Predicting and Forecasting Daily Temperature for Bike Sharing Service in Washington D.C.

LEE YONG MENG (P-COM0012/20)
School of Computer Sciences, USM | Semester II, 2020/2021 | CDS513: Predictive Business Analytics [Assignment 2]

Intro + Problem Background

Bike sharing service is a facility provided to the community in a city to enable bike renting for short-term usage. It involves the process of renting bike from one kiosk and then returning it at different kiosk of the same system. The benefits of having bike sharing service are multifold. It not only promotes green lifestyle by reducing the usage of motor vehicles by citizens to move around a city, but also enables researchers to study the mobility of citizens in a city through the data collected by the service provider.

However, just like any other outdoor activities, the demand on the bike rentals is highly dependent on the weather conditions. For example, this demand during hot summer days is expected to be different from a snowing winter season.

Among different weather conditions, temperature data shows the highest correlation to the number of bike rentals in the bike sharing demand data in Washington D.C..

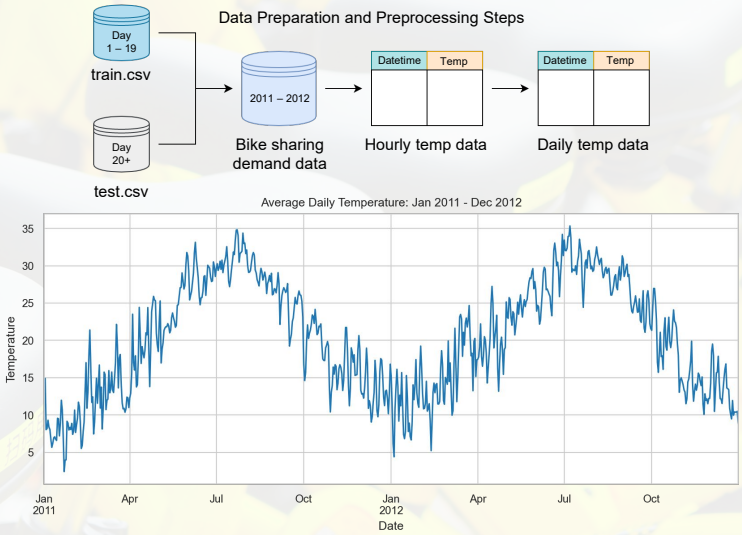
Correlation between number of bike rentals with different weather conditions (from bike sharing demand data)

Temperature	Windspeed	Humidity
0.3945 (most correlated)	0.1014	-0.3174

Therefore, in this assignment, the main objective is to **apply the knowledge of time series analysis and forecasting** to study daily temperature in Washington D.C. as time series data. Another objective is to implement and compare different **ARIMA and Machine Learning models** in predicting and forecasting the daily temperature in Washington D.C..

Daily Temperature as Time Series Data

Dataset: Bike sharing demand data[1]



1. Bike sharing demand data: https://www.kaggle.com/c/bike-sharing-demand/data

What is the data about?

Hourly records of weather conditions and number of bike rentals from Jan 2011 to Dec 2012 in Washington D.C..

- "train.csv": records of day 1 – 19 of each month.
- "test.csv": records of day 20+ of each month.

Hourly temperature data (selected time series data) is aggregated into **average daily temperature data**. The time series graph is plotted on the left.

Identifying **Time Series components** from the plot:

- **Trend**: A very slight increasing trend from 2011 – 2012 (by observing the January data).
- **Cyclical**: Yearly seasonality patterns repeats every 12 months.
- **Seasonality**: Yearly seasonality: peak at July and lowest at January + weekly seasonality.
- **Irregular**: Random noise from unknown source exists in the data.

Differencing Daily Temperature

2. ADF test: Augmented Dickey-Fuller test.
   - Null hypothesis, H_0: There is a unit root in the time series data – the series is not trend-stationary.
   - Alternative hypothesis, H_a: The time series data is trend-stationary.

Performing ADF test[2]

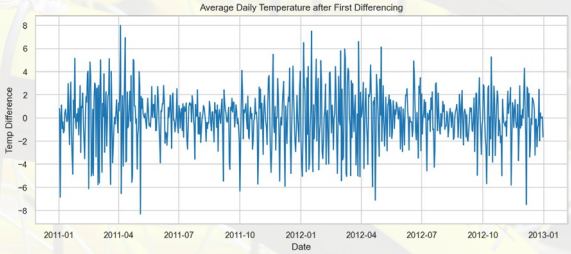
p-value **0.3016** (≥0.05)

We fail to reject H_0, implying that the time series data is not trend-stationary.

Therefore, **differencing is required**.

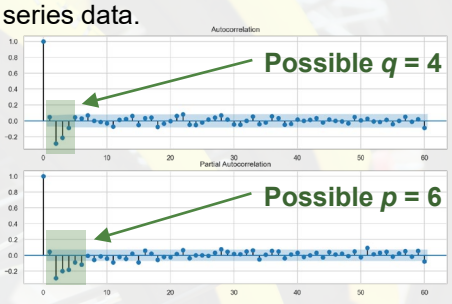
First Differencing: Find the changes in average daily temperature between current and previous timestep, t – achieved through Numpy's `diff()` function in Python.

$$\Delta x_t = x_t - x_{t-1}$$



Performing ADF test on first differences: ***p*-value = 4.1169 × 10^{-14}** (first difference is enough) **(<0.05)** **d = 1**

ACF and PACF plots: Used to identify the order of autoregressive and moving average to build ARIMA model, p and q, respectively after differencing time series data.

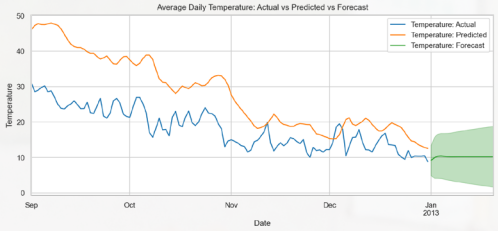


Time Series Forecasting: ARIMA vs Machine Learning[3]

3. To train Machine Learning models, the prediction and forecasting tasks are converted into regression tasks. As such, windowing techniques are used to generate two new features from the time series data: 7-day and 30-day rolling window.

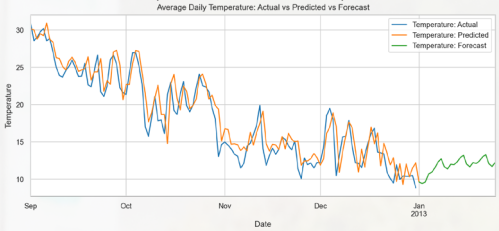
Different ARIMA and Machine Learning models are built to predict and forecast the daily temperature in Washington D.C.. The actual and predicted (Sep – Dec 2012), and forecasted temperature (10 – 24 days since 1 Jan 2013) for the optimal ARIMA, SARIMAX and Machine Learning models are plotted and displayed below for comparison.

ARIMA Model
(Autoregressive Integrated Moving Average)



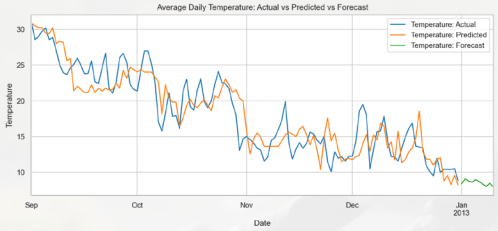
Optimal ARIMA model:
ARIMA(2,1,1)
where $(p,d,q) = (2,1,1)$

SARIMAX Model
(Seasonal Autoregressive Integrated Moving Average Exogenous)
(variation of ARIMA model)



Optimal SARIMAX model:
SARIMAX(2,0,1)(0,1,1,7)
where $(p,d,q)(P,D,Q,S) = (2,0,1)(0,1,1,7)$

Machine Learning Model
(Bagging Regressor)



Optimal Machine Learning model:
Bagging Regressor
where base estimator = Decision Tree Regressor; number of estimators = 10

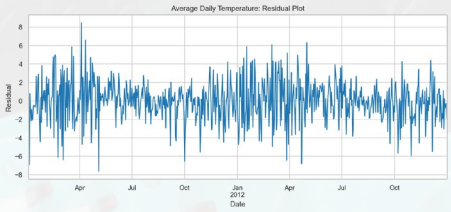
Auto ARIMA in Python: *Used to identify the optimal orders for ARIMA and SARIMAX models respectively based on the AIC values of different order combinations.*

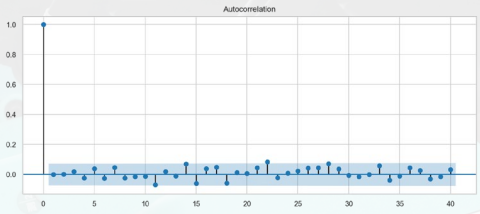
From the plots, ARIMA model performs the worst in predicting and forecasting daily temperature in Washington D.C., whereas SARIMAX model performs the best in these tasks. Machine Learning model is performing fairly good.

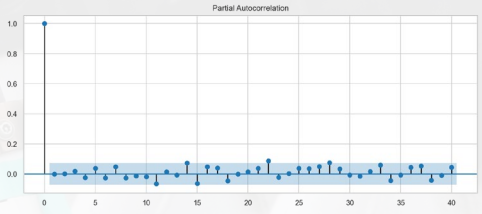
Further analysis is presented in "Result and Analysis" below.

ACF and PACF for Residuals of ARIMA Models

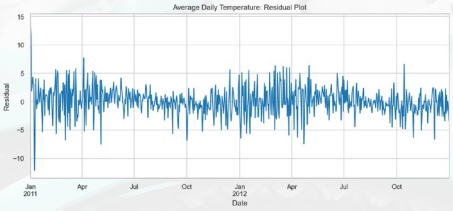
ARIMA(2,1,1): The plots of the residual, ACF and PACF.

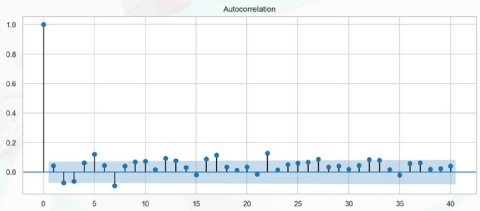


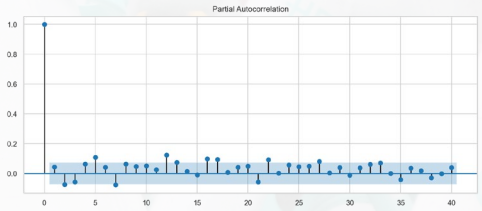




SARIMAX(2,0,1)(0,1,1,7): The plots of the residual, ACF and PACF.







Different plots show that there are no more information left for extraction by these two ARIMA models. The residuals are stationary, all ACF and PACF values stay within/very close to the confidence interval (shaded region).

Result and Analysis

AIC and BIC for both ARIMA(2,1,1) and SARIMAX(2,0,1)(0,1,1,7) are **optimal** among all combinations of orders in ARIMA and SARIMAX respectively. These values are close between these two ARIMA models, but yield **very different performance levels** on the time series data.

ARIMA model	**AIC**	**BIC**	**MAE**	**MRE**	**MSE**	**RMSE**
ARIMA(2,1,1)	**3215.991**	**3238.957**	9.9245	0.5410	132.1387	11.4952
SARIMAX(2,0,1)(0,1,1,7)	3266.043	3288.905	**1.6162**	**0.1038**	**4.4205**	**2.1025**

Machine Learning model	**MAE**	**MRE**	**MSE**	**RMSE**
XGBoost Regressor (learning rate: 0.01)	6.3457	0.3389	48.5666	6.9690
XGBoost Regressor (learning rate: 0.1)	2.3432	0.1419	8.9168	2.9861
XGBoost Regressor (learning rate: 0.3)	2.5049	0.1489	9.5917	3.0971
DT (Decision Tree) Regressor	2.9011	0.1679	13.0011	3.6057
Bagging Regressor (base estimator: DT Regressor; # estimator: 10)	**2.0862**	**0.1251**	**6.6557**	**2.5799**

Best performing **Machine Learning** model.

Best performing model **at overall**.

In this assignment, **SARIMAX(2,0,1)(0,1,1,7)** model is the best model both in predicting and forecasting the average daily temperature in Washington D.C. from Sep to Dec 2012. Machine Learning models generally performs well in these tasks.

End of poster