Consumer Price Index (CPI) Predictor

Executive Summary

This project is my first attempt in performing a time series project to model and forecast the Consumer Price Index (CPI) using the ARIMA and Prophet models. The analysis utilizes historical US CPI data from a Kaggle dataset which has records from 1913 to 2021, applying statistical time series techniques to identify patterns, achieve stationarity, and build a predictive model for inflation forecasting.

1. Introduction

1.1 Background

The Consumer Price Index (CPI) serves as a primary measure of inflation, tracking changes in the cost of goods and services over time. CPI is an indicator for inflation as its the average price over time for a basket of goods that consumers have to purchase in the US.

Accurate CPI forecasting is crucial for:

- Federal Reserve monetary policy decisions
- Economic planning and budget forecasting
- Business investment strategies
- Public financial planning and cost-of-living adjustments

1.2 Project Objectives

- Analyze historical CPI trends and patterns
- Develop a robust ARIMA time series model and Prophet for CPI prediction
- Assess model performance and statistical significance
- Provide insights into CPI behavior and forecasting capabilities

1.3 Methodology Overview

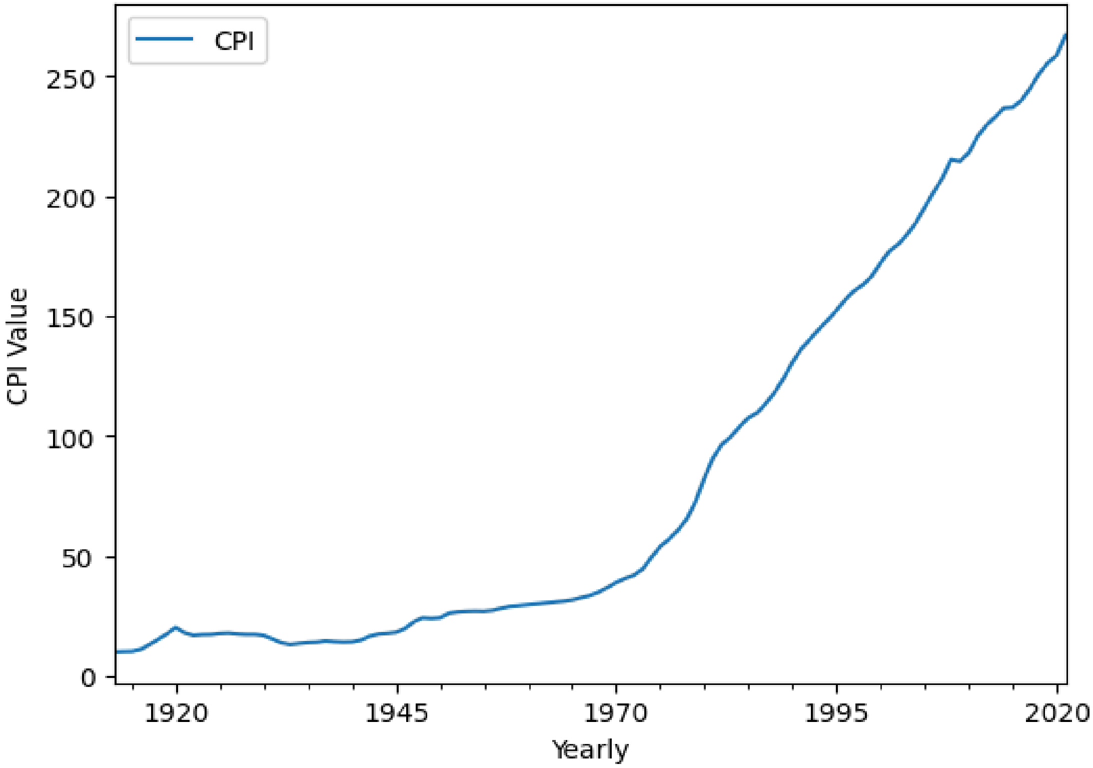
The project employs classical time series analysis techniques:

- Exploratory data analysis and visualization
- Stationarity testing and transformation
- Autocorrelation and partial autocorrelation analysis
- ARIMA model specification and estimation
- Model validation and diagnostics

2. Exploratory Data Analysis

2.1 Time Series Visualization

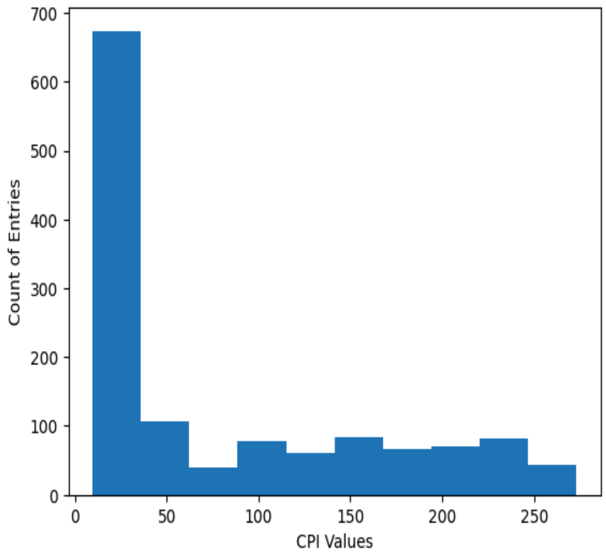
We plot both the monthly CPI and yearly CPI to check for any general trends in CPI values. The yearly aggregation shows clear long-term inflationary trends with intermittent periods of reduced growth.

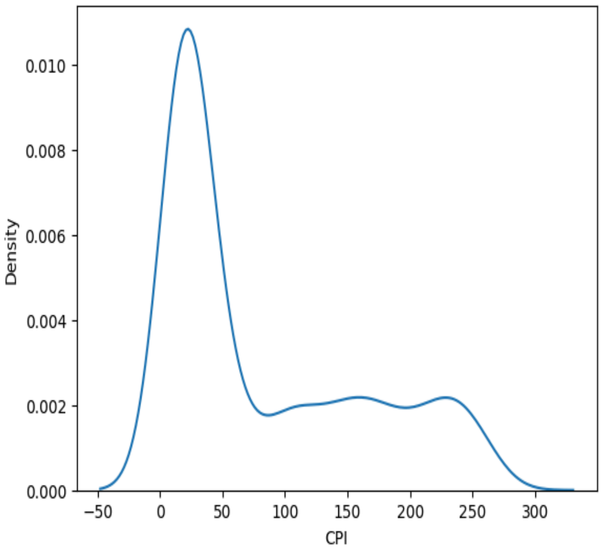


2.2 Distribution Analysis

2.2.1 CPI Value Distribution

The histogram reveals concentration of data points toward lower CPI values, indicating the historical nature of the dataset with more observations from earlier periods when CPI levels were lower. The kernel density plot indicates the probability distribution confirms a left-skewed distribution with lower values being more probable, suggesting the presence of outliers and indicating that transformation may be necessary for modeling. Ideally, we want to avoid any bias towards particular values to ensure a better performing model.

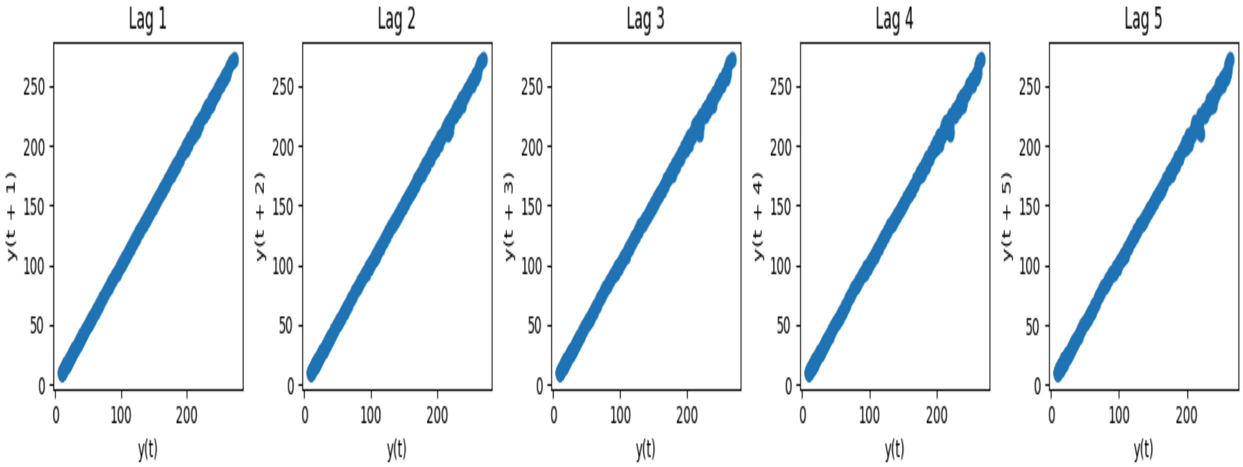




3. Time Series Analysis and Stationarity Testing

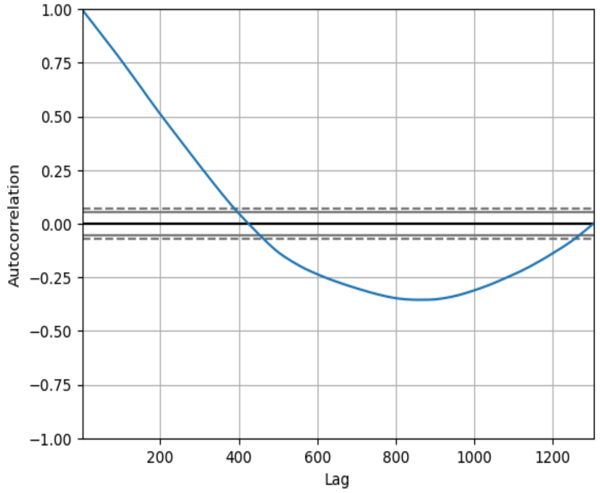
3.1 Lag Plot Analysis

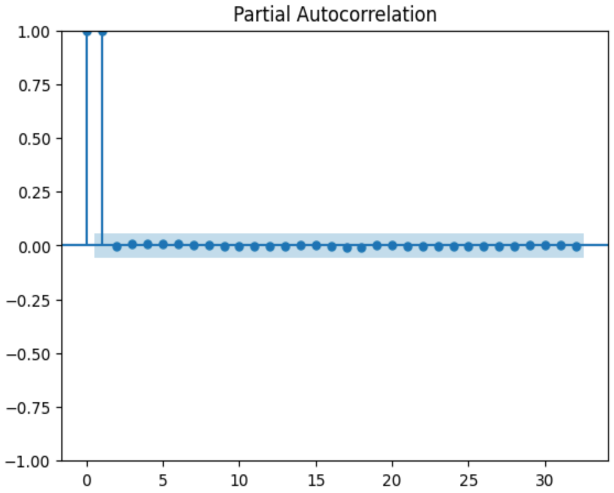
Based on plotting CPI values compared to one step behind in time (lag-1), there seems to be high correlation in current and past CPI values. We try to plot across different lags and see that correlation waivers as lags increase. This indicates that the most recent CPI values are most correlated with future ones and so some form of transformation so our ARIMA can benefit greatly by placing more weight to recent CPI values.



3.2 Autocorrelation and Partial Autocorrelation Analysis

The ACF (Autocorrelation function) plot is meant to show the significance of current CPI values across different lags. There are no clear seasonal patterns due to no consistent positive/negative values. There also is not any consistent correlation value across lags indicating that the data is not stationary (having a constant mean and variance across periods for better modeling). The effect of previous CPI seems to continue across different periods as the rate of decrease in correlation is low and therefore there is strong persistence.





To explore further we plot a Partial Autocorrelation Function (PACF) plot which takes out the effect of intermediate values on the current CPI, therefore the correlation between present CPI and lag 4 would not have the effect of lag 1 to 3. The PACF indicates strong correlation with the first few lags, with higher-order lags showing minimal direct influence which confirms our hypothesis of more recent CPI values being strong predictors of current CPI.

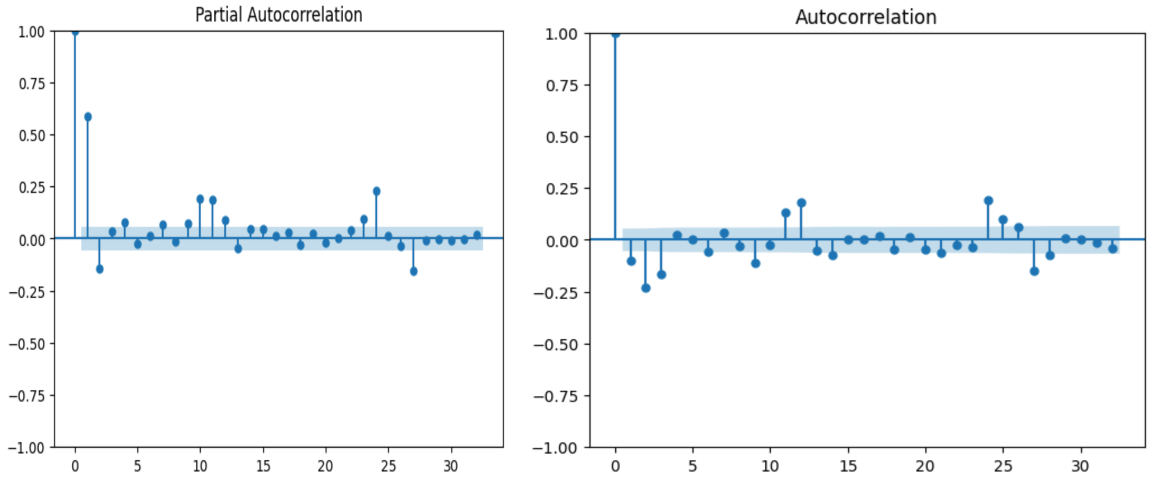
3.3 Augmented Dickey-Fuller (ADF) Test (Original Series)

We perform the ADF test on the base dataset to check for stationarity. The ADF test is meant to indicate whether a dataset follows a “random-walk” pattern (non-stationary) or goes back to a mean value across periods. If the test returns a p-value value of less than 0.05, we reject the null hypothesis to indicate the series is likely stationary. However, in this instance we had a value of 1.00 so the series is not stationary and differencing is required.

4. Data Transformation for Stationarity

4.1 Differencing

Differencing is how many times to take difference between consecutive values to represent the series. The differenced series contains values that represent the change in value from previous ones. This is meant to remove trends and have a more stationary dataset. We did both first order and second order differencing to reach an ADF acceptable score of 0.228.



PACF and ACF plot after second order differencing showing relatively closer correlation values across lags compared to before.

4.2 Model Parameter Identification

Based on ACF and PACF analysis of the second-differenced series:

- **AR Component (p)**: PACF suggests significant lags up to 2
- **Differencing Order (d)**: 2 (determined through stationarity testing)
- **MA Component (q)**: 0 (We see that the ACF plot does not have as consistent correlation values)

5. Model Development and Training

5.1 Data Splitting Strategy

We create training and test sets with each data-point being consecutive instead of random. This is to maintain the relationship of our target variable with time. We have 1,273 observations as our training and 30 observations as our test.

Training/Test Split: 97.7% / 2.3% split optimized for time series validation

We then train an ARIMA model with the parameters that we had identified on the train dataset to get the following parameter estimates across lags.

Autoregressive Components

Parameter	Coefficient	Std Error	Z-Statistic	P-Value	95% Confidence Interval
AR(1)	-0.1544	0.013	-11.685	0.000	[-0.180, -0.128]
AR(2)	-0.2392	0.013	-18.642	0.000	[-0.264, -0.214]

Error Variance

Parameter	Coefficient	Std Error	Z-Statistic	P-Value	95% Confidence Interval
σ^2	0.1394	0.002	59.721	0.000	[0.135, 0.144]

All parameters show statistical significance with p values being less than 0.001

The sigma2 being low (around 0.144) also indicates that the model does well in capturing the variance of the dataset itself.

5.3 Model Fit Quality

The following are the results we gain from fitting our model:

- **AIC**: 1108.367 (Lower values indicate better fit)
- **BIC**: 1123.810 (Balances fit quality with model complexity)
- **Log Likelihood**: -551.184
- **Statistical Significance**: All parameters except ma.L1 are highly significant ($p < 0.001$)

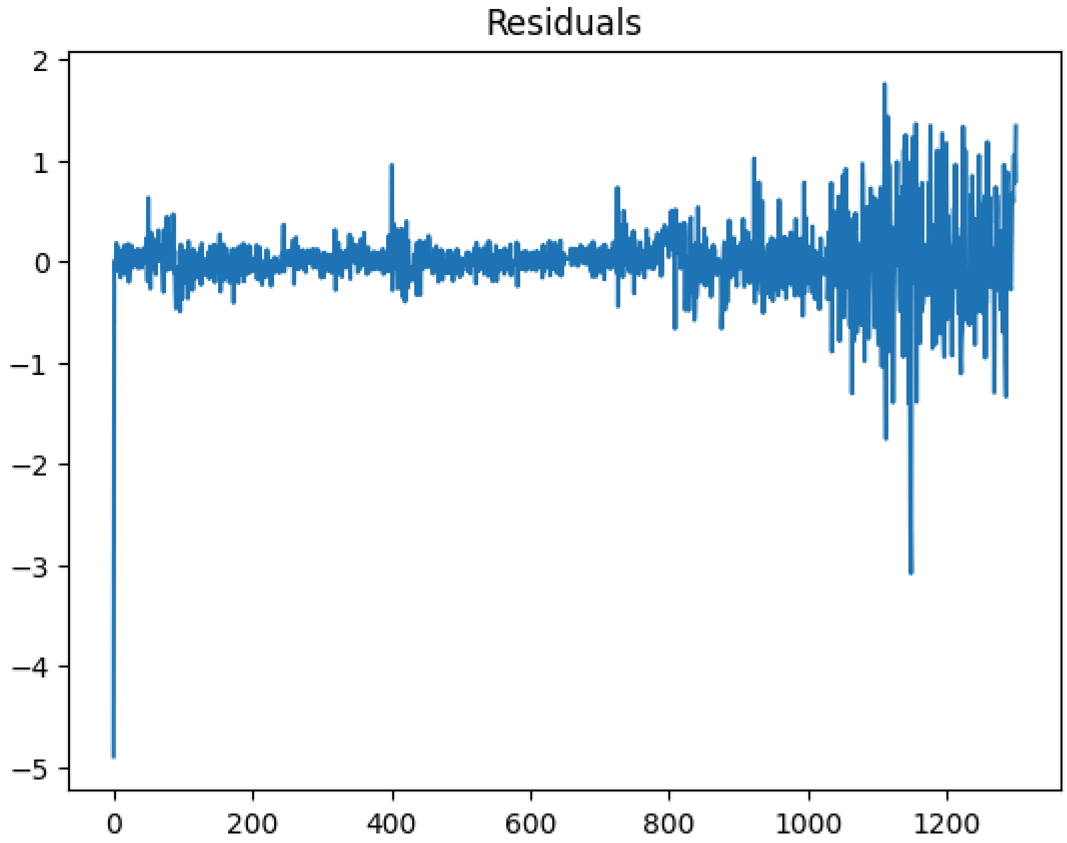
Here’s what these metrics indicate in full form:

- **Akaike Information Criterion (AIC):** Used to measure the quality of our model by taking into account how well it fits the data against the complexity of our model. Lower values are better. This is useful to penalize a model if it overfits to the data with too many parameters. Formula: 2*k-2*ln(L) where k is the number of hyper parameters, L is the likelihood.

- **Bayesian Information Criterion (BIC):** Similar to AIC but adds a stronger penalty to more complex models. Formula: k*ln(n)-2*ln(L) where n=sample size.
- **Log Likelihood:** Meant to indicate how well the model explains the data. Higher values are better and are used in the calculation of AIC and BIC.

---

6. Model Diagnostics and Validation



6.1 Residual Analysis

6.1.1 Ljung-Box Test

- **Test Statistic**: 3.95
- **P-value**: 0.05
- **Interpretation**: Residuals show significant autocorrelation (based on test statistic being higher than p-value)

6.1.2 Jarque-Bera Normality Test

- **Test Statistic**: 6295.11
- **P-value**: 0.00
- **Interpretation**: Residuals are not normally distributed, indicating potential outliers

6.2 Heteroskedasticity Testing

- **Test Statistic**: 15.22
- **P-value**: 0.00 (two-sided)
- **Interpretation**: Evidence of heteroskedasticity (non-constant variance), forecast predictions may be unreliable in some periods, too narrow/wide in some periods

6.3 Distribution Properties

- **Skewness**: 0.17 (a bit right skewed, indicating residuals are bit higher)
- **Kurtosis**: 13.90 (Heavy-tailed distribution, there are more outliers with high values)

---

7. ARIMA Model Forecasting

7.1 ARIMA Predictions

The ARIMA(2,2,5) model generates predictions for the 30-observation test set, providing point forecasts based on the trained model parameters.

When testing our model we get the following results:

- Mean Absolute Error (MAE): 0.471 which is the average absolute difference between predictions and true values
- Mean Squared Error (RME): 0.563 which is the average of taking the square difference between predictions and true values

---

8. Prophet Model Implementation and Comparison

8.1 Data Preparation for Prophet

Prophet is a model that has been developed by META to breakdown time series into the following three components:

- Trend: Long term increasing/decreasing pattern
- Seasonality: Regular patterns daily/weekly/monthly
- Holidays: If special events affect the model performance

The data was reformatted for Facebook Prophet modeling:

- **ds**: Date column (Prophet requirement)
- **y**: Target variable (second-differenced CPI values)
- **Data Split**: Same train/test split as ARIMA model

8.2 Prophet Hyperparameter Tuning

8.2.1 Parameter Grid Definition

Parameter Descriptions:

- **changepoint_prior_scale**: Controls flexibility in trend changes (0.001 to 0.5). Indicates how strong the condition for a trend to be when considering certain times to be a changepoint.
- **seasonality_prior_scale**: Controls strength of seasonal components (0.01 to 10.0). Indicates how much to consider seasonal pattern when making predictions, higher means higher change is considered, lower means not that much change from seasons is considered).

8.2.2 Cross-Validation Results

Index	changepoint_prior_scale	seasonality_prior_scale	RMSE
0	0.001	0.01	0.200254
1	0.001	0.10	0.202807
2	0.001	1.00	0.201881
3	0.001	10.00	0.201452
4	0.010	0.01	0.198938
5	0.010	0.10	0.200704
6	0.010	1.00	0.200848

7	0.010	10.00	0.200470
8	0.100	0.01	0.198518
9	0.100	0.10	0.200222
10	0.100	1.00	0.200132
11	0.100	10.00	0.199976
12	**0.500**	**0.01**	**0.198420**
13	0.500	0.10	0.200107
14	0.500	1.00	0.199704
15	0.500	10.00	0.199628

Optimal Parameters:

- **changepoint_prior_scale**: 0.500
- **seasonality_prior_scale**: 0.01
- **Best CV RMSE**: 0.198420

8.3 Prophet Model Training and Forecasting

The Prophet model decomposes the forecast into:

- **Trend**: Overall directional movement (0.211731 for final periods)
- **Weekly Seasonality**: Day-of-week effects (-0.042090 to 0.041835)
- **Yearly Seasonality**: Annual seasonal patterns (-4.448711 to 0.008099)
- **Combined Forecast (yhat)**: Trend + seasonal components

Our model gives us the following results against test data which is lower than that of our ARIMA model. However, even though the results are worse off the model is able to provide us more interpretability in overall trend along with seasonality that we might need to account for.
MAE: 1.675839

RMSE: 2.227205

9 Data Sources

1. **Kaggle Dataset**: US Inflation Data (Updated till May 2021)
    - URL: /kaggle/input/us-inflation-data-updated-till-may-2021/US CPI.csv
    - Coverage: Historical monthly CPI data through May 2021