Assignment 1 Report – DIC

Sai Swapnesh Pahi – UB Person # - 50610538

I used google colab for my work. I have my personal drive setup for the .ipynb file and downloaded the s&p500 data set and loaded the csv to my drive root directory. Mounted my google drive to the python environment, loaded all necessary python libraries and created a pandas data frame of the csv file.

Part 1: Big Data Processing

Task 1: I initially thought of normalizing the date column to pandas datetime object since the Date column was of type string. AS the task 1 suggests I printed out the meta data of the data frame:

```
1 # meta data of df from csv
2 print("Dimension of the data frame: ", df_normalized.shape)
3 df_normalized.info()
```

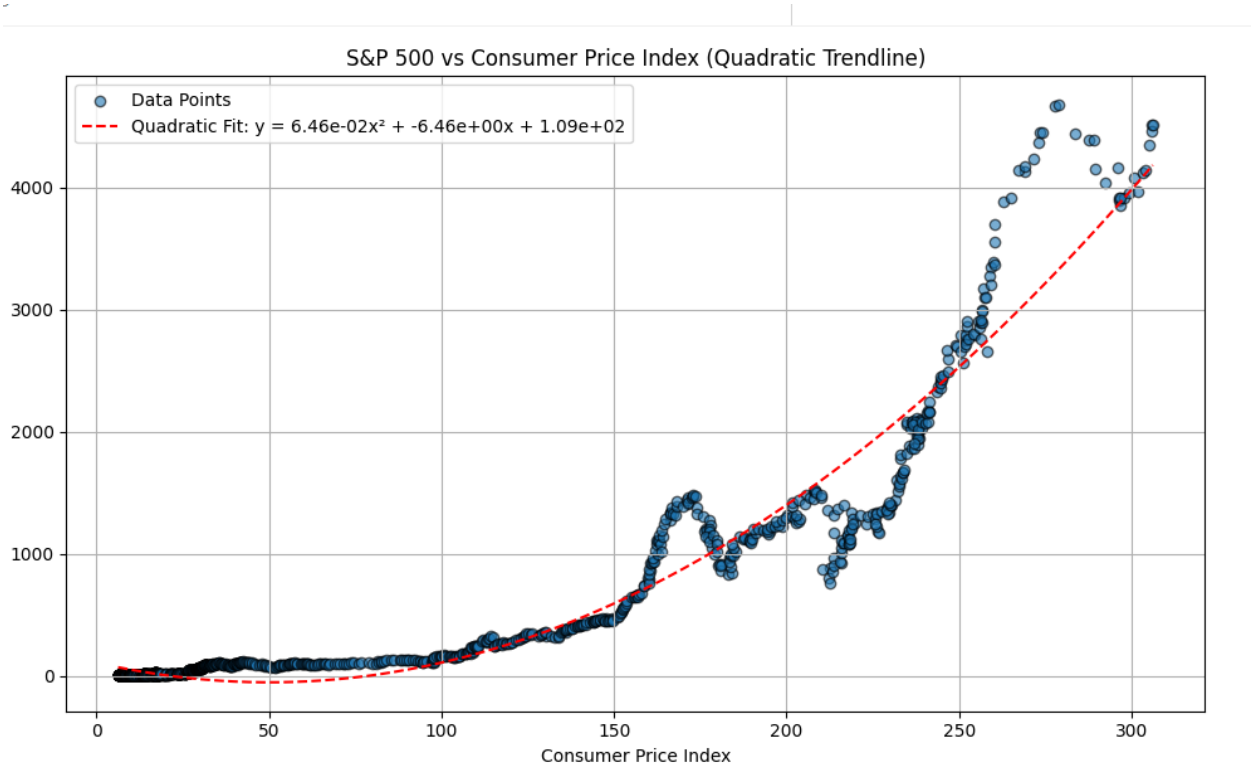
```
Dimension of the data frame:  (1833, 10)
<class 'pandas.core.frame.DataFrame'>
Index: 1833 entries, 1832 to 0
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Date                  1833 non-null   datetime64[ns]
 1   SP500                 1833 non-null   float64
 2   Dividend              1833 non-null   float64
 3   Earnings              1833 non-null   float64
 4   Consumer Price Index  1833 non-null   float64
 5   Long Interest Rate    1833 non-null   float64
 6   Real Price            1833 non-null   float64
 7   Real Dividend         1833 non-null   float64
 8   Real Earnings         1833 non-null   float64
 9   PE10                  1833 non-null   float64
dtypes: datetime64[ns](1), float64(9)
memory usage: 157.5 KB
```

Checked for duplication but there were 0 duplicate rows. Later checked for duplicate dates in the data. As we know, the S&P data for a particular month and year there can't be duplications. Upon finding there were exactly 2 duplicate rows.

Calculating the stats across the fields for these duplicate records there were a few significant differences in values like standard deviation, mean, minimum and max values it makes sense to take their weighted mean and calculate it and squash it as a single unique record, that's how I handled such records. Hence, 1833 rows reduced to 1681 rows.

There were no missing values across all the fields, so data imputation part was skipped. Next, I calculated the summary of statistics of all the fields of this cleaned data frame and recorded it as per the requirement of the assignment description.

For finding a relationship between S&P500 and CPI which are strongly correlated, I used a scatter plot which is the best way to visualize the relationship between these 2 variables.



Clearly evident that the trend is quadratic in nature. Going forward linear regression won't be a good performing model on this data for predictions.

Task 2: Feature Engineering

1. Added a new column by calculating the year-over-year percentage change in CPI. This I calculated by using (CPI in the next year – CPI in the previous year)/ CPI of the prev year * 100. I had to shift the rows by 11 steps to get to the CPI of the prev month of the previous year manually because pct_change() function was messing up the calculations. And reported the top 5 dates with highest year over year change in CPI.
2. Next, calculated the rolling average of Earnings for 12 months of a year and plotted it against the raw earnings of a year. The first 11 rows in Earnings_MA_12 will be NaN, which Matplotlib ignores when plotting. Plotting both raw earnings and moving average shows trend smoothing over time.
3. Normalized the S&P500 field and recorded the top 10 values and it makes sense that these values were the highest between 2021 and 2022 just before the recession.

Task 3: Data Visualization

Histogram of S&P500: The S&P 500 has increased significantly over time, but those higher values are less frequent. Data is not normally distributed so any statistical modeling should avoid normality assumptions. Long tail on the left hand side which means lower value SP were bought frequently in big numbers but in the latest era the value of S&P increased significantly.

Box Plot of PE10 ratio: Indicates that the central tendency for the market valuation is around 20. The box spans from about 15 to 25, representing the middle 50% of the PE10 values. Suggests that under normal market conditions, the S&P 500 traded within this valuation range many dots on the right represent outliers - unusually high PE10 values.

These reflect periods of very high valuation

Outliers suggest that there were months or years where the market was significantly more expensive than average. Longer whisker and numerous outliers on the right side indicate a right-skewed distribution.

The market tends to experience more frequent moderate valuations, with occasional high PE ratios. The market typically trades with a PE10 ratio between 15 and 25, but can experience spikes well above that, often during speculative bubbles or loose monetary periods.

Correlation Heatmap of Numerical features:

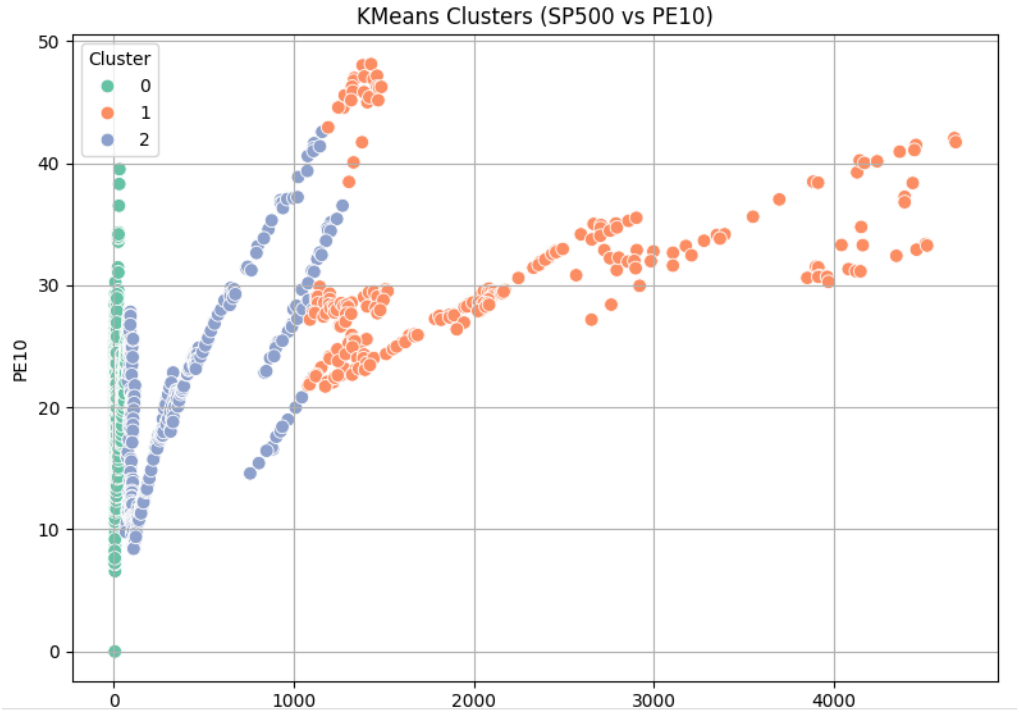
Most of these variables move together over time (highly correlated) because they're all derived from the same underlying data - SP500 index and company fundamentals.

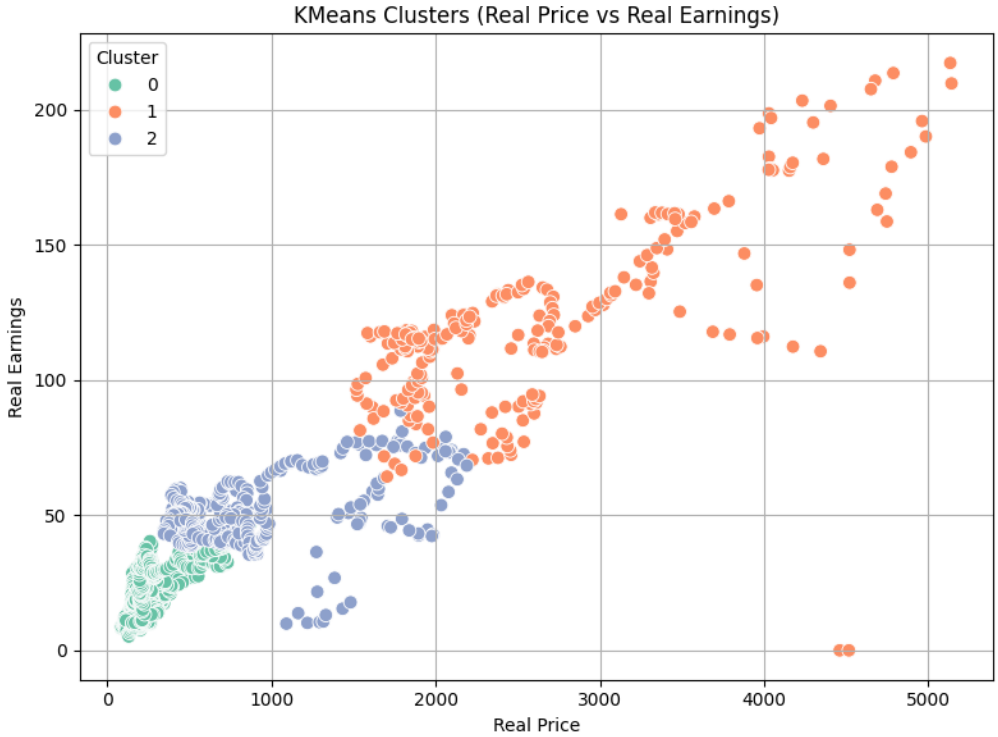
- **SP500 and Dividend** is highly correlated because as the stock value goes up the share value you hold will also go up.
- **SP500, Real Price, Dividend, Earnings, Real Dividend, Real Earnings**: All track corporate performance. If companies grow and profits rise, prices, dividends, and earnings all rise together
- **CPI, it is in general terms called cost of living**, Inflation pushes up nominal values (Dividends, Earnings, Price), so it's also strongly correlated with them. Higher the inflation, higher the stock price does not necessarily mean the same increase in percentage of earning.
- **Long Interest Rate**: This one is negatively correlated with stocks (-0.17). High interest rates generally hurt stock valuations (makes bonds more attractive vs. stocks)

- PE10 (price to earnings ratio over avg 10 years): It's not as highly correlated because it's a ratio. It compares prices to earnings. If both move together, the ratio doesn't change much, so correlations with other variables are weaker

Part 2: Machine Learning

1. Extracted all numerical features from the dataframe and scaled it. Used K Means clustering to analyze trends in S&P 500 and Earnings in unsupervised way. Calculated the Within Cluster Sum of Squares and plotted it against various values of K. Selected K=3 (Using elbow method) to keep the bias variance trade off low to avoid overfitting and underfitting.
    a. **For PE10 vs S&P500**:
        i. The data is split into 3 regimes of how the market was valued
        ii. Cluster 0 (green): Very early/low SP500 values with low PE10, early historical periods when both stock prices and valuations were small
        iii. Cluster 2 (blue): Mid-range SP500 and PE10, transitional periods with steady growth but valuations varied.
        iv. Cluster 1 (orange): Modern times, high SP500 values and higher PE10, recent decades when the stock market is much larger, and valuations (PE ratio) often higher.
    b. **For Real Earnings vs Real Price**:
        i. Cluster 0 (green): Early period - both stock prices and earnings were low in absolute terms.
        ii. Cluster 2 (blue): Mid period - moderate prices and earnings, a transition zone.
        iii. Cluster 1 (orange): Modern period - very high real prices and earnings, representing today's economy with huge corporations and high stock valuations.





2. **S&P Price Prediction (Linear Regression):** Extracted the numerical features and these will be our predictors for S&P. Performed 80-20 train test split on data and trained the model. Upon using this model on our testing data we found MSE: 4701.61, R^2: 0.99 which suggests model is explaining 99% of the variance in the

target, with loss of 4701.61 from true y values. Which is pointing towards overfitting as we have limited data.

3. Used the same data to train and ensemble of models (**random forest**) MSE: 473.81, R^2: 1.00. MSE = 473.81 -> The average squared prediction error is small. Model is accurate on the dataset it was evaluated on. R^2 = 1.00 -> The model explains 100% of the variance in SP500. But R^2 = 1.00 is too perfect, and signals possible overfitting or data leakage
4. **Consumer Price Index Trend Classification**: Converted the CPI field from numerical to categorical value by comparing the CPI value of current month with the CPI of the next month. (0 means CPI value won't go up the next month 1 CPI goes up) and used all numerical predictors in **logistic regression** model to train the model and resulted in 67% prediction accuracy. From the confusion matrix:
   (True Positives) = 121 Correctly predicted CPI will go up
   (True Negatives) = 104 Correctly predicted CPI will not go up
   (False Positives) = 48 Predicted up, but CPI didn't go up
   (False Negatives) = 63 Predicted not up, but CPI went up
5. As we know CPI is nothing but the cost of living so can't leave out the '**Date**' column for predictions. I converted it to an ordinal value (year * 12 + month) and used it as a numerical variable to club it with the other predictors. For using the **SVM model**, we first need to scale our input predictors as it is sensitive to it and then use it to train the model and we got stats as follows: 67% of total predictions were correct.
   Class 0: High recall (0.79), it means model is good at catching when CPI doesn't rise
   Class 1: Higher precision (0.77) - when it predicts CPI rise, it's right 77% of the time
   F1-scores ~ 0.65-0.68 Indicates good balance between precision and recall for both classes
6. Using **K nearest Neighbors**, KNN correctly classifies 64% of cases. Decent performance for a simple model. Model is fairly reliable in predicting CPI increases.
7. **Neural Networks**, with hidden layers 64, 32 Neural Net is on par with Logistic Regression and SVM in this case with 67% accuracy on CPI classification.
   Performance gains are marginal because:
   Dataset is not large or complex enough
   Neural nets are harder to tune, especially on small datasets
   It's a good choice, but Logistic Regression or SVM is better for interpretability and simplicity.