REGRESSION AND TIME SERIES MODELING

MA60280

TERM PROJECT

WEB APP FOR TIME SERIES FORECASTING

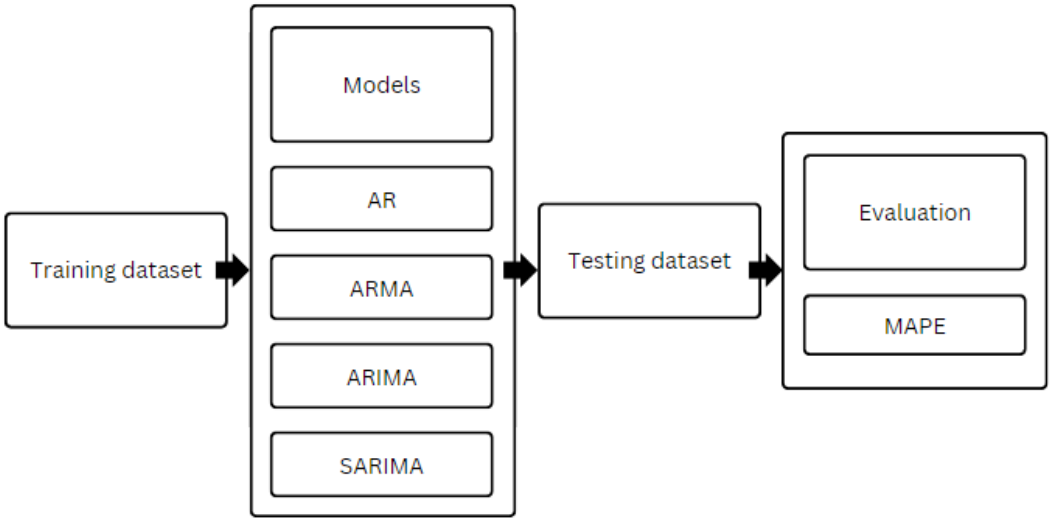
TEAM

1. Aditya Nandy (21EE3FP52)
2. Jagori Bandyopadhyay (21MT3FP40)
3. Nikhil S Kowshik (21MT3FP30)
4. Priyanshi Khetan (21CE3FP46)
5. Shahil Gupta (21ME3FP06)
6. Tarang C Rangani (21EE3FP19)

Report contents

Project Overview

Building a web app that can determine the best-suited Time series forecasting model for any given dataset



Website link: https://rtsmproject-s79ta4yr5yujjpxeaizrrv.streamlit.app/

Time Series Forecasting

Time series forecasting is the process of analyzing time series data using statistics and modeling to make predictions and inform strategic decision-making. It's not always an exact prediction, and the likelihood of forecasts can vary wildly—especially when dealing with the commonly fluctuating variables in time series data as well as factors outside our control. However, forecasting insight about which outcomes are more likely—or less likely—to occur

than other potential outcomes. Often, the more comprehensive the data we have, the more accurate the forecasts can be. While forecasting and “prediction” generally mean the same thing, there is a notable distinction. In some industries, forecasting might refer to data at a specific future point in time, while prediction refers to future data in general.

Models

Autoregressive Model (AR)

Autoregressive models operate under the premise that past values have an effect on current values, which makes the statistical technique popular for analyzing nature, economics, and other processes that vary over time.

The notation AR(*p*) indicates an autoregressive model of order *p*. The AR(*p*) model is defined as

$$X_t = \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

where **φ** are the *parameters* of the model and **ϵ** is white noise. This can be equivalently written using the backshift operator *B* as:

$$X_t = \sum_{i=1}^{p} \varphi_i B^i X_t + \varepsilon_t$$

An AR(1) autoregressive process is one in which the current value is based on the immediately preceding value, while an AR(2) process is one in which the current value is based on the previous two values. An AR(0) process is used for white noise and has no dependence between the terms.

ARMA (Autoregressive Moving Average) Model:

ARMA models combine autoregressive (AR) and moving average (MA) components. They consist of a constant, AR lags with their multipliers, MA lags with their multipliers, and white noise. The formula for an ARMA(p, q) model is:

$$y_t = c + \phi_1 y_{t-1} + \theta_1 \epsilon_{t-1} + \epsilon_t$$

where y_t and y_t-1 represent the values in the current period and 1 period ago, respectively. ϵ_t and ϵ_t-1 are the error terms for the same two periods, and c is a constant factor. The parameters ϕ_1 and θ_1 express the importance of the values and error terms for the "i-th" lag.

ARIMA (Autoregressive Integrated Moving Average) Model:

ARIMA models include a preprocessing step to make the data stationary, denoted by I(d), where d is the difference order. The formula for an ARIMA(p, d, q) model is:

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p)(1 - B)^d X_t = (1 + \theta_1 B + \theta_2 B^2 + ... + \theta_q B^q)\epsilon_t$$

where:
Xt is the time series at time,ϕ1,ϕ2,...,ϕp are the autoregressive parameters. θ1, θ2,...,θq are the moving average parameters. d is the order of differencing. ϵt is white noise with zero mean and constant variance.

SARIMA (Seasonal Autoregressive Integrated Moving Average) Model:

SARIMA models are a seasonal version of ARIMA models. They incorporate additional autoregressive and moving average components to account for seasonality. The formula for a SARIMA(p, d, q)(P, D, Q)m model is:

$$(1 - \phi_1 B - \phi_2 B^2 - ... - \phi_p B^p)(1 - \Phi_1 B^s - \Phi_2 B^{2s} - ... - \Phi_P B^{Ps})(1 - B)^d(1 - B^s)^D X_t =$$

$$(1 + \theta_1 B + \theta_2 B^2 + ... + \theta_q B^q)(1 + \Theta_1 B^s + \Theta_2 B^{2s} + ... + \Theta_Q B^{Qs})\epsilon_t$$

where y_t is the value in the current period, y_t-1, y_t-2, ..., y_t-p are the past values, ϵ_t-1, ϵ_t-2, ..., ϵ_t-q are the past error terms, and c is a constant factor. The parameters ϕ_(P, D, Q) and θ_(P, D, Q) express the importance of the values and error terms for the seasonal lag, and m specifies the number of time steps for a single seasonal period.

The major differences between these models are the presence of differencing (I) and seasonal components (S). ARMA models focus on past values and past errors, ARIMA models add trend differencing, and SARIMA models incorporate seasonal differencing. SARIMA models also have additional hyperparameters to specify the autoregression (AR), differencing (I), and moving average (MA) for the seasonal component of the series

Dataset

The datasets which we are using are:

- Airplane travel dataset: In this dataset, the number of people traveling by airplanes over the years is plotted against the axis of years is plotted
- Shampoo sales dataset: The number of shampoos sold in a month against the x axis of months is plotted
- Stock Market dataset: The value of price of the stock of a company is plotted against time

Evaluation Method

Mean absolute percentage error

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD), is a measure of prediction accuracy of a forecasting method in statistics. It usually expresses the accuracy as a ratio defined by the formula:

$$\mathrm{MAPE} = 100\frac{1}{n}\sum_{t=1}^{n}\left|\frac{A_t - F_t}{A_t}\right|$$

where At is the actual value and Ft is the forecast value. Their difference is divided by the actual value At. The absolute value of this ratio is summed for every forecasted point in time and divided by the number of fitted points n.

Methodology

We implement the AR and ARMA from scratch. We use the ARIMA and SARIMAX models as implemented in the statsmodels library.

Autoregressive (AR) implementation

Firstly, we will use SciKitLearn's Linear Regression. This is because we have established that this model is just a linear regression and SciKitlearn is extremely stable to implement the

following linear regression. The AR algorithm determines the linear regression of (Present fitted values) vs. (Past fitted values). We even plot the PACF values for the error term which helps us weed out the factors which do not significantly contribute to the current value. Based on the filtered lag values, we eventually train the model by using model.fit(). That's the basic implementation of the AR model which is done from scratch. We have even included a slider to get the number of lag values to our wish.

ARMA implementation:

The implementation of the ARMA model has the two separate curves where we perform linear regression on. It even takes the MA (Moving average) part into account which is easily implementable using the rolling window technique with a user adjustable window size. It has the same implementation of the AR model as mentioned previously and has two regressions implemented on two separate curves, the trend curve (which is more or less linear) and the deviation part, and we have used ScikitLearn for both of these curves.

ARIMA implementation:

We implement the ARIMA method in the following way: We start by importing necessary libraries, such as numpy, pandas and components from statsmodels like 'ARIMA'. Next we upload the time series and plot it using matplotlib to check for trends and seasonality. To identify the ARIMA model parameters (p, d, q), we can use the auto ARIMA method to get the best values. But, in this app we have included a slider that allows us to choose the values of (p, d, q) and accordingly check the results in the graphs and the MAPE values.

SARIMAX implementation:

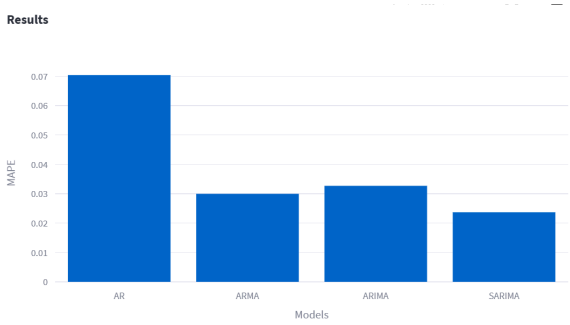
SARIMAX is implemented the same way as ARIMA, but the only difference is that we use the SARIMAX implementation of the model on statsmodel.

The entire app has been implemented using streamlit and the Plotly library to plot the PACF values. The app can be accessed using this link:
https://rtsmproject-s79ta4yr5yujjpxeaizrrv.streamlit.app/

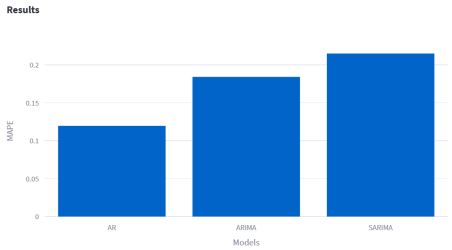
Results

The results were explicitly mentioned in the video where we gave a live demonstration of the app. But we are presenting the MAPE values of the three datasets in the report. SARIMAX and ARIMA perform the best.

Passenger Dataset



Shampoo Dataset



Stock Market Dataset