Programming for Business Analytics

Project Report

Introduction: Problem Description

This algorithm aims to forecast Airline booking demand in the Aerospace industry. Airline bookings are very difficult to predict and are influenced by many different factors. Among these factors are past trends, macroeconomic indicators as well as aircraft sales measures. This project is focused on developing an advance booking model which can further be grouped into additive models and multiplicative models. The purpose of this project is to analyze past trends, then use it with various additive and multiplicative forecasting models to develop forecasting methods. In the end, to assess the performance of forecasting models, a calculation of Mean Absolute Scaled Error (MASE) is required as an error measure. The value of MASE indicates that naive method will perform better than forecast values under consideration if it is greater than one.

Input and Output

- Input :
    - Training data
    - Validation data

- Output :
    - MASE
    - Output data frame called "forecasts" which consists of booking date, departure date, historical average (calculated from the training data set), forecast values and error term.

- Intermediate data :
    - trainingdata1: A data frame created from training data frame. Consists following data:
        a. final_bookings(days_prior=0): final bookings when days prior = 0
        b. remaining bookings: final bookings - cumulative bookings
    - historical_remaining_day_avg: Consists of historical average remaining data calculated from training data frame.
    - validationData1: A data frame, created by left joining **validationdata** dataframe and **historical_remaining_day_avg** data frame. Consists following data:
        a. error term
        b. forecast value (additive or multiplicative forecast)
        c. denominator (naive forecast - final demand).

Algorithm

- Initialize training and validation data set
- Calculate days prior for training data and validation data
- Sort values
    - by departure date and days prior for additive and multiplicative days prior forecasting
    - By departure date, days prior and weekday for additive and multiplicative days prior weekly forecasting
- Calculate final bookings when days prior = 0 and create a column for that
- Calculate remaining bookings : remaining bookings = (final bookings - cumulative bookings)
- Create another data frame historical , from training data,
    - Additive Forecast : "historical_remaining_day_avg" contains historical average remaining, which further will be used to calculate forecasting value for validation data
    - Multiplicative Forecast : "historical_remaining_day_RatioAvg" contains historical average ratio
      which further will be used to calculate forecasting value for validation data
- Left join "validation data" and "historical data" on common columns
- Remove all records when days prior = 0
- Calculate forecast values
    - (additive forecast = cumulative bookings + historical average remaining)
    - (multiplicative forecast = cumulative bookings / historical average ratio average)
- Calculate error term: (error term = final demand - additive forecast)
    - For weighted average forecasting method : assign weights to the validation data , based on the error term calculated from the weekly additive forecasting method. Then, multiply weights to the error term
    - Sum of all the weights = 1
    - Weekly additive forecasting method is chosen because its MASE is lowest among all the forecasting methods
- Sum_errorTerm = summation of all the error terms for the forecast
- Calculate error term for naive model: (error term = final demand - naive forecast)
- Denominator = summation of all the error terms for the naive model
- Create output data frame "forecasts" consisting of departure date , booking date and forecast values from the model
- Calculate MASE value

Analyzing the output and efficiency of algorithm

- MASE value for all the forecasting models are :

Forecast Method	MASE
additiveForecast_DaysPrior	1.40817736401
multiplicativeForecast_DaysPrior	0.82984905647
additiveForecast_daysPrior_Weekly	0.786117769141
multiplicativeForecast_daysPrior_Weekly	0.970549255831
Weighted_average	0.0608401089147

Conclusions

- MASE>1 implies that the actual forecast does *worse* out of sample than a naive forecast did in sample and vice versa. The MASE value of multiplicativeForecast_DaysPrior, additiveForecast_daysPrior_Weekly, multiplicativeForecast_daysPrior_Weekly and weighted_average is less than one. Hence, all the four methods performed better than naive forecast.
- Since weighted_average forecasting method has the lowest MASE value, it has performed best among all the implemented forecasting methods.