

# Aiolux Inflation-related Economic Indicators for S&P 500 Sectors

By

Zhenyu (William) Dai  
Yixuan (Jolina) Shen  
Ruohui (Jaki) Tang  
Pathik Rupwate

Supervisor: Roger Moore, MBA  
Sponsor: Biz Chatterjee, CEO of Aiolux

A Capstone Project

Submitted to the University of Chicago in partial fulfillment  
of the requirements for the degree of

Master of Science in Analytics

Division of Physical Sciences

December 2022

## **Abstract**

The research project's goal was to provide an automatic interpretation of the influence of inflation-related economic indicators on various S&P 500 sectors. Multiple machine learning models were implemented to predict the 50, 100, and 200 days relative performance of all 11 sectors of the S&P 500 Index using related CPI data. The Extremely Randomized Tree Classifier presented the best overall performance among all 33 models with an average accuracy of 78.85%. Models showed a more accurate prediction of the long-term behavior of the S&P 500 sectors' relative performance for 70% of the sectors.

*Keywords:* Investment optimization, Quantitative Finance, Stock Sector Relative Performance Prediction, Machine Learning, Customer Price Index (CPI), S&P 500, Tabular GAN

## **Executive Summary**

The project was dedicated to providing retail-investor-friendly investment research based on CPI. By building a classification model by sectors, this project predicted the relative performance of the S&P 500 sectors to the broader market. The prediction result will be significant to clients because it will provide valuable investment decision support for retail investors.

Most of the past research focused on the stock price prediction by incorporating CPI as one of the financial indexes in the model. This project emphasized on the prediction of the relative performance for each S&P 500 sector which strips out the impact of broad market issues that may be affecting all sectors. With CPI data collected from the US Bureau of Labor Statistics API, this project used CPI indicators as the only predictors to predict the relative performance of S&P 500 sectors and fill in the information gap for retail investors and small corporations.

This project desired to investigate whether S&P 500 sectors (11 individual sectors) outperform or underperform the broader market (Overall S&P 500 Index) in 3 different time periods: 50-day, 100-day, and 200-day. There were a total of 33 models. Due to a small original dataset, this project leveraged the generative adversarial nets (GAN) to generate synthetic data to increase the sample size for the training model. Techniques like SMOTE and interaction terms were implemented to extract additional insights from predictors and boost the model performance. The Extremely Randomized Tree Classifier presented the best overall performance in this project with an average accuracy of 78.85% among all 33 models. Sectors with more direct related CPI indexes tended to achieve higher accuracy than did others. Long-term prediction models outperformed short-term ones in over 70% of the sectors which indicated CPI features tend to capture more long-term behavior of the S&P 500 sectors' relative performance.

## Table of Contents

<b>Introduction</b> . . . . .	<b>1</b>
Problem Statement . . . . .	1
Goals of Analysis . . . . .	3
Scope . . . . .	3
<b>Background</b> . . . . .	<b>4</b>
Literature Review . . . . .	5
<b>Data</b> . . . . .	<b>7</b>
Data Sources . . . . .	7
Descriptive Analysis . . . . .	10
<b>Methodology</b> . . . . .	<b>13</b>
Feature Engineering . . . . .	13
Model Frameworks . . . . .	15
<b>Findings</b> . . . . .	<b>18</b>
<b>Discussion</b> . . . . .	<b>19</b>
<b>Conclusion</b> . . . . .	<b>20</b>
<b>References</b> . . . . .	<b>23</b>
<b>Appendix A : Detailed Model Results</b> . . . . .	<b>24</b>
<b>Appendix B : Example of S&amp;P 500 Sector API</b> . . . . .	<b>25</b>
<b>Appendix C : Example of CPI API</b> . . . . .	<b>26</b>

## List of Figures

1	Traditional Data Source . . . . .	2
2	Method and Data Sources Summarize . . . . .	5
3	Preview of Cleaned Data . . . . .	8
4	Correlation Matrix of S&P 500 and CPI data . . . . .	10
5	Line Plot of S&P 500 and S&P 500 Real Estate . . . . .	11
6	Seasonality Decomposition of S&P 500 Real Estate . . . . .	12
7	Line Plot of CPI data under each section . . . . .	12
8	Seasonality Decomposition of CPI overall . . . . .	12
9	Distribution of the real data (left) and synthetic data (right) for Real Estate Sector . . . . .	14
10	Example of Potential Performance of Each S&P 500 sector vs Overall S&P 500 . . . . .	21

## List of Tables

1	Empirical Analysis . . . . .	16
---	------------------------------	----

## **Introduction**

Aiolux is a financial technology company that was founded in 2019. Its primary mission is to provide complex insights and essential information for investment optimization that is out of reach for retail investors and small investment firms with limited technology analyst resources. Aiolux leveraged machine learning algorithms to predict how each sector performs leading up to and after key events and collated historical performance in similar situations. Using Quant Finance, Statistics, and Machine Learning/Artificial Intelligence to provide automated investment reports, Aiolux worked as a personalized I-Banking analyst and provided instant deep analysis that otherwise requires significant time and labor.

This capstone project aimed to design a machine learning algorithm that uses the Customer Price Index (CPI) as an anchor to predict the sector's performance compared to the overall S&P 500 Index. The goal was to interpret how the economic event would affect retail investors' portfolios and create investment opportunities for them. The principal client contractor is Biz Chatterjee, founder of Aiolux, an experienced operator with prior business (P&L, Multifunctional organization). He graduated from The University of Chicago Booth School of Business in 2011 and worked as general manager (North America) at Groupon Inc.

### **Problem Statement**

Aiolux intended to provide a service to clients that automatically interpreted news events and evaluated the potential impact on various S&P 500 sectors. Therefore, as CPI data was released every month, Aiolux will automatically surface insight predicting a positive/negative 50, 100, and 200-day performance of sectors relative to the broader market (S&P 500). As there was a change in investing behaviors, more and more retail investors chose to control and design their own portfolios, and there was an increase in demand for automation research for investment suggestions. Due to the massive news sources and the volatile and complex nature of the market, retail investors were at a signif-

icant disadvantage without professional analysts and fundamental economic knowledge. Since the stock market is very sensitive to external information, starting to use the most fundamental and influential economic indicator for inflation can give general knowledge about the market. Below, there was a list of scenarios in the order of accessibility. Scenarios 1-2 involve the data sources most used in traditional stock market prediction, which also align with this research (Weng [2017]).

**Figure 1.** *Traditional Data Source*

Scenario #	Description
1	Market data
2	Market data, technical indicators
3	Market data, technical indicators, Wikipedia Traffic
4	Market data, technical indicators, Google news counts
5	Market data, technical indicators, Wikipedia Traffic, generated features
6	Market data, technical indicators, Google news counts, generated features
7	Market data, technical indicators, Wikipedia traffic, Google news counts, and generated features

Market data and technical indicators are two of the most accessible information sources generally used by investors to estimate the future performance of their portfolio and change the weight of their stock accordingly.

Previous academic researchers had mostly done systematic analysis using historical stock prices with additional technical indicators to predict market prices using the time series concept. This project is here to fill in the information gap for retail investors and small corporations. The machine learning algorithm in this project took a different path by using CPI as the primary resource to build classification models by sector to predict the relative performance of each sector, overperform or underperform, compared to the S&P

500 Index. It is important to provide investors with reports that can accurately inform them which sector of the stock will perform better and which will not so that they can monitor their portfolio to maximize the gain.

With the prediction result, customers will clearly understand how to allocate their money in their trading portfolio. Unlike news and rumors that sometimes can be ambiguous, CPI releases every month and keeps track of the price of goods in different sectors. It is a direct indicator of inflation and has a significant impact on the understanding of the market. With this fundamental knowledge in mind, retail investors can have more control over their money and investment plan.

### **Goals of Analysis**

The goal of this analysis was to capture each sector's relative performance based on the information given by CPI. There are different CPI data corresponding to different baskets of goods. When new CPI data is released every month, Aiolut will automatically surface insight predicting a positive/negative 50, 100, and 200-day performance of sectors relative to the broader market (S&P 500). Suppose there is an economic recession, all sectors may be underperforming over the next 100-day period. However, the Consumer Discretionary sector may underperform abnormally on a relative basis due to the added impact of high inflation. This project wanted to provide an algorithm that can predict the future relative performance of each sector compared to the overall S&P 500 Index so that when Aiolut generates the customized report with financial information, they will be able to include the result in the report. Furthermore, the report will allow retail investors and corporations that used Aiolut's services to have complex insights essential for investment optimization like those large and sophisticated trading firms.

### **Scope**

For Aiolut, in this analysis, the final algorithm provided a model that will suggest the inflation impact on each sector. It will be a practical add-on to the customized report for Aiolut. Python will be used to build a classification model in this capstone project, allow-



ing the client to directly implement the algorithm and apply it to the report production.

The project's desired outcome was to predict future relative performance by sectors based on CPI. However, as figure 1 shown above, there is also other information on the field that is important to predict the relative performance of each sector. This project solely depended on CPI to make the recommendations.

In future research, this project aimed to expand the news events and data sources beyond the CPI data to predict relative performance by sector.

## Background

Unlike large trading firms, retail investors have limited access to detailed historical stock prices and related events that drove the market. Most investors do not have professional financial knowledge, and they make their investment decisions mainly based on the news and rumors which are available in the market. Investors want to understand the impact of important news events on their portfolios, Economics Indicators are an important type of news in this respect. The CPI data releases every month, and it looks at the cost changes from the perspective of consumers for their day-to-day lives. Different CPI data corresponding to different sectors is released by the US government (Bureau of Labor Statistics) every month and is broadly used by the business community to understand inflation.

$$CPI_t = \frac{C_t}{C_0} * 100 \quad (1)$$

where  $CPI_t$  is the consumer price index in current period.  $C_t$  and  $C_0$  refer to cost of market basket in current period and base period. The CPI score represents the average change in prices that consumers will spend on a basket of goods and service overtime. CPI at territorial level is highly inflated by number of residence, average income and year of education; therefore this analysis using all US seasonal adjusted CPI for stock market is national level data ([Gabrielli et al. \[2016\]](#)).

## Literature Review

### *Application of machine learning techniques for stock market prediction*

In Application of machine learning techniques for stock market predictions, CPI was used as an additional variable along with historical stock price to implement time series analysis to predict future stock price (Weng [2017]). There is very limited research that depends solely on CPI data to predict stock market performance. The table shown below summarizes the AI approaches that scholars have utilized in their research. Even though this analysis is not using stock price as an independent variable, the methodology and the logic in this research are still applicable.

**Figure 2.** *Method and Data Sources Summarize*

Paper	Sources for Knowledge Base			AI Approach
	Traditional	Crowd-sourcing	News	
Kimoto et al. (1990)	✓			ANN
Lee and Jo (1999)	✓			Time Series
K.-j. Kim and Han (2000)	✓			ANN, GA
K.-j. Kim (2003)	✓			SVM
Qian and Rasheed (2007)	✓			ANN, DT
S.-T. Li and Kuo (2008)	✓			ANN
Schumaker and Chen (2009)			✓	SVM
Vu et al. (2012)		✓		DT
M.-Y. Chen, Chen, Fan, and Huang (2013)	✓			ANN
Adebisi, Adewumi, and Ayo (2014)	✓			ANN, ARIMA
Nguyen, Shirai, and Velcin (2015)		✓		SVM
Shynkevich, McGinnity, Coleman, and Belatreche (2015)			✓	ANN, SVM
Chourmouziadis and Chatzoglou (2016)	✓			Fuzzy System
<b>Our Financial Expert System</b>	✓	✓	✓	ANN, SVM, DT

Previous analysis commonly used artificial neural networks (ANN), Time series analysis, support vector machines (SVM), and decision trees (DT) with a traditional data source, which is similar to the data used for analysis in this paper. Weng provided a detailed comparison using all three models and concluded that SVM has the worst performance since it faces the curse of dimensionality, while ANN and Boosting Decision Trees have a better result, especially after PCA. Boosting Decision Trees have a slightly better result than ANN.

When developing risk models for binary data with small or sparse data sets, the stan-

dard maximum likelihood estimation (MLE) based logistic regression faces several problems including biased or infinite estimate of the regression coefficient and frequent convergence failure of the likelihood due to separation. This is also applicable to the data used in this project as CPI data is only released monthly.

### ***Performance of Firth-and logF-type Penalized Methods in Risk Prediction for Small or Sparse Binary Data***

In Performance of Firth-and logF-type Penalized Methods in Risk Prediction for Small or Sparse Binary Data ([Rahman and Sultana \[2017\]](#)), the MLE showed poor performance in risk prediction with small or sparse data sets. All penalized methods offered some improvements in calibration, discrimination and overall predictive performance. Although the Firth-and logF-type methods showed almost equal amounts of improvement, Firth-type penalization produced some bias in the average predicted probability, and the amount of bias was even larger than that produced by MLE of the logF(1,1) and logF(2,2) penalization. LogF(2,2) provided slight bias in the estimate of regression coefficient of binary predictor but logF(1,1) performed better in all aspects. The logF-type penalized method, particularly logF(1,1), could be used in practice when developing risk models for small or sparse data sets.

### ***Generative Adversarial Nets (GAN)***

The problem of small datasets could be addressed by generating new synthetic data to expand the current dataset. Generative adversarial nets (GAN) is an industry-leading method for generating synthetic data. GANs are composed of two deep networks, the generative model G that captures the data distribution, and the discriminative model D that estimates the probability that a sample came from the training dataset rather than G ([Goodfellow et al. \[2020\]](#)). The training procedure for G is to maximize the probability of D making a mistake. Competition in this game drives both models to improve their methods until the synthetic data from the generator is indistinguishable from the training dataset.

Tabular data can contain a mix of discrete and continuous columns in the table. Continuous columns may have multiple modes whereas discrete columns are sometimes imbalanced making the modeling difficult. Although GAN is usually used for image-type data, there is a variation called Conditional Tabular GAN (CTGAN) which can be used to generate tabular datasets such as CPI data. CTGAN is a GAN-based method to model tabular data distribution and sample rows from the distribution (Xu et al. [2019]). CTGAN augments the training procedure with mode-specific normalization, architectural changes, and addresses the data imbalance by employing a conditional generator and training-by-sample. “Modeling Tabular Data using Conditional GAN”, 2019 (Xu et al. [2019]) evaluated CLBN, PrivBN, MedGAN, VeeGAN, TableGAN, CTGAN, and TVAE using a benchmark framework. CTGAN outperformed both Bayesian network baselines and other GANs tested.

## **Data**

### **Data Sources**

The source of the stock performance data came from Aiolux API provided by the client. The URL of the API was S&P 500 Sectors API Endpoint. It had one Payload: symbol. Symbol Payload took in the symbol of the S&P 500 Sector. For example, the symbol of S&P 500 Real Estate was SP500-60.

The retrieved data was in the form of JSON and consisted of a list of daily stock prices data, including Open, High, Low, Close, Adj Close etc. After retrieving the API data, the algorithm parsed them into Pandas Dataframe for further analysis.

The source of CPI Data was the US Bureau of Labor Statistics API. [The US Bureau of Labor Statistics](#) provided an API Endpoint where users could specify the CPI Series IDs and Time Period to retrieve monthly CPI data. Series IDs were unique identifiers composed with CPI properties code, geolocation code and item code. For example, the seasonal adjusted CPI for housing items with all urban consumers was CUSR0000SAH. CUSR meant CPI with seasonal adjustment. 0000 meant all geo locations. SAH meant

Housing CPI items.

Multiple series could be requested at a time. The JSON object consisted of a list of data including year, month and monthly CPI data, for each series. The API Request was sent through Python and this project parsed the JSON output to Pandas DataFrame for further analysis.

### **Target Variable (S&P 500)**

The target variable was a binary variable representing whether the S&P 500 Sector outperforms or underperforms the S&P 500 Overall in the same period of time.

**Figure 3. Preview of Cleaned Data**

	S&P 500	S&P 500 Real Estate (Sector)	S&P 500 Performance (%)	S&P 500 Real Estate (Sector) Performance (%)	S&P 500 Real Estate (Sector) Outperform/Underperform	overall	commodities	housing	shelters	transportations
2021-11-24	4701.46	306.609985	4.852728	4.852599	UNDERPERFORM	280.126	210.452	288.259	341.963	326.397
2021-11-26	4594.62	298.359985	1.616263	2.854384	OUTPERFORM	280.126	210.452	288.259	341.963	326.397
2021-11-29	4655.27	302.339996	1.484354	1.804830	OUTPERFORM	280.126	210.452	288.259	341.963	326.397
2021-11-30	4567.00	295.940002	1.396956	2.585965	OUTPERFORM	280.126	210.452	288.259	341.963	326.397
2021-12-01	4513.04	291.970001	2.136404	2.589598	OUTPERFORM	280.126	210.452	288.259	341.963	326.397

This project investigated the following eleven S&P 500 Sectors (Symbol): Communication Services (previously included in XLK and newly published as XLC since 2018 Jun), Consumer Discretionary (XLY), Consumer Staples (XLP), Energy (XLE), Financials (XLF), Health Care (XLV), Information Technology (XLK), Industrials (XLI), Materials (XLB), Real Estate (XLRE), Utilities (XLU), S&P 500 Overall (SPY).

### **Predictor Variables (CPI)**

The [Consumer Price Index](#)(CPI) measured the average change over time in the prices paid by urban consumers for a market basket of consumer goods and services. CPI data was subdivided into geography, item categories, and more. Hundreds of series correspond to price index changes to particular categories, e.g., Footwear, men's apparel, and women's apparel. However, this project only selected top-level (Level 0 & Level 1) categories to significantly reduce the CPI numbers of interest. In the future, more categories

might be added if necessary. All urban US data was used for the geography selection instead of the regional subdivide. In addition, this project would only include seasonally adjusted CPI data because seasonally adjusted CPI data has less noise.

This analysis investigated how certain CPIs have an impact on S&P 500 Sectors. Therefore, there were multiple CPI indexes related to one S&P 500 Sector. For example, the S&P 500 Real Estate would be at least correlated to these five CPI data series: Overall CPI, Commodities CPI, Housing CPI, Shelter CPI, and Transportation CPI. For other CPI data series, the Apparel CPI, for example, might not be a relevant independent variable for the S&P 500 Real Estate Sector performance but would be essential for the S&P 500 Consumer Discretionary Sector because it contained Apparel/Clothing companies.

### ***Data Gap and Concerns***

This project didn't identify any gaps in data. Stock data and CPI data were both monitored and collected by government agencies. Therefore, there was no data gap in these data sources. However, this project had concerns on the size of data. S&P 500 Overall data started from December 1946 but most S&P 500 Sectors data starts from 1993. This meant this project only had around 30 years of data for our target variable. In addition, CPI data was only available per month. The US Bureau of Labor Statistics did not provide daily CPI data. Therefore, the number of the predictor data would be around 347 records per sector (calculated by the number of months between 1993 Jan and 2021 Dec) .

Aiolux was aware that the lack of data might bring many potential problems on the model accuracy. Small samples of data were much less likely to accurately reflect the population's distribution. In addition, the maximum likelihood estimation procedure that powers many classification algorithms was only unbiased with large datasets. Therefore, this project leveraged GAN and created several models of subsets of the data chosen with replacement by the means of bootstrapping or cross validation. Then, this project could get a better estimation of our model parameters before pushing all our data to the model.

## Descriptive Analysis

### Correlation Analysis

This project intended to evaluate CPI data's ability in capturing each S&P 500 sector's performance. Therefore, the correlation analysis between S&P 500 data and CPI data is important.

**Figure 4.** *Correlation Matrix of S&P 500 and CPI data*



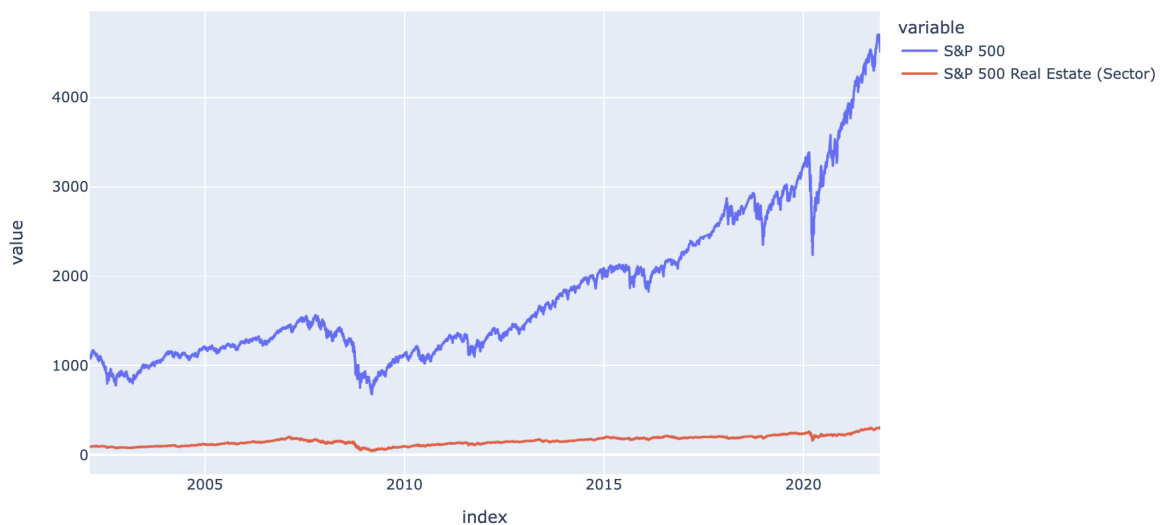
The figure 4 showed the correlations between S&P 500, S&P 500 Sector and related CPI data. This correlation matrix figure used S&P 500 Real Estate as an example. From the figure 4, the S&P 500 Real Estate is highly correlated to identified CPI data. The difference

in S&P 500 Real Estate with different CPI data also showed the strength of the correlation. The commodities CPI had 0.71 correlation coefficient with S&P 500 Real Estate because commodities CPI contained other commodities than housing and shelter. On the other hand, housing and shelter CPI both had over 0.9 correlation coefficient with S&P 500 Real Estate because these two CPI data were the main commodities in the Real Estate industry.

### ***Time Series Analysis***

Time series analysis was focused on the trends of S&P data and CPI data because both S&P data and CPI data were time series. Therefore, the line plot showing their trends could better describe their features.

**Figure 5.** *Line Plot of S&P 500 and S&P 500 Real Estate*

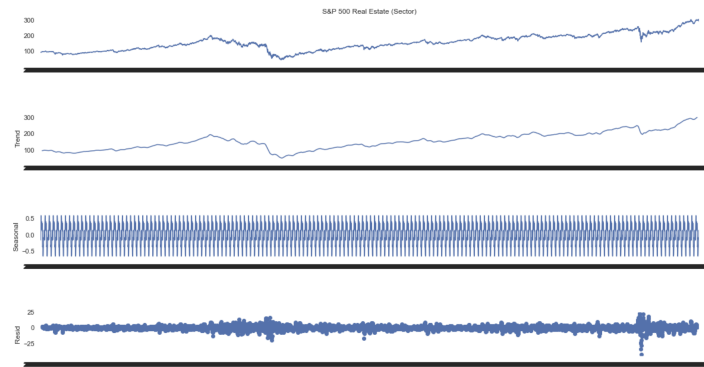


The figure 5 was the line plot of S&P 500 and S&P 500 Real Estate. From the line plot, the trend could be identified as an overall upward trend. The trend experienced some ups and downs as a stock generally did. It did not show a strong seasonality as the seasonal component does not give any clearer picture from seasonality analysis (figure 6).

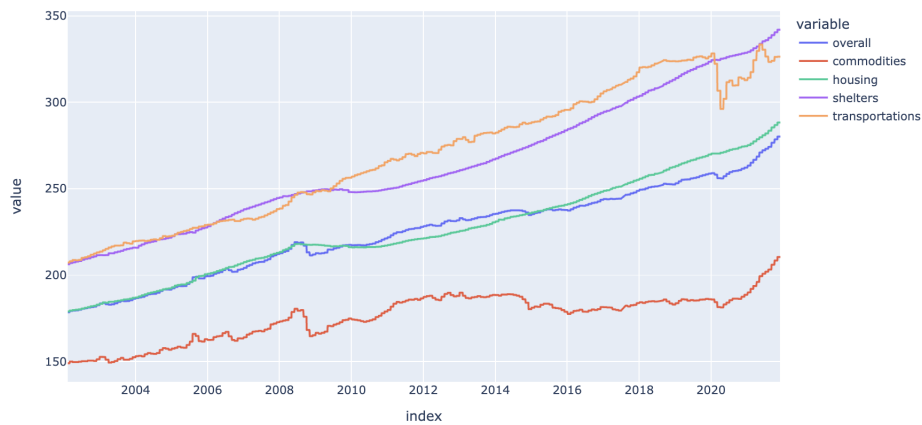
The figure 7 is the line plot of CPI data under each section. From the line plot, the trend of CPI data could be identified as an overall upward trend. The same upward trend further supported the previous high correlation results. The CPI data also had a similar



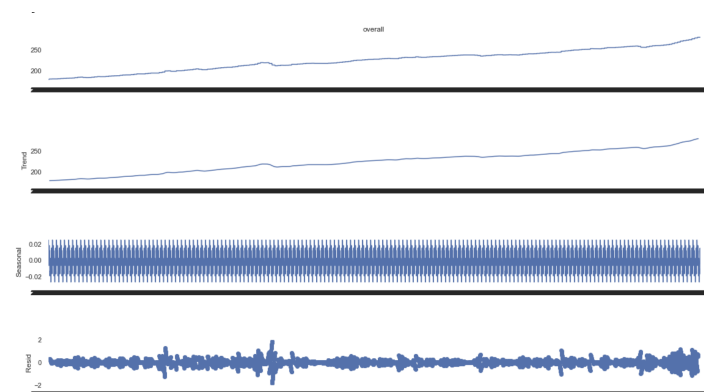
**Figure 6.** *Seasonality Decomposition of S&P 500 Real Estate*



**Figure 7.** *Line Plot of CPI data under each section*



**Figure 8.** *Seasonality Decomposition of CPI overall*



positive trend and variance to the S&P 500 data. The CPI data also did not show a clearer picture in the seasonal component from seasonality analysis (figure 8), thus did not have

a strong seasonality. The similarity between CPI data and S&P 500 Sector data from time series perspective further supported the correlation between the predictive variable and target variable.

## **Methodology**

### **Feature Engineering**

This project's feature engineering started by joining CPI data from different sections together. After joining by date, the CPI data table was converted with an index of the datetime object in the format of year and month. The same steps were also applied to the S&P 500 and the S&P 500 Sector data. After joining the S&P 500 and the S&P 500 Sector by date, the Close Price value column was converted to the S&P 500 price and the S&P 500 Sector price. Then, the target variable the S&P 500 Sector, Outperform/Underperform, was calculated by comparing the S&P 500 Sector and the S&P 500 Overall performance. The performance of each sector was calculated by comparing the Adj Close of the current period to that of the last period. This analysis used relative 50-day performance, 100-day performance, and 200-day performance.

The CPI data was joined to the S&P 500 performance table by date. Since CPI only had monthly data, only one record per month was kept from the S&P 500 performance data. The data from the second Thursday of each month was used for the S&P 500 performance data since it was the closest date to CPI release date in each month. Most of the Sectors started from 1993-05-05 and ended in 2021-12-31. Therefore, most of the cleaned dataset had 347 records of data. The only exception was the S&P 500 Real Estate Sector. The S&P 500 Real Estate Sector started 2002-02-11 and thus had 347 counts in the finalized dataset.

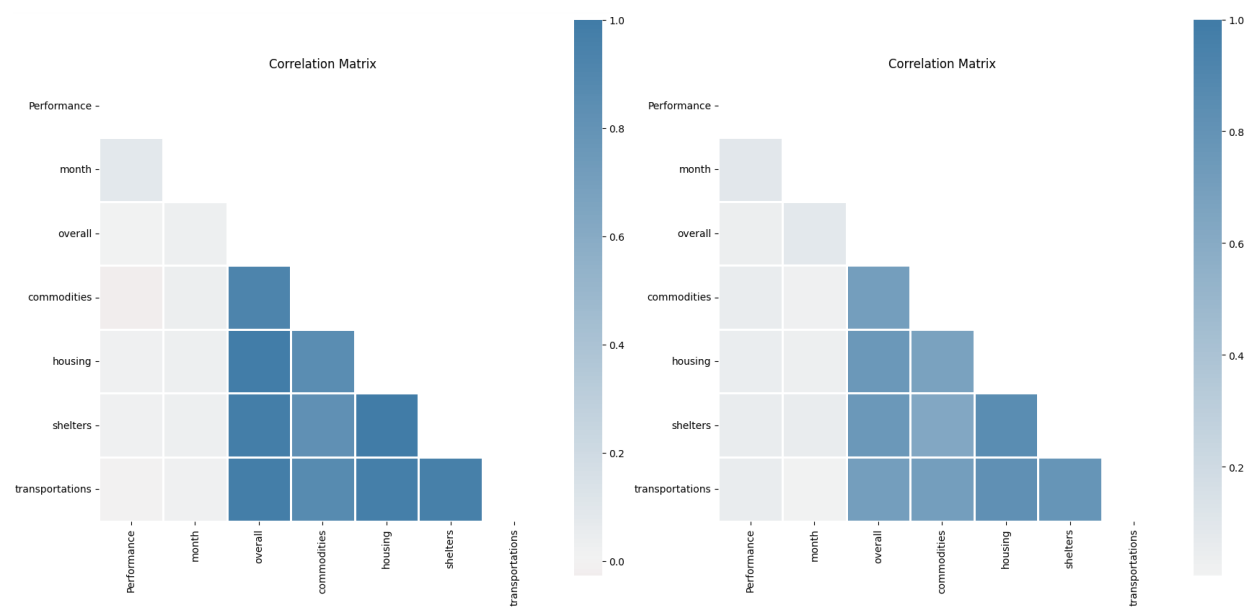
### ***Generating Synthetic Data with CTGAN***

To remedy the problem of the small dataset, an approach to expand the original dataset by generating new synthetic data was implemented for better model performance. As discussed in the literature review, Generative Adversarial Nets (GAN) is an industry-leading method for generating synthetic data. Since the analysis goal was to construct

a classification model to predict whether the sector performance was overperforming or underperforming compared to the overall S&P 500 Index, conditional GAN was applied to take advantage of the classification labels during the training process. Although the method was generally used for image-type data, a variation called Tabular GAN can generate tabular data like this project’s dataset. In conclusion, Conditional Tabular GAN (CTGAN) was chosen as the algorithm to perform the task of generating synthetic data.

For each model, the raw stock performance data and CPI features were used to generate 10,000 new data. With 5,000 epochs in a single training process, the CTGAN model could successfully capture the characteristics of the original input data. After sampling 10,000 synthetic data points, the target variable was calculated accordingly for the new dataset. To ensure the synthetic data could represent the actual data, distribution and correlation were checked for every variable. As shown in Figure 9, the new synthetic data’s distribution was identical to the real data. Parallel to the distribution, the correlation matrices for the two datasets showed highly similar behavior in terms of correlation between all the variables. That said, the new synthetic data was an excellent representation of the real dataset.

**Figure 9.** *Distribution of the real data (left) and synthetic data (right) for Real Estate Sector*



### ***Oversampling the Dataset with SMOTE***

After obtaining the new synthetic data, there existed a slight class imbalance in the dataset. An imbalanced dataset causes a classifier to be overwhelmed by the majority classes and thus omit the minorities, which further leads to a significant performance plunge in the result. To alleviate this problem, the Synthetic Minority Oversampling Technique (SMOTE) was implemented to balance the dataset. SMOTE was chosen over other techniques since it can resolve the overfitting problem caused by random oversampling. The number of data records increased from 10,000 to a range of 12,000-16,000, depending on the different models.

### ***Creating Additional Features with Interaction Terms***

Since this project only considered the related CPI features as predictors, the algorithm utilized interaction terms to capture more insights from the limited attributes. Interaction terms referred to a variable representing the interaction effect between two or more variables. Although non-linear models like bagging and boosting algorithms already have the ability to consider some interactions between variables without explicitly specifying them, recent empirical studies indicate that marginal effects would disguise the interactions, which causes difficulty in separating the interaction effects from marginal effects (Wright et al. [2016]). In order to extract maximal information from the CPI features, interactions between each pair of two CPI indexes were calculated and included as additional predictors. For a sector with 5 CPI predictors originally, the incremental number of features was 10.

## **Model Frameworks**

### ***Empirical Analysis for Feature Selection***

The first step in the modeling phase was an empirical analysis of the correlation between various CPIs and S&P 500 sectors. This step took the initial screening of feature selection since only a part of CPIs were related to each of the sectors and thus influenced each S&P 500 sector's relative performance. The three levels of importance were

as follows: strong direct relation, weak direct relation, and indirect relation. Strong direct relations included CPIs that were identical to the sector's topic. For instance, energy CPI would have a strong direct relation for the Energy Sector. Weak direct relations included CPIs having a first-layer relationship with the sector but not within the sector's definition. A good example would be that fuels and utilities CPI has a weak direct relation to the Energy Sector. Indirect relations would be those CPIs that have some influence to the sector. For instance, transportation services would be the indirect CPI to the Energy Sector. Identifying the three levels of the relationship would be critical for the next step of the analysis and directly reflecting on the final accuracy. Table 1 below present the empirical analysis result:

**Table 1. Empirical Analysis**

Sector	Strong Direct	Weak Direct	Indirect
Energy	Energy (SA0E) Energy Commodities (SACE)	Fuels and utilities (SAH2) Overall (SA0) Transportation (SAT)	Transportation services (SAS4) Utilities and public transportation (SAS24) Commodities (SAC)
Financials	Overall (SA0) Other personal services (SAGS) Purchasing power of the consumer dollar (SA0R)	Housing (SAH) Shelter (SAH1) Services (SAS)	Other services (SAS367) Commodities (SAC)
Communication Services	Communication (SAE2) Information technology commodities (SEEEC) Video and audio services (SERAS)	Education and communication commodities (SAEC) Education and communication services (SAES) Video and audio products (SERAC)	Utilities and public transportation (SAS24) Rent of shelter (SAS2R3) Recreational reading materials
Consumer Discretionary	Overall (SA0) Jewelry and watches (SEAG) Apparel (SAA)	Tobacco and smoking products (SEGA) Durable (SAD) Housing (SAH)	Education (SAE1) "Pets, pet products and services (SERB)" Sporting goods (SERC)
Consumer Staples	Nondurables (SAN) Tobacco and smoking products (SEGA) Food and beverages (SAF)	Household furnishings and operations (SAH3) Household furnishings and supplies (SAH31) Apparel (SAA)	Domestically produced farm food (SAN1D) Footwear (SEAE) Infants' and toddlers' apparel (SEAF)
Health Care	Medical Care (SAM) Medical care commodities (SAM1) Medical care services (SAM2) Personal Care (SAG1)	Overall (SA0)	
Information Technology	Communication (SAE2) Information technology commodities (SEEEC)	Video and audio products (SERAC) Video and audio (SERA) Video and audio services (SERAS)	
Real Estate	Housing (SAH) Shelter (SAH1)	Overall (SA0) Commodities (SAC)	Transportation (SAT) Transportation services (SAS4)
Utilities	Utilities and public transportation (SAS24) Fuels and utilities (SAH2)	Overall (SA0) Shelter (SAH1)	
Materials	Fuels and utilities (SAH2) Energy Commodities (SACE)	Transportation (SAT) Energy (SA0E)	Transportation services (SAS4)
Industrials	Energy (SA0E) Services (SAS)	Transportation (SAT) Commodities (SAC) Durable (SAD)	Communication (SAE2)

## ***Predictive Modeling***

This project focused on developing classification models that predict the relative performance of the S&P 500 sectors compared to the broader market in a 50-day, 100-day, or 200-day period. Since there are 11 sectors, this project presented 33 models in total. For each sector, depending on the length of the cleaned dataset, multiple sub-samples were taken to train the model independently for comparison. The model with the best performance was used as the final model for the specific sector in the given timeframe.

As introduced in the descriptive analysis, CPI features selected for a single S&P sector were highly correlated with each other. To deal with the multicollinearity, this project only experimented with non-linear models such as bagging and boosting models. In this project, the predictors for the model had relatively high variance, and little noise existed in the generated data. The model inherited bagging methodology was better suited for the analysis than boosting algorithms since bootstrap aggregation (bagging) can reduce variance within a noisy dataset and prevent overfitting. In bagging, several random data samples are selected from a training set to be trained independently by weak learners. The final result from bagging is an aggregation of the output from the individual weak learners to achieve the optimal prediction. In this project, the models incorporating bagging algorithms were Random Forest Classifier, Extremely Randomized Trees Classifier, and Bagging Classifier.

To validate the hypothesis and compare the model performance, this project also implemented boosting algorithms, such as Gradient Boosting Classifier, Adaptive Boosting Classifier, and other non-linear models like Support Vector Machines (SVM) and Naive Bayesian Classifier.

## ***Model Validation***

Stratified K-fold cross validation was used to validate each model. It is a resampling procedure used to evaluate machine learning models on a limited data sample. This technique generally results in a less biased or optimistic estimate of the model than other

methods, such as a simple train/test split.

Since the final sample size after generating the synthetic data for each model was over 12,000, this project utilized 10-fold cross validation. As  $k$  gets larger, the difference in size between the training set and the resampling subsets gets smaller. As this difference decreases, the bias of the technique becomes smaller which leads to a more satisfactory result.

### ***Model Evaluation Metrics***

The following evaluation metrics were used since models for this project were classification models:

**Accuracy** is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

**Precision** is defined as the fraction of relevant examples (true positives) among all of the examples which were predicted to belong in a certain class.

**Recall** is defined as the fraction of examples which were predicted to belong to a class with respect to all of the examples that truly belong in the class. Recall was especially useful as this project emphasized on measuring how good the model was at correctly predicting the classes. An incorrect prediction may cause inaccurate investment suggestions to consumers.

**The F1 score** combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers.

## **Findings**

Each S&P 500 sector required three models to predict its performance versus the overall S&P 500. One model each was required to predict the relative performance over the next 50, 100, and 200 trading days. This brought the total number of models required to 33 models for the 11 different S&P 500 sectors.

The naive models for each sector had an average accuracy of 37% and their training dataset contained only about 234 records. To deal with the problem of small datasets, the Log-F(m,m) models were trained on the real small datasets to enhance the performance. This method had an average accuracy of 58%. As discussed in the feature engineering section, the final models leveraged the techniques of CTGAN, SMOTE and Interaction Terms which led to over 12,000 records in the dataset and achieved an average accuracy of 78.85%.

The Random Forest, Adaptive Boosting, Gradient Boosting, Extremely Randomized Trees, Bagging, Support Vector Machines, and Naive Bayes classifiers were then trained on the synthetic data. Cross Validation (CV) accuracy, Test accuracy, Precision, Recall, and the F-1 Score were used to evaluate and compare the models (Appendix A).

The highest accuracy model for every sector and time period was the Extremely Randomized Trees Classifier. The Bagging and Random Forest models had greater accuracy than the Naive Bayes and Boosting methods in most cases. The average accuracy of the Extremely Randomized Trees for the 33 models was 78.85%. The highest average accuracy across three time periods was 86.13% for the Energy sector and the lowest was 72.87% for the Consumer Staples sector.

Each sector had 3 models to predict its performance over 50, 100 and 200 trading days. The longer term predictive models had greater accuracy than the short term models for majority of the sectors. The S&P 500 sectors with more directly related CPI indicators had models which displayed greater accuracy on average than sectors with fewer related CPI indicators.

## **Discussion**

The small dataset issue was addressed by the utilization of CTGAN on stock performance raw data and CPI data to generate the synthetic data. The correlation matrices of the real data and the synthetic data confirmed that the synthetic data had a similar distribution to the real data. Models using SMOTE and Interaction Terms on the synthetic data



achieved an accuracy of 78.85%.

The best performing model across every evaluation metric is the Extremely Randomized Trees Classifier. The other models with bagging methodology performed better than the Boosting methods and the other non-linear models. This validated the previous hypothesis that bootstrap aggregation algorithms were the best fit for this project's data and context.

Sectors with more directly related CPI indexes tended to achieve higher accuracy than the others. Sectors like Energy and Information Technology with an accuracy of 86.13% and 82.26%, respectively, had well defined strong direct relationships with CPI indicators. Sectors such as Consumer Staples and Utility with an accuracy of 72.97% and 74.89%, respectively, had fewer direct relationships.

Long-term prediction models also outperformed short-term ones in over 70% of the sectors. This indicated that CPI features tended to capture more long-term behavior of the S&P 500 sectors' relative performance. Longer term investments are on average safer to predict than short term investments using these models.

These models did have their limitations. The average accuracy achieved by all 33 models was 78.85%. While this is a high accuracy, about 21% of the time the investors could still be given incorrect insights as to the performance of the S&P sector when compared to the overall S&P 500, which could lead to an unsatisfactory investment.

The model had a generalization challenge as the original real dataset prevented the machine learning models from generalizing and conducting a robust result. The generated synthetic data causes bias as potential gaps exist between the real data and generated data. Steps were taken to minimize the bias but some bias will persist.

## **Conclusion**

The final deliverables for the project were the models which used CPI data to predict the comparative performance between one S&P 500 sector to the overall S&P 500 over the next 50, 100, and 200 trading days. There were 3 models for each of the 11 S&P 500

sectors, which brought the total deliverables to 33 models.

The models could enable potential investors to correctly invest in the better performing S&P 500 sectors with up to 86% accuracy. This could help them to diversify their portfolios efficiently and grow at a potentially greater rate than the overall S&P 500.

Below is an example of potential findings which could provide decision support to retail customers for which sector and period of time to invest in.

**Figure 10.** *Example of Potential Performance of Each S&P 500 sector vs Overall S&P 500*

Sector	50 Days Performance	100 Days Performance	200 Days Performance
Real Estate	Outperform	Outperform	Underperform
Information Technology	Underperform	Outperform	Outperform
Communication Services	Underperform	Underperform	Outperform
Consumer Discretionary	Underperform	Outperform	Underperform
Consumer Staples	Outperform	Underperform	Underperform
Energy	Underperform	Outperform	Underperform
Financials	Underperform	Underperform	Outperform
Health Care	Outperform	Outperform	Outperform
Industrials	Underperform	Outperform	Underperform
Materials	Underperform	Outperform	Underperform
Utilities	Underperform	Outperform	Outperform

The above table shows the insights the investors would have available to them. They could then decide to invest in sectors which would outperform the overall S&P 500. With an average accuracy of 78.85%, this would result in the portfolio growing at a greater rate than the overall market.

The project is ethically sound. Existing S&P 500 and S&P 500 sector data as well as

CPI data is freely available and was used to predict how sectors will perform compared to the S&P 500 to give our clients the information on which sectors they should invest in and for what period of time.

Future steps to improve the models are to include data from crowd-sourcing and news sources. This should greatly improve the model accuracy and impact on investment outcomes. More advanced techniques to reduce the gaps and bias between the synthetic data and real data can be implemented. For the empirical analysis, this project only included level 0 and 1 CPI indexes. For future improvement, more detailed CPI indicators can be investigated and used as predictors to further boost the model performance.

## References

- Lorenzo Gabrielli, Giovanni Riccardi, and Luca Pappalardo. Using retail market big data to nowcast customer price index. 2016.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- M Shafiqur Rahman and Mahbuba Sultana. Performance of firth-and logf-type penalized methods in risk prediction for small or sparse binary data. *BMC medical research methodology*, 17(1):1–15, 2017.
- Bin Weng. Application of machine learning techniques for stock market prediction. 2017.
- Marvin N Wright, Andreas Ziegler, and Inke R König. Do little interactions get lost in dark random forests? *BMC bioinformatics*, 17(1):1–10, 2016.
- Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

# Appendix A: Detailed Model Results

**Table 2. Detailed Model Results**

		50 Days		100 Days		200 Days	Average Accuracy
Communication Services	Accuracy:	71.74%	Accuracy:	76.90%	Accuracy:	78.47%	75.70%
	Precision:	70.52%	Precision:	79.17%	Precision:	81.22%	
	Recall:	73.25%	Recall:	74.15%	Recall:	74.67%	
	F-1 Score:	71.86%	F-1 Score:	76.58%	F-1 Score:	77.80%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
Consumer Discretionary	Data Used:	All data	Data Used:	All Data	Data Used:	All Data	78.08%
	Accuracy:	73.94%	Accuracy:	84.14%	Accuracy:	76.15%	
	Precision:	75.66%	Precision:	83.61%	Precision:	77.33%	
	Recall:	71.03%	Recall:	84.97%	Recall:	74.72%	
	F-1 Score:	73.27%	F-1 Score:	84.28%	F-1 Score:	76.00%	
Consumer Staples	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	72.87%
	Data Used:	After 2010	Data Used:	All Data	Data Used:	After 2010	
	Accuracy:	70.38%	Accuracy:	76.76%	Accuracy:	71.48%	
	Precision:	71.36%	Precision:	80.69%	Precision:	70.46%	
	Recall:	66.86%	Recall:	70.37%	Recall:	73.51%	
Energy	F-1 Score:	69.04%	F-1 Score:	75.18%	F-1 Score:	71.96%	86.13%
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	All Data	Data Used:	All Data	Data Used:	All Data	
	Accuracy:	83.25%	Accuracy:	87.43%	Accuracy:	87.70%	
	Precision:	88.22%	Precision:	93.45%	Precision:	89.57%	
Financial	Recall:	77.24%	Recall:	80.77%	Recall:	85.46%	76.79%
	F-1 Score:	82.36%	F-1 Score:	86.65%	F-1 Score:	87.47%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	After 2010	Data Used:	After 2010	Data Used:	After 2010	
	Accuracy:	73.49%	Accuracy:	74.04%	Accuracy:	82.83%	
Health Care	Precision:	73.71%	Precision:	73.93%	Precision:	81.68%	76.80%
	Recall:	72.40%	Recall:	74.60%	Recall:	84.06%	
	F-1 Score:	73.05%	F-1 Score:	74.26%	F-1 Score:	82.85%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	All Data	Data Used:	After 2000	Data Used:	After 2010	
Industrials	Accuracy:	77.39%	Accuracy:	79.59%	Accuracy:	73.42%	82.92%
	Precision:	76.80%	Precision:	77.89%	Precision:	72.27%	
	Recall:	78.28%	Recall:	81.62%	Recall:	75.09%	
	F-1 Score:	77.53%	F-1 Score:	79.71%	F-1 Score:	73.65%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
Information Technology	Data Used:	All data	Data Used:	After 2010	Data Used:	After 2010	82.26%
	Accuracy:	79.23%	Accuracy:	84.05%	Accuracy:	85.47%	
	Precision:	77.88%	Precision:	82.83%	Precision:	83.93%	
	Recall:	81.11%	Recall:	85.49%	Recall:	88.13%	
	F-1 Score:	79.64%	F-1 Score:	84.14%	F-1 Score:	85.98%	
Materials	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	80.78%
	Data Used:	All Data	Data Used:	After 2000	Data Used:	All Data	
	Accuracy:	90.22%	Accuracy:	77.70%	Accuracy:	78.85%	
	Precision:	87.48%	Precision:	74.26%	Precision:	78.42%	
	Recall:	93.28%	Recall:	83.25%	Recall:	79.67%	
Real Estate	F-1 Score:	90.29%	F-1 Score:	78.50%	F-1 Score:	79.04%	80.16%
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	All Data	Data Used:	All Data	Data Used:	All Data	
	Accuracy:	78.82%	Accuracy:	80.77%	Accuracy:	82.76%	
	Precision:	80.95%	Precision:	78.75%	Precision:	86.53%	
Utility	Recall:	74.73%	Recall:	83.59%	Recall:	78.91%	74.89%
	F-1 Score:	77.72%	F-1 Score:	81.10%	F-1 Score:	82.54%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	After 2010	Data Used:	After 2000	Data Used:	After 2010	
	Accuracy:	82.70%	Accuracy:	77.75%	Accuracy:	80.03%	
Average Accuracy	Precision:	86.16%	Precision:	80.38%	Precision:	83.72%	78.85%
	Recall:	78.77%	Recall:	73.55%	Recall:	74.70%	
	F-1 Score:	82.30%	F-1 Score:	76.81%	F-1 Score:	78.95%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	All Data	Data Used:	After 2010	Data Used:	After 2010	
	Accuracy:	75.83%	Accuracy:	73.53%	Accuracy:	75.32%	
	Precision:	75.50%	Precision:	75.37%	Precision:	77.59%	
	Recall:	76.76%	Recall:	71.37%	Recall:	72.59%	
	F-1 Score:	76.13%	F-1 Score:	73.32%	F-1 Score:	75.00%	
	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	Best Model:	ExtraTreesClassifier	
	Data Used:	After 2000	Data Used:	After 2010	Data Used:	After 2010	

## Appendix B: Example of S&P 500 Sector API

**Figure 11.** *Example of S&P 500 Sector API*

<https://us-east1-alpine-agent-164921.cloudfunctions.net/sector-index-history?symbol=^SP500-60>

GET


https://us-east1-alpine-agent-164921.cloudfunctions.net/sector-index-history?symbol=^SP500-60

Params ● Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

	KEY	VALUE
<input checked="" type="checkbox"/>	symbol	^SP500-60
	Key	Value

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize Text 

```
1 Date,Open,High,Low,Close,Adj Close,Volume
2 2002-02-11,91.680000,92.589996,91.480003,92.480003,92.480003,0
3 2002-02-12,92.500000,92.540001,91.379997,91.610001,91.610001,0
4 2002-02-13,91.580002,91.989998,91.489998,91.989998,91.989998,0
5 2002-02-14,91.989998,93.000000,91.900002,92.470001,92.470001,0
6 2002-02-15,92.470001,93.559998,92.470001,93.330002,93.330002,0
7 2002-02-19,93.330002,93.599998,91.680000,91.680000,91.680000,0
8 2002-02-20,91.750000,92.900002,91.690002,92.900002,92.900002,0
9 2002-02-21,93.050003,93.239998,91.889999,91.989998,91.989998,0
10 2002-02-22,91.970001,93.709999,91.879997,93.459999,93.459999,0
11 2002-02-25,93.430000,94.050003,93.370003,93.860001,93.860001,0
12 2002-02-26,93.860001,94.050003,93.379997,93.480003,93.480003,0
13 2002-02-27,93.430000,94.050003,93.220001,93.650002,93.650002,0
14 2002-02-28,93.610001,94.120003,93.209999,93.309998,93.309998,0
15 2002-03-01,93.570000,94.169998,93.500000,94.139999,94.139999,0
16 2002-03-04,94.139999,95.309998,94.059998,95.250000,95.250000,0
```

## Appendix C: Example of CPI API

Figure 12. *Example of CPI API*

POST <https://api.bls.gov/publicAPI/v2/timeseries/data/?startyear=2000&endyear=2022>

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE
seriesid	CUSR0000SA0,CUSR0000SAC,CUSR0000SAH,CUSR0000SAI,CUSR0000SAS4

Body Cookies (1) Headers (13) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": "REQUEST_SUCCEEDED",
3   "responseTime": 478,
4   "message": [
5     "Year range has been reduced to the system-allowed limit of 10 years."
6   ],
7   "Results": {
8     "series": [
9       {
10        "seriesID": "CUSR0000SA0",
11        "data": [
12          {
13            "year": "2009",
14            "period": "M12",
15            "periodName": "December",
16            "value": "217.347",
17            "footnotes": [
18              {}
19            ]
20          },
21          {
22            "year": "2009",
23            "period": "M11",
24            "periodName": "November",
25            "value": "217.234",
26            "footnotes": [
27              {}
28            ]
29          },
30          {
31            "year": "2009",
32            "period": "M10",
33            "periodName": "October",
34            "value": "216.509",
35            "footnotes": [
36              {}
37            ]
38          }
39        ]
40      }
41    ]
42  }
```