List of Lab programs

Week	Class	Unit	List of Programs Demonstrate the following with a suitable time series dataset
4	4	**2**	i. Different forecasting techniques like Simple Exponential Smoothing (SES), Simple Moving Average (SMA) & Holt-Winters Smoothing. ii. Calculate the evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) for each forecasting technique. iii. Identify the trends and seasonal patterns for the above forecasting techniques for the given dataset.
5	5	**3**	i. Generate a sequence of white noise data and visualize it. ii. Compare the graphs of both White Noise and the time series data. iii. Use statistical tests such as the Augmented Dickey-Fuller Test and the Kwiatkowski–Phillips–Schmidt–Shin test to determine the presence of stationarity.
6	6	**3**	i. Use statistical techniques such as moving averages to detect and quantify trends in the data. ii. Plot the ACF and PACF for a given time series dataset using Python libraries such as statsmodels and analyze the data to identify any underlying patterns such as seasonality or long-term trends.
7	7	**3**	i. Examine the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots to determine the order of the AR model. ii. Fit an AR(1) model to the dataset and evaluate its performance. iii. Explore the possibility of fitting higher lag AR models to potentially capture more complex dependencies in the dataset.
8	8	**3**	i. Plot the ACF and PACF ii. Fit an MA(1) model iii. Fit higher lag MA model iv. Compare the performances of MA(1) and higher lag MA model.
9	9	**3**	i. Initialize the ARMA ii. Train the model on the dataset using the fit() method. iii. Generate forecasts by utilizing the predict() function and designating the desired time index or indices.
10	10	**3**	i. Initialize the ARIMA model by invoking ARIMA() and specifying the p, d, and q parameters (p is the number of autoregressive terms, d is the number of nonseasonal differences needed for stationarity and q is the number of lagged forecast errors). ii. Train the model on your dataset using the fit() method. iii. Generate forecasts by utilizing the predict() function and designating the desired time index or indices.