0.32420603730906594,
-0.24495234353034842, 0.24446015187639958, 0.6828607949193097,
0.2057174076187283, 1.7008881539565048, 0.951614387275517, 0.21030657113562937,
0.1943539373073362, 0.33716489496456575, 0.12512297925131446,
0.3147618699834286, 0.44604124493303954, 0.5558660565157053,
0.44546499903924985, 0.28454255532508943, 1.037837712175086, 1.3960449197995883,
0.7728593738850895, -0.07897556654785792, -0.23381066060520672,
0.36154415270263596, 0.5994255648133814, 0.3270371438918918,
0.02123692548753944, -0.0062776528136101305, 0.23517797876189564,
0.5945148713060092, 0.4050020918727509, 0.7028623325923116, 0.33812236539012835,
0.5620644980839027, 0.27251818019223306, 0.4316763451483097,
0.11205745150702855, -0.49638236308738826, 1.7811857569036744,
-0.3260971015521773, 1.0435391846097162, 0.21215365993588023,
-0.0956652519655588, 0.12994765821872775, -0.06826285372329767,
0.9067514946895133, 0.5098665659540706, 0.7420211311272932, 0.12038462013396986,
-0.2753809235982069, -0.0416371228295124, 1.9831984811174836,
-0.42035388987752365, 0.21716932881776368, -0.3198716061599838,
0.025007783923160873, 0.350785806757741, 0.7079179991337804,
0.21446370014929497, -0.170575572559394, -0.22740323478751887,
-0.18541810251329746, -0.021695677223270607, 0.17849466240980974,
0.8421086382135621, -0.1134798631261269, 1.9412390989768087, 0.5021601011182174,
-0.5758847574040025, 0.3015075298501361, -0.6301102740203033,

0.18903572921269654, -0.44329028891557193, 0.6178841907946888,
-0.14125170504899107, 0.39079394766116754, 1.088034974861129,
-0.11840554890038539, 0.11762334133080654, 1.9628673969861925,
1.1850300466151746, 1.2003057020151537, 0.057655809838151016,
1.4846189148932074, 0.1432901446246588, -0.14294317213693714,
0.25246515680587195, -0.0010595248008287064, 2.074009078979237,
0.5599844149605797, -0.008604403260975809, 0.43956293686934234,
0.2205233974327799, 0.1291622286338565, 0.0807349455354169, 0.5550597992616234,
-0.39578279973549035, 0.3816476461255321, 1.2996349461024355,
0.9373438087033327, 0.0469292957850289, 0.21778134141792707, 0.3535151561997826,
0.5727021174285325, 0.07492184684275184, 0.22311767016604103,
0.042158402465315575, 1.2646071629058768, 0.1882429258060336,
-0.08914625349432138, 0.4814405464782149, 2.082666296692073,
-0.01372243961663761, -0.013684556514739992, 0.31351127377185534,
0.3686747084374569, -0.3761710780899029, 0.4801313476727461, 2.355954372572752,
0.13963158040973894, 0.7580585495700729, 0.11682531448579943,
0.1850618983065182, 0.49146391082546614, 1.3164607855302508, 0.4199463647424391,
0.012329916268312993, 0.15395203436315402, 0.5154266009119173,
0.4554960470332927, 0.22339475561392977, 0.18313326553523018,
0.13553631777835073, 0.7300348815319283, 0.270235519052913, 0.19240404897499983,
0.5890816239377518, 0.18853973404291083, 0.1128177214546634, 0.3807607350520994,
0.30678175648467043, -0.002161755970475987, 0.19490140597849193,
0.420009679344283, 1.114608707997793, 0.5605975654400395, 0.173958941207483,
0.35309729596766476, 0.4742432966990809, 0.8399907129692094, 0.6334259694710219,
0.1454481092242395, 0.7496834436516325, 0.607717586716309, 0.33810383232631636,
1.45202992483495, 0.2986454148699891, 0.17913855937221396, 0.19610561001667784,
0.454181413072701, 0.16008022665783617, 0.13274311309838974,
0.22799254666561797, 0.09259643781091095, 0.12631084363389558,
0.39564986291385784]
[0.4142666543439113, 0.5710721391000113, 0.35838499170652416,
0.4479053497444085, 2.3444866610105337, 0.5759779856664411, 1.4922006594926192,
0.5088552126082243, 0.20225914491572725, 0.2976992602286255, 0.1978238936817814,
0.9834763778207347, 0.9182991379097157, 0.21209757523786998, 0.9248547469124445,
0.7341436150696985, 0.9810806316155359, 0.19536196783858029, 0.203147001042724,
0.2491877142308661, 0.7057514429205799, -0.5500998305839508, 0.6019448310473624,
0.6416284484893454, 0.2370513660564978, -0.5010942840601946, 0.5669679119190111,
0.23123840715236918, 0.3797118149558152, 0.6699160429822983,
0.20362131299398276, 0.35222304256179277, 0.5313786447707309,
0.5096353161359051, 0.5916340754815583, 0.4593864216253106, 0.3047741278341176,
0.5898006188996012, 1.3343218350781538, 0.4817158111041038, 0.2744342253038181,
0.35578126886547917, 0.3097539462601414, 0.21813485371393798,
0.2668428505267832, 0.437259288652291, 0.47529690528769214, 0.4980521377429851,
0.6845449200288876, 0.3526857842022358, 1.6328155178376118, 0.42859607173399017,
0.5651070303930731, 0.03845268921796314, 0.46880869351717624,
-0.2847323326288729, 0.5627500101391667, 2.328803875457947, 1.9125441762295279,
1.4112458411344866, 0.7386051748885498, 0.6846614024722847, 0.8780720953131759,
1.577699780884338, 1.171268687356031, 3.9755592638263932, 1.4247178469448563,
1.0331157067443635, 2.2484648072132054, 4.482588648024972, 1.607282737570318,

0.6878971577048497, 1.929871068982061, 0.5580502860267379, 1.2299164762902146,
1.240087893941281, 1.9506786632200588, 0.9853032287900783, 0.7012275440137224,
2.56939055847459, 1.649743629591535, 0.8181926171756307, 1.4035993967813265,
1.6331133321107316, 1.630704708744513, 1.7914608195688249, 5.033334675048552,
1.1918975481792347, 0.5910403071472786, 0.6693130838020735, 0.4842791259651729,
2.4517854057780357, 0.2707763754604936, -0.0017150483471995367,
0.18242542632691047, 0.23476835597612522, 0.47275264764535563,
0.2527864262699388, 0.4990026487685499, 0.45942393458787634,
0.20250781281158137, 0.18435770315972413, 0.4925135856387481,
0.2369861668713579, 0.6074695321589924, 2.3655006496405506, 0.46268214354440607,
1.585011502450826, 0.5188232611556168, 0.6904919008653627, 0.5813475269444487,
0.1907100999397022, 0.2164722713575743, 0.9419675318319924, 0.2359958563178308,
0.3474582825896819, 0.3998309223545116, 0.24009115363557731, 0.6030039528648313,
0.21718040511072997, 0.3265817324426125, 0.19565885543808328,
0.2075940886297987, 0.41101233905778445, 0.3496527870256465, 0.8496665613025725,
0.6903997083418368, 0.3172004342010345, 0.8159400817786127, 0.9527768751456592,
0.48896477874041744, 2.1674881669050547, 0.6730462337345446,
0.44076937712067366, 0.17952750730687822, 0.4654293484298403,
0.37260682490968133, 0.7312994197073449, 0.3886534977024654, 0.8268229903994855,
-0.3078622673958172, 0.5357483950821902, 1.326331751112263, 3.897444003344262,
0.40914695851773564, 0.31641651788383673, 0.6234010862308914,
0.3160968180246288, 2.691997038509457, 0.310016129930639, -0.24172811710173686,
0.2313939944530757, 0.8872677638624891, 1.2982712306127173, 0.9838712394107774,
1.3057922077174096, 0.47854619082313027, 0.2859662464707668, 0.9403801666563228,
0.5162339878762836, 0.5967740025094195, 0.2648585987745585, 0.23784799224426967,
0.18361298268681656, 2.050244221312007, -0.1437027627734829,
0.28308875852976195, 0.8509924476194859, 0.4153377457854916, 0.3470207195003873,
1.311319577028758, 0.21801764127450884, 0.30402692073934984, 0.7887146474815644,
0.3400372999505674, 0.29934105954940676, 0.3892970336590765, 0.6932683426724564,
0.748724821769922, 0.13035061153036684, 0.4954609343705863, 1.0722216375606994,
0.7659645590773566, 0.9509649560331026, 1.9533584497714929, 0.3660034675186326,
0.6462720662925779, 0.201229389971671, 0.9920996522252675, 0.2732854693527646,
0.650127187530345, 0.788618594990843, 0.2748472903999895, 0.9951403913843766,
1.1570561090303968, -0.2395833588634527, 0.3987221928991689, 1.1266092224873776,
1.0145878487227833, -0.2568327291280308, 0.27276077189514886,
0.9150051242418937, 0.6798727991349135, 0.2990192113517311, 0.6979117515146395,
0.31280121829424296, 0.3208527490570134, 0.4087014478782557, 0.5529874571670721,
-0.2990393940574302, 1.8259048533040854, 3.364282541815597, 0.5739982282490672,
1.3430358770699073, 0.5904029855346464, 0.4359512517793129, 0.3973370241049246,
0.23180555343485834, 1.501619041920574, 0.8998784760454481, 0.6562333617294124,
1.6545187605990022, 0.4450915837470865, 0.589809240277653, 0.32420603730906594,
0.24446015187639958, 0.6828607949193097, 0.42652109042095326,
0.2057174076187283, 1.7008881539565048, 0.951614387275517, 0.3830222153050196,
0.1943539373073362, 0.33716489496456575, 0.3147618699834286,
0.44604124493303954, 0.5558660565157053, 0.28454255532508943, 1.037837712175086,
1.3960449197995883, 0.7728593738850895, 0.36154415270263596, 0.5994255648133814,
0.3270371438918918, -0.0062776528136101305, 0.23517797876189564,
0.5945148713060092, 0.4050020918727509, 0.7028623325923116, 0.33812236539012835,

0.5620644980839027, 0.27251818019223306, 0.4316763451483097, 1.7811857569036744,
1.0435391846097162, 0.21215365993588023, -0.06826285372329767,
0.9067514946895133, 0.7420211311272932, 1.9831984811174836, 0.350785806757741,
0.7079179991337804, 1.1176458313201314, -0.22740323478751887,
0.17849466240980974, 0.8421086382135621, 1.9412390989768087, 0.5021601011182174,
0.3148795714144094, 0.3015075298501361, 0.6178841907946888, 1.088034974861129,
1.9628673969861925, 1.2003057020151537, 1.4846189148932074, 0.25246515680587195,
2.074009078979237, 0.43956293686934234, 0.2205233974327799, 0.5550597992616234,
0.3816476461255321, 1.2996349461024355, 1.4100333660022468, 0.3535151561997826,
0.5727021174285325, 0.22311767016604103, 1.2646071629058768, 0.1882429258060336,
0.4814405464782149, 2.082666296692073, 0.31351127377185534, 0.3686747084374569,
-0.3761710780899029, 2.355954372572752, 0.1850618983065182, 1.3164607855302508,
0.4199463647424391, 0.5154266009119173, 0.4554960470332927, 0.22339475561392977,
0.7300348815319283, 0.19240404897499983, 0.40325471279454916,
0.5890816239377518, 0.3807607350520994, 0.30678175648467043,
0.19490140597849193, 0.420009679344283, 1.114608707997793, 0.35309729596766476,
0.4742432966990809, 0.6334259694710219, 0.5583447709023601, 0.7496834436516325,
0.607717586716309, 0.33810383232631636, 1.45202992483495, 0.2986454148699891,
0.19610561001667784, 0.454181413072701, 0.22799254666561797,
0.39564986291385784, 0.7279442015412911]
[-0.13950249688023003, 0.09504447823183973, 0.3275768668283222,
0.5103125456411304, 0.1216372052213896, 0.040035974078756704]
[-0.08825025604398619, 0.31771380987644465, 0.21365045881478353,
0.1739288334709101, 0.48205214710253746, 0.1392495483901865,
-0.7165035618232669, 0.21365045881478353, 0.47197748535599326,
0.3013279058058583, 0.007063152227780206, 0.03750941626348716,
0.14217476564729378, 0.8824613613633195, 1.2336424727315543,
0.23882536265997475, 0.1739288334709101, -0.06584286866797653,
0.03750941626348716, -0.006267326293252301, 0.1403125083239895,
0.15593677918642243, 0.5097615968227203, -0.5125026463842767,
0.9388513434061176, 0.539305636007754, -0.6171719735730511, 0.11652161853125897,
0.6489541350418935, -0.35246936701091575, -0.31286415949263024,
0.01676787057596669, 0.43917461663529705, 0.29901471878717495,
0.15531037475851142, 0.35850054405674997, 0.18895318112853787,
0.1561453519376068, 1.4336055326852764, 0.11702846860343367,
-0.03730218079867091, -0.0951233742546075, 0.16416750674699396,
0.43036262754428734, 0.4189693260848989, 0.2978513098080994,
-0.4590558517342529, 0.3615147888210593, 0.20178976953945785,
-0.23762814562284248, 0.6604492781114665, 0.10277126932002346,
0.12124663873908083, 0.27492643853835314, 0.25760650713546246,
0.20367941217364216, 0.0364878502015579, 0.35958085336728995,
0.2546099978103614, 0.11226883871012294, -0.14319774689485004,
0.2183233386375307, 0.2924883169246306, -0.38748320688418564, 0.276915373838934,
-0.2646526362320401, 0.8145212236194225, 1.714241500418035, 0.19067002648937828,
0.23868499340129132, 0.1765082010984752, 0.5433078099158543,
0.24100397583094352, 0.6059262257260156, 0.1586459594630166,
-0.04216241250041946, 0.4582619902229387, 0.47477420595440406,
0.342361798701168, 0.4737993403739297, -0.1843498045468246, 0.63762510662563,

0.12609280367793943, 0.28729812788278736, 0.29193933758502555,
0.11019976505409478, -0.20110909580011294, -0.39442678932556, 0.115024117243016,
0.28800521164332127, 0.28800521164332127, -0.00885003099830198,
0.14726101065916117, -0.031782969825087444, 0.15700139393720547,
0.39887410703097353, 0.18284327090769495, 0.20720357296527311,
-0.11001769167755243, 0.2607868853034868, 0.7118349272823096,
0.19960878014253078, 0.20023762798634812, -0.04363365598177213,
1.1695945946354032, 1.1111277899119831, 0.8618223602083633, 0.2685994658987826,
0.16244659681086948, -0.013154519078929713, 0.32327600007189417,
0.07308847059471918, 0.11019049049486394, 0.22886776975544235,
0.24033038702852044, -0.2692796706083479, 0.19548403379969703,
0.6728422624239265, 0.31483804302623036, 0.2618503531143067,
-0.12780357431589984, 0.09805782282824023, 0.1802321571622213,
0.5862739050395771, 0.29464059913654833, 0.24094355059316477,
0.4244764183637572, 0.2237843040814369, 0.042093799139288894,
0.4628954910859621, 0.1416642555340749, -0.16915099096835773,
0.2016619393935729, 0.36016635974441985, 0.09413182062500754,
0.3781333748678906, 0.26676817904658257, 0.19808682477148218,
0.21733818289506715, 0.035790379659039785, 0.2171819703630772,
0.1464144574677163, 0.17511452650180262, 0.3247686691298351,
0.12805124112556945, 0.1234820134451624, 0.3493541365485117,
0.18408127174311056, 0.3679401833358425, 0.2575925766843234, 0.4769284565157031,
0.3976550518590981, 0.2282331793539448, 0.1281017847107082, 0.16958161174599168,
0.3581198073247255, 0.09504447823183973, -0.3534876412624504,
0.2654919378901189, 0.2804062699176862, 0.31872205288136635,
0.17972980923583465, -0.05054523676397823, 0.1841674813415534,
0.29510839330585786, 0.1683516185426453, 0.39490503190797666,
0.9190889034820919, 0.1552152434059402, 0.3656402448121586, 0.29664334602748754,
0.17650229021611943, 0.5812670170872972, 0.343578552609798, 0.5742301083342458,
0.2069465788251167, 0.2896393154618618, 0.5203793182523301, 0.3614908350155227,
0.4699908979355395, 0.3275768668283222, 0.16477885841791645,
0.12168732527478802, 0.2965139128180393, 0.6264238630113408,
0.25513396684101125, 0.26164995545485564, 0.2358698979500746,
0.17358603970844286, 0.21532017519170563, 0.1831295881754969,
0.08296038531428768, -0.18580976119397172, 0.356274174269402,
0.2241189000387244, 0.4170232037146397, 0.8939767635138255, 0.34324835034476936,
0.2117885529994979, 0.14214586938243864, 0.22963323274912323,
0.04119700644876357, 0.1706660548288969, 1.3590176271142729,
0.18813020778971026, 0.2532801866562996, 0.7672477748738268, 0.343379425967451,
0.3437270542548817, 0.1685996974159433, 0.49447056797587596, 0.4284852747806454,
0.16711995588468215, 0.4709721369559841, 0.19739590097473828,
0.3649067207025391, 0.3098745216447973, 0.2196350348338764, 0.16824792040214356,
0.24476703272677147, 0.37330830187218894, 0.2244060667116141,
0.7082681562194038, 0.5552855133091843, 0.16451600566840932,
0.27385180738315823, 0.9300949597372863, 0.3973024559054642, 0.5103125456411304,
0.29941667095457997, 0.020631752395934025, 0.1956792428608066,
0.1532878042750184, 0.20720001372332678, 0.9775638468897404, 0.5748455586513986,
0.3988410632848609, 0.3943812057299776, 0.23887160862443893, 0.4833371034886711,

0.39880093985393156, 0.2799671117222743, 0.09431859439198055,
0.4014245350291708, 0.09896372136855544, 0.03494228981022281, 0.427436185186029,
0.42823786226155663, 0.21406886195140257, 0.37222921042348023,
0.09167786922510736, 0.40705632010375276, 0.5713901754422074,
0.1991567294346128, 0.2104098714218663, 0.19543321715764217, 0.5709757018714308,
0.11652008117482086, 0.42702085683125407, 0.20731568608330647,
0.22665646362363165, 0.2890994343993689, 0.20944502768075798, 0.400709661471108,
0.21888934188695366, 0.45097266398944885, 0.14395993834735213,
0.35212705116154047, 0.25760032107941266, 0.37551032288091124,
0.11832159392696623, 0.3132427697394125, 0.20323652163539632,
0.01285747416769302, 0.27877329647084875, 0.2907338989176383,
0.9641132022905505, 0.27403206297166827, -0.007409182559997528,
0.2539987580256059, 0.18268159155719566, 0.10584734003423654,
0.16353359026365266, 0.6781047121129542, 0.25796977434983526,
0.15471834687795646, 0.10275622614142202, 0.7527828161653474,
0.3850654838633981, 1.1414630663359728, 0.1747258554491594,
0.005095687040941317, 0.5098419973442542, 0.2591941106981112,
-0.2048762451425421, -0.10726533060333784, -0.25598761265885006,
0.5875992723710873, -0.002863048203628983, 0.2358610306149133,
-0.39062882534347776, 0.13445114565494096, 0.16416525696723586,
0.9285656986624847, 0.3470134945291991, 0.3816960443137617, 0.1544804421669558,
0.12383246296779161, 0.3716230136606885, 0.09568054140456007,
0.21145534710106492, 0.269873473322546, 0.7032106051090018, 0.889104590988304,
0.10100596875940021, 0.12496365083848174, 0.1534660809073914,
0.7709932948689712, 0.5329831923784581, 0.2111469140216237, 0.3323399737105158,
0.14355371052259516, -0.025556136905736913, 0.2505745528573956,
0.17872618372952315, 0.4480214128079672, 0.5922248902636985,
-0.0810041216445158, 0.4304552138573715, 0.19130471005916191,
0.4467175981976625, -0.018307160018494917, 0.9193561684064095,
0.8648352203509083, -0.07176167712381316, 0.196110381579216,
0.26557994215967057, 0.08791334552106052, -0.28191250877951846,
0.31411003368598805, 0.52719443710913, 0.1342353849854263, 0.17889381598597653,
0.42689425799262803, 0.1232510800937778, 0.4696220901719113, 0.3138806281530129,
0.21328996874274128, 0.5956405946256814, 0.30516606224614934,
0.002657535549531492, 0.10286596158591801, 1.1785829035808921,
0.5337533419030767, 0.16411966786617124, 0.24077339451867255,
0.26030440010862077, 0.3663102334736016, 0.025307812512662953,
0.044840733450389594, -0.027902636588580842, -0.08790818469922142,
0.3032797540172131, -0.18503272327008224, 0.2779493769618943,
0.17008888419429305, 0.1216372052213896, 0.182347582685476, 0.21272957254810515,
0.25826076777101603, 0.3146865818308366, 0.28440063524336784,
-0.1089082735819212, -0.2598470451384433, 0.20482447291638348,
0.5547244360394437, 0.19486413309464637, 0.37438962971573536,
-0.06266980076694396, -0.3511638808091911, 0.040841422647931926,
0.37145802408486805, 0.17896163138635146, 0.006513780683712009,
0.025528137343847304, 0.4395581636465671, 0.33755455440358606,
0.4002809632073378, 0.11659273634173481, 0.13616437537027706,
0.07816467883290636, 0.7696227392851946, 0.5068916722316822,

-0.4258576475489214, 0.16935473603722961, 0.10815297517371336,
0.05473040287032179, 0.12454800797664689, 0.13327370757193385,
0.4091564223037326, 0.2991762254282078, 0.1733075436153778, 1.308080770077605,
-0.009301368126793906, 0.4128893761938556, 0.33944406846127484,
0.12225074104158774, 0.11643160155434161, 0.13095988339699458,
0.2239851651135771, -0.2149706212520823, 0.11252627221707039,
0.2552742616033755, 0.18232386587175475, 0.24970927266154666,
0.7781190418874002, 0.20946570872859105, 0.46127420437660743,
0.4586446857845793, 0.0005382461163017038, -0.160245992595189,
0.48552333847795864, 0.275394414764982, 0.3726974329834084, 0.15453736573738064,
0.09066418906102441, 0.2582008519931764, 0.17942549192603138,
0.11709796331316473, 0.34352712733675905, -0.48292296667701595,
0.25922999828393284, 0.1582283219186833, 0.14881351502813708,
0.1164206044373422, 0.45446798009547623, 0.18226847773630717,
0.4071213013517459, 0.665052517355618, 0.4550395178207538, 0.34685386724606454,
1.4720748039904425, 0.4053254454324712, -0.4197485877067749,
0.18115855633558567, 0.2948497194009972, 0.031354079896236724,
-0.014951672907451691, 0.41211795946605895, 0.08037342031071562,
0.35920774522305227, 0.09861189112800943, 0.4122518838317239,
0.2606832481497402, 0.30093257270659335, 0.06913235918697176,
0.5686397998635858, 0.3780809118101353, 0.12825502720219145, 0.400985108274592,
0.24745076912395533, 0.22310142333410687, -0.19133685380522117,
0.015996868961363462, -0.3071051200817905, 0.03538313702656289,
0.2917626883142005, 0.2767408997980821, -0.07995393654818711,
0.23318930255871775, 0.5457770504060951, 0.22129792520802125,
0.36894007214321345, 0.3988486886664379, 1.82847425297854, -0.2877612221114466,
0.22494876553174617, 0.4944078575538014, 0.23147917610783403,
0.27501276567070454, 0.12014578035666934, 0.7983343392118485,
0.04222335601635987, 0.18694424215634795, 0.4418918234372035,
0.3581388275868253, 0.38881952192787206, 0.1322646820173049,
0.41739076620893706, 1.5856805772258844, 0.16160132027727234,
0.9046077624642652, 0.1427374501217748, -0.11403662495314382,
-0.2651172181862033, 0.15311494940295367, 0.24661948168400125,
0.25152591974592897, 0.22278304096739845, -0.07433775579592893,
0.4947560442021916, 0.20080505626945894, 0.17747882611561697,
0.2760945352362577, 0.34974808794501316, 0.1673709528128819,
0.10542512459281358, 0.6791443488481388, 0.37966978670096124,
0.21004110067817794, 0.18455133792078646, 0.3408892782821733,
0.17129267676036836, 0.2512678330083772, 0.31588648100372935,
0.6456945509655793, 0.153334636421658, 0.4661286627743985, 0.7718661827128077,
0.42330832285083664, 0.2849567506240873, 0.20375144399662137,
0.44997576026195946, 0.02307271035579276, -0.2725168374930868,
0.6194821416399414, 0.2892862675466414, 0.5994178406228312, 0.21758923498060878,
0.8170444566934394, 0.23144504806605484, 0.4474912292423999, 0.7011566580753716,
0.27205927296212373, 0.17387855398189786, 1.1277583186552402, 0.827244531425703,
-0.08219910442730811, 0.38630807477986523, 0.3940716741024779,
0.3153696359627576, 0.2032399739471067, 0.3921058059341157, 0.4091921571257324,
0.5631466991222027, 1.1748276211691702, 0.21708505371019796,

0.15869117304612404, 0.38273327995212586, 0.06201893403450687,
0.45971849763111755, -0.08514630063648716, 1.8968721557484407,
0.5216382336291764, 0.2633331437164786, -0.0864781764488196,
0.32224174198469757, 0.46786454267896915, 0.2385293090731321,
-0.11469562440598102, 0.5250923202199964, -0.10378256403877409,
0.24031026741286152, 0.24312823969760225, 0.36091436489803436,
0.7394074240623709, 0.21354792776108678, 0.23481035501899716,
0.0186635173594497, 0.13784301340185268, 0.3971355810864114,
0.15565296017599548, -0.0017532315788045417, 0.29662310185791313,
0.09865318408801939, 0.10019462145686311, -0.1910474729667011,
0.406062368938611, 0.1902621384165286, 0.10787350532914583, 0.15245017256957527,
0.15205701348345194, 0.14126093729034347, 0.20179539120401918,
0.42318998309239547, 0.03693048617061261, 0.24941366618429547,
0.13073741483409387, 0.1617643033545552, -0.1965548410598585,
0.2866787742682851, 0.5249580684728212, 0.010518408574626617,
0.5925938203413132, 0.20652443692473338, 0.6586977195674666,
0.08130408735008209, 0.24648220663081088, 0.1416740411202459,
0.031016765175329798, 0.39607646849215317, 0.29604502798708,
0.13075455662435487, 0.39934095389803465, 0.14699198817271245,
0.4083218946504673, 0.29351475583688624, 0.2645089177472382, 0.1525435125549085,
0.2851871700134454, 0.09584917587147675, 0.25290166215001736,
0.35419872565903343, 0.524903399796345, 0.2815152437711048, 0.11544869818162085,
0.20496063567493297, -9.575117293524348e-05, 0.23371721875646256,
0.7040471314289649, 0.2704470922050294, 0.32827398542082215, 0.5462904884333777,
0.11186670127957432, 0.16289779507289284, 0.2801013768262493,
0.29165307682958136, 0.16593488134497641, 0.10847043133947347,
0.1663538874779715, 0.2620100024128824, 0.12315300074035633, 0.2824843210799532,
0.21266166269045592, 0.5984300903218711, 0.1475920597148171,
0.12885062189009505, 0.09670556967355118, 0.2260570463430371,
0.20944809861904287, 0.15222231501970826, 0.1599815991296515,
0.1350688360144282, 0.1501526032425876, -0.034994996258750216,
0.0906802881063437, 0.15674442987144965, 0.11878825854432118,
0.11112422385167549, 0.023289003577469162, 0.2504374696274066,
0.10100064668596218, 1.1200348141382086, 0.20827734182693505,
0.10860232283777507, 0.289806811465071, 0.12191147996376002,
0.11002874601978324, -0.03577325450951531, 0.0626349388312571,
0.1687240168008482, 0.1452280758771861, 0.1255393005553547, 0.12299508111647339,
0.10961485548394757, 0.25973985136382804, 0.3814566854324869,
0.23036218790580418]
[0.31771380987644465, -0.40942830122975393, 0.21365045881478353,
0.30650630922700456, 0.5047765425497804, 0.48205214710253746,
0.36168976890930876, 0.21365045881478353, 0.47197748535599326,
0.3013279058058583, 0.8824613613633195, 0.23882536265997475, 0.389862534072901,
0.5097615968227203, 0.9388513434061176, 0.2549752648205657, 0.539305636007754,
0.30613368440006, 0.6489541350418935, 0.26068157869721575, -0.08034626764427906,
0.43917461663529705, 0.29901471878717495, 0.35850054405674997,
0.18895318112853787, 1.4336055326852764, 0.11702846860343367,
0.30990759382059924, 0.42418501277006637, 0.43036262754428734,

0.2978513098080994, 0.3043869356191349, 0.3615147888210593, 0.20178976953945785,
0.6604492781114665, 0.27492643853835314, 0.25760650713546246,
0.20367941217364216, 0.4138854118759055, 0.35958085336728995,
0.2546099978103614, 0.25119889433214493, 0.2183233386375307, 0.3815664172147645,
0.2924883169246306, -0.38748320688418564, 0.276915373838934,
0.24103645617792763, 0.8145212236194225, 1.714241500418035, 0.23868499340129132,
0.5433078099158543, 0.24100397583094352, 0.1586459594630166,
0.05133233776410916, 0.4582619902229387, 0.47477420595440406,
0.4737993403739297, 1.2012529056110575, 0.63762510662563, 0.28729812788278736,
0.29193933758502555, 0.28800521164332127, 0.28800521164332127,
0.39887410703097353, 0.18284327090769495, 0.2607868853034868,
0.7118349272823096, 0.19960878014253078, 0.375472502918016, 1.1111277899119831,
0.8618223602083633, 0.2685994658987826, 0.32327600007189417,
0.22886776975544235, 0.24033038702852044, 0.19548403379969703,
0.28517021940932663, 0.6728422624239265, 0.31483804302623036,
0.2618503531143067, 0.09805782282824023, 0.5862739050395771,
0.29464059913654833, 0.24094355059316477, 0.4244764183637572,
0.2237843040814369, 0.4628954910859621, 0.36016635974441985, 0.2922720951644761,
0.3781333748678906, 0.26676817904658257, 0.19808682477148218,
0.21733818289506715, 0.2893298395559195, 0.2171819703630772, 0.3247686691298351,
0.12805124112556945, 0.3493541365485117, 0.18408127174311056,
0.3679401833358425, 0.2575925766843234, 0.4769284565157031, 0.3976550518590981,
0.2282331793539448, 0.3581198073247255, 0.2654919378901189, 0.2804062699176862,
0.31872205288136635, -0.05054523676397823, 0.201731395711365,
0.1841674813415534, 0.29510839330585786, 0.39490503190797666,
0.9190889034820919, 0.3656402448121586, 0.29664334602748754,
0.17650229021611943, 0.5812670170872972, 0.343578552609798, 0.5742301083342458,
1.3949463394366093, 0.2069465788251167, 0.2896393154618618, 0.5203793182523301,
0.3614908350155227, 0.20405724678246168, 0.4699908979355395, 0.3275768668283222,
0.16477885841791645, 0.6264238630113408, 0.25513396684101125,
0.26164995545485564, 0.2358698979500746, 0.2145389875844129,
0.21532017519170563, 0.1831295881754969, -0.18580976119397172,
0.356274174269402, 0.2241189000387244, 0.4170232037146397, 0.8939767635138255,
0.34324835034476936, 0.2117885529994979, 0.22963323274912323,
1.3590176271142729, -0.021211771521697086, 0.18813020778971026,
0.2532801866562996, 0.7672477748738268, 0.343379425967451, 0.3437270542548817,
0.49447056797587596, 0.4284852747806454, 0.16711995588468215,
0.4709721369559841, 0.19739590097473828, 0.3649067207025391, 0.3098745216447973,
0.2196350348338764, 0.2687487337128552, 0.24476703272677147, 0.3790846578969924,
0.37330830187218894, 0.2244060667116141, 0.7082681562194038, 0.5552855133091843,
0.27385180738315823, 0.9300949597372863, 0.3973024559054642, 0.5103125456411304,
0.29941667095457997, 0.1956792428608066, 0.20720001372332678,
0.9775638468897404, 0.5748455586513986, 0.3988410632848609, 0.3943812057299776,
0.23887160862443893, 0.4833371034886711, -0.061852055410886134,
0.39880093985393156, 0.2799671117222743, 0.4014245350291708,
0.09896372136855544, 0.427436185186029, 0.42823786226155663,
0.19553518249271895, 0.21406886195140257, 0.154107859329002,
0.37222921042348023, 0.9033554983725932, 0.40705632010375276,

0.5713901754422074, 0.1991567294346128, 0.2104098714218663, 0.19543321715764217,
0.5709757018714308, 0.42702085683125407, 0.20731568608330647,
0.22665646362363165, 0.2890994343993689, 0.20944502768075798, 0.400709661471108,
0.21888934188695366, 0.45097266398944885, 0.14395993834735213,
0.35212705116154047, 0.2576003210794126б, 0.37551032288091124,
0.3132427697394125, 0.20323652163539632, 0.27877329647084875,
0.9641132022905505, 0.27403206297166827, 0.2539987580256059,
0.18268159155719566, 0.6781047121129542, 0.25796977434983526,
0.7527828161653474, 0.3850654838633981, 1.1414630663359728, 0.33440080241678155,
0.5098419973442542, 0.2591941106981112, 0.20685018061091687, 0.6953235117102973,
-0.2048762451425421, -0.10726533060333784, 0.5875992723710873,
0.8134892918563403, -0.002863048203628983, 0.7006033467987816,
0.2358610306149133, 0.9285656986624847, 0.3167591836146476, 0.3470134945291991,
0.3816960443137617, 0.3731287815009729, 0.21145534710106492, 0.269873473322546,
0.7032106051090018, 0.889104590988304, 0.7709932948689712, 0.2824409716218123,
0.5329831923784581, 0.2111469140216237, 0.3323399737105158, 0.2505745528573956,
0.6918942240999894, 0.4704739138889973, 0.4480214128079672, 0.5922248902636985,
0.2136040930727279, 0.4304552138573715, 0.19130471005916191, 0.1342006370247375,
0.4467175981976625, 0.9193561684064095, 0.8648352203509083, 0.6609119634199445,
0.26557994215967057, 0.31411003368598805, 0.52719443710913, 0.42689425799262803,
0.4696220901719113, 0.3138806281530129, 0.21328996874274128, 0.5956405946256814,
0.30516606224614934, 0.002657535549531492, 0.5337533419030767,
0.18890935694490693, 0.24077339451867255, 0.26030440010862077,
0.24157879752596442, 0.3663102334736016, 0.3202872483794082, 0.3032797540172131,
0.2779493769618943, 0.24812186617163098, 0.182347582685476, 0.21272957254810515,
0.25826076777101603, 0.3146865818308366, 0.28440063524336784,
-0.3366540909711182, 0.5547244360394437, 0.19486413309464637,
0.2245282713123063, 0.37438962971573536, 0.37145802408486805,
0.4395581636465671, 0.33755455440358606, 0.4002809632073378, 0.5709384834303833,
0.7696227392851946, 0.5068916722316822, -0.4258576475489214, 0.3085228168653482,
0.4091564223037326, 0.2991762254282078, 0.331907858756967, 1.308080770077605,
0.4128893761938556, 0.33944406846127484, 0.12225074104158774,
0.46326506544402524, 0.2606832655976654, 0.28935868558087213,
0.32778358880918534, 0.2239851651135771, 0.39874796116104805,
-0.2149706212520823, 0.2552742616033755, 0.18232386587175475,
0.24970927266154666, 0.7781190418874002, 0.20946570872859105,
0.46127420437660743, 0.4586446857845793, 0.48552333847795864, 0.275394414764982,
0.3726974329834084, 0.2582008519931764, 0.34352712733675905,
-0.48292296667701595, 0.031170195684332082, 0.2948662322797313,
0.25922999828393284, 0.45446798009547623, 0.18226847773630717,
0.4071213013517459, 0.7465172685655237, 0.40028955337196503, 0.665052517355618,
0.4550395178207538, 0.34685386724606454, 0.24762702062427383,
1.4720748039904425, 0.4053254454324712, -0.4197485877067749,
0.18115855633558567, 0.41211795946605895, 0.35920774522305227,
0.4122518838317239, 0.2606832481497402, 0.30093257270659335,
0.32243511710773787, 0.5686397998635858, 0.3780809118101353, 0.400985108274592,
0.22310142333410687, 0.015996868961363462, 0.6546175351949252,
0.25067414159613954, 0.2767408997980821, 0.23318930255871775,

-0.16307353246842982, 0.5457770504060951, 0.22129792520802125,
0.36894007214321345, 0.3988486886664379, 1.82847425297854, 0.22494876553174617,
0.4944078575538014, 0.23147917610783403, 0.27501276567070454,
0.32833413873224515, 0.7983343392118485, 0.022890416793279623,
0.18694424215634795, 0.4418918234372035, 0.35813882758682536,
0.38881952192787206, 0.41739076620893706, 1.5856805772258844,
0.9340084351605198, 0.16160132027727234, 0.9046077624642652, 0.2987513621830152,
0.2637733206316914, 0.24661948168400125, 0.25152591974592897,
0.4947560442021916, 0.17747882611561697, 0.44621598336858675,
0.2760945352362577, 0.34974808794501316, -0.05951781245103033,
0.28677981509548145, 0.6791443488481388, 0.37966978670096124,
0.21004110067817794, 0.18455133792078646, 0.040035974078756704,
0.2512678330083772, 0.31588648100372935, 0.6456945509655793, 0.4661286627743985,
0.7718661827128077, 0.2849567506240873, 0.20375144399662137,
0.44997576026195946, 0.6194821416399414, 0.2892862675466414, 0.5994178406228312,
0.21758923498060878, 0.8170444566934394, 0.31759032471361026,
0.23144504806605484, 0.31749077719593527, 0.25615059148639296,
0.4474912292423999, 0.41803003166924346, 0.7011566580753716,
0.27205927296212373, 1.1277583186552402, 0.827244531425703, 0.38630807477986523,
0.3940716741024779, 0.2940173102867533, 0.3153696359627576, 0.4091921571257324,
0.5631466991222027, 1.1748276211691702, 0.22895482196716158,
0.38273327995212586, 0.21517695188873848, 0.3402844856099957,
0.06209759815859521, 0.45971849763111755, 1.8968721557484407,
0.5216382336291764, 0.2633331437164786, 0.8949246945340479, 0.3222417419846975,
0.2385293090731321, 0.36032884518573594, 0.5250923202199964,
0.24031026741286152, 0.24312823969760225, 0.36091436489803436,
0.21354792776108678, 0.23481035501899716, 0.29662310185791313,
0.1902621384165286, 2.6162339449409973, 0.14126093729034347,
0.42318998309239547, 0.24941366618429547, 0.2866787742682851,
0.21871656491158187, 0.5925938203413132, 0.6586977195674666,
0.30551172347664624, 0.24648220663081088, 0.6192069333223414,
0.39607646849215317, 0.29604502798708, 0.18901396177102106, 0.39934095389803465,
0.4083218946504673, 0.29351475583688624, 0.2645089177472382, 0.2704470922050294,
0.32827398542082215, 0.5462904884333777, 0.2801013768262493,
0.29165307682958136, 0.2620100024128824, 0.21266166269045592,
0.5984300903218711, 0.2260570463430371, 1.1200348141382086, 0.20827734182693505,
0.289806811465071]
[0.45695964150255436, 0.4544875458576641]
[0.12336954468165501, 0.41771443791629204, -0.1247667075830567,
-0.14080759374546478, 0.16311445415545373, 0.8150276399858329,
0.16240996407747563, 0.7892445138092523, 0.18633403050147473,
0.24272391002173294, 0.20425736468906217, -0.2713024585348293,
0.12585725337419465, 0.48227949823239485, 0.79725447132604, 0.6273696277554213,
0.79725447132604, 0.20425736468906217, 0.5156640591743662, 0.048940639151627816,
2.378569559851778, 0.32482926730767, 0.12585725337419465, 0.18474510384580886,
0.4135613796331323, -0.00040450471569707813, 0.9390751669135264,
0.1460564944338035, 1.0527600380890123, 0.5647236899839554, 0.3767865852293039,
0.7190125489263661, 0.30411696812510175, 0.1470503457449368, 0.6706897269403782,

0.032635088253473075, 0.012612118110135373, 0.23629016339214884,
0.45560195129258235, 0.22945548522655057, 0.310704579812342, 0.6797822646402439,
0.5759196141154208, -0.4945857806756885, 0.4113462774005256,
0.29280779710308585, 0.4011030596270986, 0.4418185171857315, 1.9395976423167984,
2.2377373303588164, 0.34912707977835344, 0.12529655786470248,
0.20152015369855183, 0.32013609442942875, 1.008857403103812,
-0.1838825469754636, 0.6070657806648447, 0.6617765947992095,
0.24952689566409347, -0.27596929355747746, -0.11812543818885184,
0.15061790146322931, 0.23034871108683, 0.23065427124632412, 0.9350946254204374,
-0.2907949450977844, 0.4767734855032222, 0.24031614709152574,
0.24054953487907169, 0.5687816718244835, 0.1973646272209565, 0.8017996275865574,
0.26866511993946723, 0.4738011523226926, 0.34940076460779235,
0.9912080478939088, 0.3333081440147987, 1.104365402878712, 0.09315955402145562,
-0.2193408538501497, 0.2502920126276384, -0.06533267047808436,
1.3103351792061626, 0.3862347825499357, 0.9304465430730731, 0.1940768359691836,
0.3968639818165869, -0.1661863287490087, 0.5122742536725231, 0.4318641269455023,
0.13763413519557086, 0.2581384566079587 6, 1.0628696330619467,
0.6399212194042021, 0.7906736086635671, 0.3397750422776676, 0.5887838675141398,
0.3158774108173754, 0.807930628226298, 0.745843431578063, 0.24028208045543886,
0.2742438394138297, -0.4374446551750858, 0.24655419003935586,
0.20486527060582496, 0.08532648816567598, 0.3388190566777761,
0.055843385789606424, 0.23822968166088254, 0.6386151421428904,
0.5105379746835443, 0.25237820335414457, 0.3862347825499357, 1.0698954015547302,
0.3032897028564022, 0.2487503867170017, 0.20357843762331868,
0.13459831935260727, 0.2974547909397887, 0.2768524913910771,
-0.09111714711991813, 1.8948965633542398, -0.06162386284591998,
0.32602369762167654, 0.18450218263367726, 0.4287263177238012,
0.18167501661152638, 0.20254223079062583, 0.36767268335376035,
0.19487340972513056, 0.20080206071725037, 0.18450218263367726,
0.1872909233870292, 0.14688870321475564, 0.9960439416928358,
0.18775701071654477, 0.3092998499999893, 0.11421644734704219,
0.9321607210985144, -0.03582249575311044, 0.13948718236562882,
0.1520437162468207, 0.3653134215655017, 0.19309971152264005, 0.6369999721968473,
0.21401801010525795, 0.148933772466196, -0.11066658350344423,
0.1515567505302841, 0.39731593362352, 0.4908793948125364, 0.4135187510502257,
0.3832503221854329, 0.24210059451978932, -0.09712106511124319,
0.33210724637160305, 0.4204624845654864, 0.5449203656883909, 1.03746124648345,
0.3499781003940507, 1.0946418979862074, 0.18110839912595808, 0.7966485515091025,
0.5240410132418584, 1.1966043810460403, 0.5404398151857832, 0.3383078998139677 5,
0.06320244693270535, 0.08594009034565012, 0.6058063336897295,
0.18560892840753626, 0.31467694593896733, 0.5319783055231226,
0.2770290272746601, 0.24024397082610952, 0.18153257177701332,
0.11194782769779948, 0.2548652549966804, 0.4441921295930378,
0.14382124415089273, 0.3820074695710009, 0.2824341086854203,
0.23217541554744722, 0.2957460900480315, 0.7237830637355812, 0.2764580763731055,
0.14299267524788598, 0.2146839885528135, -0.08747437960876243,
0.17493698429073046, 0.0694790805754189, 0.12237290688015905, 0.534913878074023,
0.2302377860417527, 0.2942990275133873, 0.278138250755447, 0.21750259876882616,

0.36841593104213355, 0.03589516728879573, 0.13921416579185122,
0.1335867407128234, 0.07563325538599466, 0.14601702910187758,
0.2512591544423763, 0.28145237360434855, 0.2086746873202963,
0.10872672648385023, -0.057965282086528444, 0.22115713773112033,
0.2867365183316885, 0.3492712217294791, 0.3354423890730324, 0.11858530177063498,
0.051948374446947804, 0.0885722851094873, 0.047742061329279224,
0.38291039425741824, 0.1850369791178517, 0.17186612214991276,
0.1556688577120678, 0.48107739479869227, 0.22741252981703744,
0.07393097808346805, 0.17430255721028273, 0.38081604295940613,
-0.3298348191042398, 1.0448583325305332, -0.000570630980500884,
0.06850804262213382, 0.2143630757285146, 0.09353499766489268,
0.5984270586882685, 0.2963151010357009, 0.08160905448060322, 0.1485539404094733,
0.22683522420353355, 0.7028756317957574, 0.20733973766163774,
0.10881274191980221, 0.637834055089671, -0.1523548787847079, 0.5199181637887252,
0.18592650198593233, 0.15689607155739624, 0.6785953508613298,
0.24945515076500224, 0.18456709045273637, 0.1545593251140865,
0.23809478376370324, 0.10559242981100418, 0.1438610437082276,
0.1900031816355654, 0.1905376467439103, 0.3661603432206036, 0.12192450758960498,
0.2204137983879901, 0.40365298835956986, 0.14287288605185997,
0.38455428387869145, 0.02453033531863632, 0.44086083009477345,
0.6004992301256464, 0.1803354961940473, 0.3718701304471551, 0.13297619919811776,
0.11734584906682523, 0.22311301901769978, 0.18087945003669534,
0.3807113449413337, 0.26990600423036176, 0.5610328885827705,
0.45781184995934376, 0.11343334484483113, 0.7578878118448835,
0.19843511283212636, 0.3116350605418492, 0.13436157061338133,
0.28699201243803557, -0.0974639438903261, 0.16997259318257643,
0.2908311154705766, -0.17940620548028144, -0.035202788019596955,
0.1399159810550525, 0.4447972584489513, 0.06929017165410316, 2.046204283374478,
0.27880915746428014, 0.12430693854257482, 0.5259529859872641,
0.3847219436838223, -0.003689787771638707, -0.1834281442984356,
0.09046904121827684, 0.1743435465043127, 0.16505504509606708,
0.2869626818995232, 0.8120221697530415, 0.1215429261002442, 0.4146126515497056,
-0.23076956854617495, 0.43165958672793864, 0.2192086627591798,
0.15798493066710653, 0.21886069121795684, -0.118727123865029,
0.2862262723624687, 0.27336692801048146, 0.5862547557247867, 0.6984227271066139,
0.10016644726324599, 1.3746336947879663, 0.26875915651247334,
0.0017026426536432338, 0.05611272988325877, 0.49206473209561935,
0.6877103733281497, 0.006056719583794038, 0.18542643706391948,
1.141220019744178, 0.32614845953810295, 0.25744159545889994, 0.8110923379399644,
0.6673232903179567, 0.3271298844454758, 0.01993584112576975, 0.7095186752845621,
0.2976131898214123, -0.022084767600937596, 0.27761496848911005,
0.16975558697256996, 0.595228703404214, 0.45695964150255436,
-0.09270452477602197, 0.1873633165854404, 0.9292207325622117,
0.24924667529519046, 0.13854761349955344, -0.15694384091258876,
1.047163923530435, 0.6168491261027085, 0.3425212904557993, 0.1729141691266254,
0.5635938643411562, 0.619377572508134, -0.0901563090456129, -0.4131021626591834,
1.004636078733798, 0.28088115008957676, -0.09406501367506176,
0.2427126412595411, 0.13331318726744448, 0.6653468122995245, 0.6427198222122978,

0.4623062652178967, 0.09710284820444526, 0.3827557761647502,
0.15771449161216808, 0.014384990498528243, 0.2045364152612585,
0.4461786154934191, 0.1430588587435702, -0.017499205288885683,
-0.004140933400028471, 0.43319116762354776, 0.27053597131716006,
0.17071967763892756, 0.5602784235085682, -0.027149159513506382,
-0.01685676872546188, -0.09264856824799059, 0.5285878249081325,
0.031884200814897505, 0.18538736884699836, 0.6926418186028065,
0.01021370513448256, 0.6372538163372979, -0.10092799528375702,
0.0885491754576513, -0.06948766560210111, 0.014119588177137736,
-0.05636543751527281, 0.4544875458576641, 0.38006634436176306,
0.8445249383936357, 0.12797009206071178, -0.13917059934232548,
0.1259907087254509, 0.3248137550225666, 0.2829569329773607, 0.3566485924713791,
0.41468426561163424, 0.30049139011923715, -0.012622490486268236,
0.07029718316823162, 0.12862088180182982, 0.3609504429396407,
0.11268180015025753, -0.011585602860236513, 0.41676990753468707,
0.5690277720051832, -0.019716203932499128, 0.32449367829209036,
0.756141211920514‌3, 0.5147720886495148, 0.11617925736365857, 0.4559651520173852,
0.2965916765155074, 0.40575396661293806, 1.4051912047097428, 0.3634579215850656,
0.19639862865302735, 0.16623889057576635, 0.3233602529119926,
-0.20170788967540929, 0.1781382810335021, 0.9541911675297604,
0.4626816023223965, -0.001834216573111244, 0.20248956738998966,
-0.14950089853702442, 0.5155076968378348, -0.16779210202225392,
0.16424139347080519, 0.21948607421414532, -0.11616053266346334,
0.1738143207917547, 0.1786370885823968, 0.06774171531136983,
0.28751519161493766, 0.1753714053367046, -0.1968133866389558,
0.21673282877702327, 0.2799709884460998, -0.08112881263199412,
0.24882500205227362, 0.25063847337306605, 0.7456248618452805, 0.390881443919339,
0.2543915964121285, 0.12864628949894916, 0.27090736526385956,
0.48664183611748635, 0.3869563075964397, 0.26467505642796335,
-0.01946796399799416, 0.08869376443237922, 0.2057711819507302,
0.5276696367608322, 0.3471977906753513, 0.1658693615937217, 0.3439447199112902,
0.26658855451406016, -0.005034210224239669, 0.8356366329109087,
0.048798055414204476, 0.455357974622333, -0.0026282498446714574,
0.08731856033879298, 0.11128374346950548, -0.07647192709832591,
0.13477420447668434, 0.10900651804180123, -0.22652989005437715,
-0.019385049332715423, -0.09876702510961353, 0.4105489255091845,
0.6610349842842368, -0.15735105911999828, 0.6000052250990804,
0.011372826078473766, 2.2687052989084986, 0.4163221005219493,
-0.07257422382832986, 0.34097546370085935, 0.2027229721508345,
-0.04400510383725299, 0.22673179120681056, 0.41541854184819904,
-0.08894366723388561, 0.30326900956902686, 0.24537359312382392,
0.246679183053163, 0.33832414313247866, 0.13818394241175722, 0.540185917058124,
0.35414673389643314, -0.045300870353927734, 0.10104295230287044,
0.25323173336536264, 0.3617636908644355, -0.07960169013220726,
0.07474878675685369, -0.11764615117808924, 0.26791283923697495,
0.573825949710145, 0.4152184696457512, 0.11328872851375218, 0.0726062290056879,
0.16155018491086406, 0.25526754408923413, 0.24679202821165644,
-0.09646075794976296, 0.045370889752355335, 0.21574098235832256,

0.4317277000062125, -0.17738065735835343, -0.11056911266789703,
0.8107061361155943, 0.11257703847550692, -0.08569102602115883,
0.33913746673026646, 0.22647598337253505, 0.27116138798600653,
0.10022652125595462, 0.09003213205204527, 0.31012912309237495,
0.6269265394205067, 1.000971856241393, 0.23740330076706673, 0.3766140422193021,
0.2229942692317272, 0.03515495599116045, -0.326473210184659, 0.64265966591548,
0.09635766317845931, 0.029626018500831866, 0.3497470494844308,
0.5957323647714481, 0.5594472820532456, 0.23242132706750276,
0.49201975097962924, 0.3370947001968221, -0.15458537117784116, 0.14829582901737,
0.35811614908303774, 0.2466142982911238, -0.4048092556042382,
0.3217109739047013, -0.22409511671349572, -0.0360552895252391,
0.11164905187202238, 0.035597385707520836, -0.005443113678390808,
0.6145196103264015, 0.3343207083098999, -0.0590015230636393,
0.40900187319746145, -0.08224292966613238, 0.11875161130921964,
0.3370147585120467, 0.6200587500492668, 0.18209891835062944,
0.12555512771000896, 0.4187334899205163, 0.19090732402767988,
0.3492384434682164, 0.3514872831182041, 0.15140999003787545, 0.8494415970561705,
0.29932227489562263, 0.1604722205788654, 0.34380800152899116,
0.1343659865380697, -0.1556449502715334, 0.2569461641750552, 0.2451043227389502,
0.2331581636989233, 0.09462563660890984, 0.02040349875030053,
0.13855074643051077, -0.055434368305007156, 0.3213466738155986,
0.1398559112736645, 0.5395769077278849, 0.41344047503309217, 0.1510859584131629,
0.3115520867262661, 0.09599887295221916, 0.1167201131177357,
0.31560369815340167, 0.4297655294993236, -0.49483017467833745,
0.4429371821395292, 0.4182730623924566, 0.13086651593266319,
-0.04734250031856498, 0.06657817144044592, 0.0349942421410923,
0.179560963176751, 0.01505515573528447, 0.48388537532603626, 0.1436651018593473,
0.011648827511014268, 0.17879982431230554, 0.19927551568308072,
0.13164450123513083, 0.10743358433400546, 0.18346280717791902,
0.1521424450060223, 0.15948362725379628, 0.1501906975026131,
-0.11026640149328866, 0.3887284604031412, 0.5171839701476927,
0.44339714209388864, -0.05805021653882817, 0.0517341600032833,
0.2556351637665836, -0.34143551388177573, -0.026361477257019882,
0.1011419245161974, 0.6323115867287523, 0.19779399806759587, 0.4670162710118126,
0.2719932266486552, 0.3631972700235311, 0.26644159922866234,
0.09124689574362377, 0.17560665207658507, 0.31850571316098725,
0.21503881189655671, 0.3062314952542597, 0.10144033843708655,
0.24333119498160993, 0.17165698638437185, 0.12015638972657064,
0.3508590958074804, 0.139740637523747, 0.14861227812738273, 0.09910939347410293,
0.199197110622933, 0.0723773724187828, 0.12215913253813479, 0.189477654025645,
0.0727898911989191, 0.06639242010895526, 0.05842820003612088,
-0.355004990445983]
[0.1923923243268431, 0.41771443791629204, -0.14080759374546478,
0.7892445138092523, 0.18633403050147473, 0.24272391002173294,
0.34235729928003505, 0.20425736468906217, 0.48227949823239485, 0.79725447132604,
0.6273696277554213, 0.79725447132604, 0.20425736468906217, 0.3000575654971518,
0.5156640591743662, 2.378569559851778, 0.32482926730767, 0.18474510384580886,
0.4135613796331323, 0.9390751669135264, 1.0527600380890123, 0.32360070761440185,

0.7190125489263661, 0.6706897269403782, 0.032635088253473075,
0.23629016339214884, 0.22945548522655057, 0.310704579812342, 0.6797822646402439,
0.5759196141154208, -0.4945857806756885, 0.4113462774005256,
0.38871294794543415, 0.29280779710308585, 0.4011030596270986,
0.2884568808380758, 0.4157714703797633, 2.2377373303588164, 0.34912707977835344,
0.20152015369855183, 0.4158716472497382, 0.3154263717418153,
0.32013609442942875, 0.3923279356853723, 1.008857403103812, 0.6070657806648447,
0.6617765947992095, 0.24952689566409347, 0.41427877157445553,
0.20083634457183597, 0.23034871108683, 0.23065427124632412, 0.9350946254204374,
0.4767734855032222, 0.4915679309673831, 0.216513454000127, 0.24031614709152574,
0.24054953487907169, 0.5687816718244835, 0.1973646272209565, 0.8017996275865574,
0.26866511993946723, 0.4738011523226926, 0.5369516235227617,
0.34940076460779235, 0.2523268632440967, 0.3333081440147987, 1.104365402878712,
0.2502920126276384, -0.06533267047808436, 1.3103351792061626,
0.3862347825499357, 0.9304465430730731, 0.1940768359691836, 0.3968639818165869,
0.5122742536725231, 1.0628696330619467, 0.6399212194042021, 0.7906736086635671,
0.3397750422776676, 0.5887838675141398, 0.3158774108173754, 0.807930628226298,
0.745843431578063, 0.2742438394138297, 0.24655419003935586, 0.20486527060582496,
0.3388190566777761, 0.23822968166088254, 0.6386151421428904,
0.27504269460373076, 0.5105379746835443, 0.25237820335414457,
0.3862347825499357, 1.0698954015547302, 0.3032897028564022, 0.3163002897381313,
0.2487503867170017, 0.20357843762331868, 0.2974547909397887,
0.31803800155727413, 0.8448919293407505, 0.19062515749137923,
0.2768524913910771, 1.8948965633542398, 0.32602369762167654,
0.18450218263367726, 0.4287263177238012, 0.33498626789813224,
0.20254223079062583, 0.36767268335376035, 0.19487340972513056,
0.18450218263367726, 0.1872909233870292, 0.14688870321475564,
0.9960439416928358, 0.18775701071654477, 0.31424062974467437,
0.9321607210985144, 0.3653134215655017, 0.19309971152264005, 0.6369999721968473,
0.39731593362352, 0.4908793948125364, 0.4135187510502257, 0.24210059451978932,
0.4204624845654864, 0.5449203656883909, 1.03746124648345, 1.0946418979862074,
0.18110839912595808, 0.7966485515091025, 0.5240410132418584, 1.1966043810460403,
0.5404398151857832, 0.33830789981396775, 0.08594009034565012,
0.6058063336897295, 0.18560892840753626, 0.31467694593896733,
0.5319783055231226, 0.2770290272746601, 0.24024397082610952,
0.18153257177701332, 0.2548652549966804, 0.4441921295930378, 0.3820074695710009,
0.2824341086854203, 0.23217541554744722, 0.2957460900480315, 0.7237830637355812,
0.2764580763731055, 0.2146839885528135, 0.2842076689671584, 0.17493698429073046,
0.534913878074023, 0.2302377860417527, 0.2942990275133873, 0.278138250755447,
0.21750259876882616, 0.36841593104213355, 0.03589516728879573,
0.14601702910187758, 0.2512591544423763, 0.28145237360434855,
0.2086746873202963, 0.22115713773112033, 0.2867365183316885, 0.3492712217294791,
0.3354423890730324, 0.38291039425741824, 0.1850369791178517,
0.48107739479869227, 0.22741252981703744, 0.07393097808346805,
0.38081604295940613, 0.517023419328839, 1.0448583325305332, 0.2143630757285146,
0.09353499766489268, 0.5984270586882685, 0.2963151010357009,
0.22683522420353355, 0.7028756317957574, 0.20733973766163774, 0.637834055089671,
0.5199181637887252, 0.6785953508613298, 0.24945515076500224,

0.18456709045273637, -0.05495114529675135, 0.2996902752085504,
0.23809478376370324, 0.1900031816355654, 0.1905376467439103, 0.3661603432206036,
0.2204137983879901, 0.40365298835956986, 0.38455428387869145,
0.44086083009477345, 0.6004992301256464, 0.3718701304471551,
0.22311301901769978, 0.39277024079484374, 0.26990600423036176,
0.5610328885827705, 0.45781184995934376, 0.7578878118448835,
0.19843511283212636, 0.28699201243803557, 0.24024359075036308,
0.4447972584489513, 2.046204283374478, 0.5259529859872641, 0.3847219436838223,
0.8120221697530415, 0.4146126515497056, 0.43165958672793864, 0.2192086627591798,
0.21886069121795684, 0.2862262723624687, 0.2034864332709392,
0.27336692801048146, 0.5862547557247867, 0.6984227271066139, 1.3746336947879663,
0.42104713378770386, 0.3477225218434848, 0.26875915651247334,
0.28759792710924775, 0.05611272988325877, 0.35626952873141343,
0.49206473209561935, 0.6877103733281497, 0.18542643706391948, 1.141220019744178,
0.32614845953810295, 0.25744159545889994, 0.8110923379399644,
0.6673232903179567, 0.3271298844454758, 0.01993584112576975, 0.7095186752845621,
0.2976131898214123, 0.6906254510907274, 0.6572299420835408, 0.2170538598726784,
0.27761496848911005, 0.40854223936083967, 0.16975558697256996,
0.45695964150255436, 0.1873633165854404, 0.24924667529519046,
0.13854761349955344, 1.047163923530435, 0.3425212904557993, 0.20050316994633344,
0.5635938643411562, 0.619377572508134, 0.4894844058934843, 0.28088115008957676,
0.21986814993812698, 0.2427126412595411, -0.016793451449685012,
0.6653468122995245, 0.4732553081714621, 0.6427198222122978, 0.4623062652178967,
0.3827557761647502, 1.7641851488512617, 0.014384990498528243,
0.18724833820158954, 0.2045364152612585, 0.4461786154934191,
-0.017499205288885683, 0.43319116762354776, 0.27053597131716006,
0.5602784235085682, 0.5285878249081325, 0.18538736884699836, 0.6926418186028065,
0.6372538163372979, 0.4544875458576641, 0.38006634436176306, 0.8445249383936357,
0.1259907087254509, 0.3248137550225666, 0.2829569329773607, 0.3566485924713791,
0.41468426561163424, 0.30049139011923715, 0.40458691156090154,
0.26110564508854434, 0.18411600235682413, 0.3609504429396407,
0.41676990753468707, 0.5690277720051832, 0.32449367829209036,
0.7561412119205143, 0.5147720886495148, 0.4559651520173852, 0.27287637695894457,
0.2965916765155074, 1.4051912047097428, 0.3634579215850656, 0.19639862865302735,
0.3233602529119926, 0.9541911675297604, 0.4626816023223965, 0.20248956738998966,
0.5155076968378348, 0.21948607421414532, 0.4848494767061658,
0.19657872333331447, 0.06774171531136983, 0.47768639438758487,
0.28751519161493766, 0.21673282877702327, 0.2799709884460998,
0.24882500205227362, 0.25063847337306605, 0.8158309240615401,
0.7456248618452805, 0.390881443919339, 0.2543915964121285, 0.27090736526385956,
0.48664183611748635, 0.3869563075964397, 0.26467505642796335,
0.2994585458655045, 0.2057711819507302, 0.5276696367608322, 0.419063905071031,
0.3782951437378546, 0.3439447199112902, 0.26658855451406016, 0.8356366329109087,
0.23561241326361407, 0.455357974622333, -0.0026282498446714574,
0.23806967425318534, 0.2316863636471427, 0.4105489255091845, 0.6610349842842368,
0.6000052250990804, 2.2687052989084986, 0.34097546370085935, 0.2027229721508345,
0.22673179120681056, 0.41541854184819904, 0.30326900956902686,
0.3706453345142889, 0.24537359312382392, 0.246679183053163, 0.33832414313247866,

0.4310595052458275, 0.540185917058124, 0.35414673389643314, 0.45694909525372296,
0.3317496006608919, 0.25323173336536264, 0.3617636908644355,
0.26791283923697495, 0.573825949710145, 0.4152184696457512, 0.43767530994563725,
0.6617983162535463, 0.34791333962606585, 0.25526754408923413,
0.21574098235832256, 0.4317277000062125, 0.33913746673026646,
0.22647598337253505, 0.27116138798600653, 0.31012912309237495,
0.6269265394205067, 1.000971856241393, 0.23740330076706673, 0.4544100431868222,
0.26845721189013283, 0.3766140422193021, 0.2229942692317272,
0.23248012430114431, 0.64265966591548, 0.3497470494844308, 0.5957323647714481,
0.23242132706750276, 0.49201975097962924, 0.5569109252109246,
0.3370947001968221, 0.35811614908303774, 0.2466142982911238, 0.7397855809547796,
0.6145196103264015, 0.8068003134926898, 0.3343207083098999, 0.3370147585120467,
0.0014555181535891007, 0.6200587500492668, 0.18209891835062944,
0.4187334899205163, 0.19090732402767988, 0.20442873596343428,
0.3326950212824481, 0.3514872831182041, 0.8494415970561705, 0.29932227489562263,
0.34380800152899116, 0.7790556324495904, 0.5042616328525417, 0.2569461641750552,
0.2451043227389502, 0.2331581636989233, 0.3213466738155986, 0.5395769077278849,
0.18508727661981306, 0.31560369815340167, 0.4297655294993236,
0.2134755831709546, 0.4429371821395292, 0.4182730623924566, 0.48388537532603626,
0.30577946822038005, 0.19927551568308072, 0.3887284604031412,
0.44339714209388864, 0.2556351637665836, 0.6323115867287523,
0.19779399806759587, 0.4670162710118126, 0.3631972700235311,
0.26644159922866234, 0.31850571316098725, 0.199197110622933, 0.189477654025645,
0.1837003127870874]
[0.07089021980832316, 0.7377504697522319, -0.3571654026363833,
-0.1700055453484541, -0.4810101563297344, 0.3846865683456414,
-0.15984754287892616, -0.15689801609599077]
[1.0258697862730768, 0.13502851320065473, -0.23926184936438838,
-0.45446725286615924, -0.11653499298165544, -0.7382980678191303,
0.5841373766515076, 0.1486593292720599, -0.3928823076191278,
0.08423192545028285, 0.09144776356164254, -0.3825243222992674,
0.15080434043618354, -0.24286237997912244, 0.10305307176369378,
-0.6923944884560763, 0.20313267914608635, 0.28958670283749305,
-0.925602488073147, 0.19337872120805522, 0.13510104984295232,
0.0897288873140264, 0.2300547990627536, 0.09752953229344596,
0.22786514641794103, 0.339438327312814, 0.07413235016155702, 0.4652380973301887,
0.869368498850413, 0.33563580442634533, 0.4034503886863362, 0.3618292323838321,
0.15109158925398902, 0.4131149138286468, 0.24813650976178775,
0.4785533198893012, 0.1891972122983803, 0.1351509656427837, 0.4181964369449573,
0.640932309690594, 0.22150231091079883, 0.7456753057084354, 0.2181309321608525,
0.5497706027204458, 0.35655770100930156, 0.44268554275957067,
0.3791837628295653, 0.3408042234402626, 0.41663095543382855,
0.14203431914428688, 0.4443155468932214, 1.5796678403303832,
0.33283598892189004, -0.05070506667784388, 0.29314879695600504,
0.3153227480428732, 0.25174162746492074, 0.11108360107888013,
0.13483205196541023, 0.32455620144467734, 0.12178078378158985,
-0.4216898979935534, 0.05833911672801729, 0.239007578900857,
0.24876011000910328, -0.20206271836678769, 0.16770713051168193,

-0.4254536834002644, 0.05554077813573939, -0.5670754944260326,
-0.5015502429584382, -0.57275383317942, -0.3552122475082896,
0.25194070324799844, -0.07065502830724371, 0.5162465426716395,
-0.003554072027819124, -0.40513609926240574, 0.7377504697522319,
0.11500457634327106, 0.2533334972344091, -0.0863606223510222,
0.013550673014220053, 0.4788742051222833, 0.2194787128877741,
-0.4268358389361234, 0.32371824621127615, -0.2730326375508356,
0.0227271514651065, -0.23884022179797101, -0.7645244777675803,
0.15328384771789955, 0.21314781613727418, 0.029617463147026856,
-0.008769771939104644, 0.2203939015586344, 0.13337293205818046,
-0.4785593558127078, -0.4358004540423299, 0.24662273684501845,
-0.05331777162638201, 0.17990673440514346, 0.7255131453347734,
-0.3037397317235045, -0.4460930615455974, 0.0396169582370862,
-0.2967512570397666, 0.3147487836351023, -0.4287743041431911,
-0.1700055453484541, -0.11397655233319391, -0.045224656218697,
0.09711105677932325, -0.24966542080635595, 0.1307739101307324,
-0.4810101563297344, -0.20974044771410782, 0.08302891542330443,
-0.32728405331637195, -0.0632186151654468, 0.03896161352445109,
-0.297038244382659, -0.43943107753121935, -0.054521628131888565,
-0.40389700198374545, -0.12544580582662598, -0.630329735541189,
0.4120601219671131, 0.24786578724907143, -0.5350501496119633,
-0.5354878548707638, -0.14217452192538585, -0.15787990921594452,
0.20194570994523242, 0.13464645612729226, 0.16818428253466952,
0.3846865683456414, -0.40793259704711315, 0.21448676737694422,
0.17833659326725598, -0.21946072645004108, 0.1050647613610607,
0.09595365518677706, -0.03837541237702606, 0.3614982538480341,
0.6102993088458594, -0.15984754287892616, -0.351809179541463,
-0.2921916764626442, 0.1928045855560474, -0.09693152464839248,
-0.4782804365915596, -0.07244561271328032, 0.13544928474238946,
-0.15689801609599077, 0.03397285599259017, -0.4689403325254266,
0.09421716036684813, -0.17068666781518774, -0.18450952544847463,
-0.1496562665256477, -0.6394005160255389, -0.1772627441375443,
-0.3367347671618135, 0.2068673810983191, 0.39731172031025486,
0.22724327661628946, 0.12257055014634397, 1.382137816641649,
-0.5521978605684328, -0.25485903129099896, 0.09462236391878473,
0.020565644241331373, -0.7980739432638818, 0.03858194387538482,
0.44111581395841803, -0.10770007566693716, 0.2866407689192016, 0.27658522586128,
0.192870610209718, 0.13815807358344878, 0.28690350344426657]
[1.0258697862730768, -0.4112623312705365, -0.3549796922818309,
-0.20647458380544942, -0.2966783680317957, -0.2966783680317957,
-0.21083816612145395, -0.4626687483681174, -0.6124339529351444,
0.07089021980832316, -0.09140841896896176, -0.0483822114902313,
-0.2267494104080366, -0.582169257767289, -0.1918105389057364,
-0.11653499298165544, -0.5378574166412106, -0.7382980678191303,
-0.611399672475665, -0.6683417816651699, -0.3928823076191278,
0.08423192545028285, -0.39122109590433085, -0.310202081292463,
-0.31657453118754036, -0.4467477272341864, 0.0609918276128096,
-0.3825243222992674, -0.582169257767289, -0.7912675302208034,

0.06025141753438595, -0.489835929635145, -0.14418466614373174,
-0.7683458972544441, 0.02319303693135338, -0.6556710036968152,
-0.1930965630056691, -0.22012355374612005, -0.6811087095875991,
-0.5900898482793147, -0.925602488073147, -0.8081008508998586,
-0.3681406060838774, -0.03588825902271891, 0.06929949340613695,
0.32811256337406947, 0.21265136428388634, 1.5988336984431213,
-0.2929517785822414, -0.03398002534984622, -0.05721645099002249,
0.3618292323838321, 0.4785533198893012, -0.5209624527717784, 0.5497706027204458,
0.08482659231891801, 0.41663095543382855, 0.33283598892189004,
-0.05070506667784388, -0.09815405122487994, -0.4935102304563517,
-0.44718907180910794, 0.2504162002893661, -0.07425979931509753,
-0.2359823825350234, -0.7428632273012711, -0.3942797520245947,
-0.6750498151346871, -0.2401670001804638, -0.1622822103493647,
-0.5563458180872878, -0.2504478553026782, -0.10993417335274759,
-0.3038284659789826, -0.15666857319270988, -0.14948836482215758,
-0.2344022514145714, 0.0027534644253891867, 0.18909816634272197,
-0.12469146629621222, -0.09163255638349922, -0.17791021345300764,
-0.4216898979935534, -0.3560686853134471, -0.09357234664733984,
-0.42421077597510626, 0.05833911672801729, -0.2512547297272536,
-0.17563263953866556, -0.16123937026021734, -0.05947364248366298,
-0.29787243047179934, -0.27691198788654087, -0.634907224791894,
0.19096778146463197, -0.16048112863789843, -0.6723234001842155,
-0.17888017822378277, -0.17586739666077952, -0.5868624322830293,
-0.18939815347524883, -0.3205066710534224, -0.6532406232792465,
0.24876011000910328, -0.3355458297934662, -0.40009415951409805,
-0.16057884053923124, -0.3353405486139844, -0.3892439062181392,
-0.06568609044538107, 0.06823158642577394, -0.29087883842608636,
-0.4261327395181676, -0.508396440755165, 0.05554077813573939,
-0.2556561793392562, 0.6311822366152334, -0.1919391201978047,
-0.2860798538711552, 0.11941051144765254, -0.12286162690324227,
0.17253426968639865, 0.763981247697525, 0.03633593837550633,
-0.15009430363356843, -0.3039466559844281, -0.5670754944260326,
-0.5015502429584382, -0.2785405655390156, 0.02251885944596596,
-0.17780908767439815, -0.2895900340115293, -0.461197365037353,
-0.57275383317942, -0.14650073089798368, -0.775987406561485,
-0.3552122475082896, -0.234701986662315, -0.47362263247249015,
-0.3562100313164385, -0.07265654960479631, -0.22095045288868134,
-0.11293687720547627, -0.22165788317722365, -0.5812102684771789,
-0.1330325101828397, 0.01602120941969364, -0.39343814652646797,
-0.38333919988360743, 0.4054675418506838, -0.15979890800658217,
0.5162465426716395, -0.2067609623967102, -0.11866317980368196,
-0.28434477273450776, -0.3897743556969997, -0.23418359102834946,
-0.003554072027819124, -0.37821975843345623, -0.29776489260736505,
-0.6025328430704265, 0.7377504697522319, -0.10211923703805803,
-0.47682968347247684, -0.0863606223510222, 0.013550673014220053,
-0.6072546834790797, -0.0798815519569527, -0.20371940044972045,
-0.4268358389361234, 0.32371824621127615, 0.199424416123984,
-0.2730326375508356, 0.043306389885385904, -0.06794043484120874,

0.0227271514651065, -0.36778032023006807, -0.3571654026363833,
-0.7126710311131876, 0.07236915640490728, -0.7433464306324521,
0.10457754142293038, 0.21314781613727418, 0.029617463147026856,
-0.25197503432416307, -0.25772676616661927, 0.031835638533911394,
0.13337293205818046, 0.21571052285841502, -0.4785593558127078,
-0.32558487117256385, -0.4358004540423299, -0.05331777162638201,
-0.20266094390981249, -0.1256857947237569, 0.0396169582370862,
-0.2967512570397666, -0.4287743041431911, -0.6069357546214249,
-0.1700055453484541, -0.5059795202351601, -0.11397655233319391,
-0.1041703781180017, 0.09711105677932325, 0.1307739101307324,
0.07372425960132173, -0.33060356732830787, -0.2143554333054305,
-0.35222313221278795, -0.4810101563297344, -0.20974044771410782,
0.08302891542330443, -0.32728405331637195, -0.33260132058515307,
-0.06321861516544668, -0.21035709093858967, -0.2872410942397055,
0.19550725305163583, -0.15936668374633456, -0.297038244382659,
-0.43943107753121935, -0.054521628131888565, -0.12671387164142842,
-0.40389700198374545, -0.12544580582662598, -0.18586668511477109,
-0.630329735541189, -0.21055576568345907, -0.05490934344458993,
-0.5350501496119633, -0.5354878548707638, -0.14217452192538585,
0.13464645612729226, -0.0609626489700896, -0.019878556977403313,
-0.0005393838396360921, -0.3119242711698855, 0.16818428253466952,
0.3846865683456414, -0.40793259704711315, 0.21448676737694422,
0.17833659326725598, -0.23579152804049938, -0.21946072645004108,
-0.20830798479073728, 0.09595365518677706, 0.05155763921600503,
-0.03837541237702606, -0.17550036355879076, 0.11887409377470623,
-0.1658969231588028, 0.3614982538480341, -0.22154418063870318,
0.6102993088458594, -0.5857542554714679, -0.15984754287892616,
-0.3042642365349031, -0.351809179541463, -0.11502894289411109,
-0.2658403326356276, 0.2190221968442154, -0.16232287272647358,
-0.37647687944649355, -0.27396377540939154, -0.41479343300864746,
-0.41946671514077716, -0.40440493039407666, -0.25004357561602686,
-0.2921916764626442, -0.8125276378239602, -0.15689801609599077,
-0.35867178933817057, 0.6959770392085014, 0.04768304030882333,
-0.5816602508789499, -0.03548346392778497, -0.6365432910132752,
0.09421716036684813, -0.1128113619862754, -0.14800415592857327,
-0.31935364725117815, -0.6554951226303982, -0.30629689730211335,
-0.1795069508547581, -0.39878656502415954, -0.45263490587964517,
0.08572440413442241, -0.3135837771179986, -0.1496562665256477,
-0.7475151514463716, -0.18886796605491954, 0.1185210613491423,
-0.4105409861069635, -0.05986921646048944, -0.17917338152871182,
-0.1772627441375443, -0.01850648208943426, -0.3367347671618135,
-0.19969748901875017, -0.4176465265662031, -0.0078037534268241,
-0.1873855166974157, 0.04289514026718595, -0.41494962051632045,
0.12257055014634397, -0.0008624932921612126, 1.382137816641649,
-0.5521978605684328, -0.24156056160622086, -0.22863303848621808,
0.06008222908233625, 0.03858194387538482, 0.44111581395841803,
-0.2931430194648842, 0.27658522586128, -0.4214053218807345,
-0.37696536276165077, -0.2360029549858149, -0.0692472186434217,

```
-0.08623746326322505]
```

```
[ ]: # importing comparative indices/assets, gold is still saved
sp1500 = yf_import("^SP1500", periods.iloc[0]["Start"], periods.
 ↪iloc[-1]["Last"])
nasdaq = yf_import("^IXIC", periods.iloc[0]["Start"], periods.iloc[-1]["Last"])
WTI = web.DataReader(
    "WTISPLC", "fred", periods.iloc[0]["Start"], periods.iloc[-1]["Last"]
)
```

```
[*********************100%***********************]  1 of 1 completed
[*********************100%***********************]  1 of 1 completed
```

```
[ ]: performance = periods
gold_performance = []
sp1500_performance = []
nasdaq_performance = []
WTI_performance = []

# creating a data-frame to compare performances of the portfolios with common␣
 ↪indices and "crisis" resources
for index, row in periods.iterrows():
    gold_performance.append(
        (
            1
            + ((gold.loc[row["Last"]].values) - (gold.loc[row["Start"]].values))
            / (gold.loc[row["Start"]].values)
        ).item()
    )

    sp1500_performance.append(
        (
            1
            + ((sp1500.loc[row["Last"]].values) - (sp1500.loc[row["Start"]].
 ↪values))
            / (sp1500.loc[row["Start"]].values)
        ).item()
    )

    nasdaq_performance.append(
        (
            1
            + ((nasdaq.loc[row["Last"]].values) - (nasdaq.loc[row["Start"]].
 ↪values))
            / (nasdaq.loc[row["Start"]].values)
        ).item()
    )
```

```
    WTI_performance.append(
        (
            1
            + ((WTI.loc[row["Last"]].values) - (WTI.loc[row["Start"]].values))
            / (WTI.loc[row["Start"]].values)
        ).item()
    )
```

[ ]:
```
performance["Gold"] = gold_performance
performance["Nasdaq"] = nasdaq_performance
performance["Spot Crude Oil Price WTI"] = WTI_performance
performance["S&P 1500"] = sp1500_performance
performance["all_Grid"] = all_classifier_performance
performance["concluded_Grid"] = concluded_classifier_performance
performance["current_test_Grid"] = current_test_classifier_performance
dfi.export(
    performance.style.set_properties(
        **{"background-color": "white", "color": "black", "border-color":␣
 ↪"#948b8b"}
    ),
    "performance.png",
)
performance
```

[ ]:
```
       Name       Start        Last  Duration      Gold    Nasdaq  \
0  Period 1  2004-04-01  2006-08-01        27  1.626804  1.047715
1  Period 2  2016-09-01  2017-08-01        11  1.000984  1.201220
2  Period 3  2017-09-01  2018-07-01        10  0.957095  1.206716
3  Period 4  2022-01-01  2023-04-01        15  1.125556  0.830855

   Spot Crude Oil Price WTI  S&P 1500  all_Grid  concluded_Grid  \
0                  1.991006  1.148445  1.465244        1.722937
1                  1.063302  1.135936  1.264555        1.383578
2                  1.424729  1.124224  1.282267        1.421511
3                  0.954698  0.900910  1.043568        0.817482

   current_test_Grid
0           1.502160
1           1.159184
2           1.455724
3           0.983550
```