STAT 52900 Project Report

Promotion Predictor: An Advanced Bayesian Approach

By: Mahesh Moodukonaje

INTRODUCTION

In the dynamic realm of marketing, orchestrating promotions for optimal return on investment demands finesse. Deciding the best day for promotional rollouts requires a profound grasp of customer buying patterns. While the conventional frequentist method relies on historical sales data to pinpoint the day with the highest average daily sales, it falls short of offering insights into the probability of increased sales post-promotion. Such probabilistic insights are invaluable for stakeholders seeking to take calculated risks. This project delves into a Bayesian decision framework, leveraging data from the superstore, to guide decision-makers in selecting the most advantageous day to launch high promotions for the "Liquor, Wine and Beer " product category. Furthermore, it furnishes them with the probability of achieving above-average sales given promotions were implemented on a specific day.

Beyond the determination of the optimal launch day, our exploration extends into a Bayesian time series forecasting framework. This sophisticated approach accommodates the incorporation of the causal effect of promotions, enabling us to predict sales for the upcoming weeks. This not only empowers stakeholders to fine-tune their promotional strategies but also facilitates a more nuanced estimation of the potential return on investment.

STATEMENT OF BAYESIAN APPROACH

The Bayesian framework proves highly valuable in hypothesis testing and decision-making. In this project, we leverage its capabilities to assist stakeholders in comprehending the likelihood of a promotional campaign's success on a specific day[3]. Success is defined as follows:

$$Outcome = \begin{cases} Success, weekday\ sales\ with\ high\ promotional\ activity > Average\ weekday\ Sales \\ Failure, weekday\ sales\ with\ high\ promotional\ activity \leq Average\ weekday\ Sales \end{cases}$$

Based on the outcome, we can choose the Beta distribution to meet our objective

Beta Model:

The Beta distribution[1] is a statistical model frequently employed in Bayesian inference to represent the uncertainty surrounding the probability of success in a binary outcome, such as the success or failure of an event. It is a continuous probability distribution defined on the interval [0, 1], making it particularly suitable for modelling probabilities. The distribution is characterized by two shape parameters, typically denoted as α and β, which govern the shape and spread of the distribution. We chose the Beta distribution for the following 2 reasons:

I. Since the outcome of the event is binary in nature, modelling a Beta-Binomial model is very useful to get the posterior probabilities
II. The model can be used to update the beliefs in the future when new data comes in

Beta-Binomial Conjugate Property:

The Beta-Binomial conjugate property[2] is a key concept in Bayesian statistics, providing a convenient relationship between the Beta distribution and the Binomial distribution. When modeling binary outcomes like success or failure, the Beta distribution serves as the prior for the probability of success, and the Binomial distribution represents the likelihood of observing successes in a fixed number of trials. The posterior distribution, updated after data observation, remains a Beta distribution.

$$\text{Prior: } p(y) \sim Beta(\alpha, \beta)$$

$$\text{Likelihood: } p(x|y) \sim Binomial(n, x)$$

$$\text{Posterior: } p(y|x) \sim Beta(\alpha + x, \beta + n - x)$$

The prior and posterior mean can be written by:

$$Prior\ Mean\ \mu_{prior} = \frac{\alpha}{\alpha + \beta}$$

$$Posterior\ Mean\ \mu_{posterior} = \frac{\alpha + k}{\alpha + \beta + n}$$

Bayesian Time Series Forecasting:

Once the we choose the best day to roll out promotions, we use bayesian time series forecasting to allow the causal impact of promotions to forecast sales for the next 2 weeks. In order to achieve this objective, we use the PyBATS package in python to forecast. The PyBATS[4] package uses Dynamic Generalised Linear Models, State space vector and Markov Chain Monte Carlo Simulations to forecast future values.

Dynamic Generalized Linear Models: Dynamic Generalized Linear Model (DGLM) is a statistical framework used for time series analysis and forecasting. It is an extension of the Generalized Linear Model (GLM) that incorporates dynamic components to model time-dependent patterns and changes in the data over time.

The GLM is a generalization of linear regression that allows for the modeling of relationships between variables when the distribution of the response variable is not necessarily normal. The following equation can be used to to represent the dynamic linear model:

$$Y_t = \beta_t + \Phi_t + Y_{past} + \varepsilon_t$$

- β_t: Time-varying intercept, representing the base level of the series at time t.
- Φ_t: Time-varying autoregressive coefficient, capturing the influence of past observations on the current value.
- ε_t: Independent error term at time t, representing random noise

Bayesian Inference: PyBATS utilizes Bayesian statistics, enabling it to estimate model parameters and uncertainty simultaneously. Parameters β_t and Φ_t are posterior probabilities estimated using the data, selected prior and Markov Chain Monte Carlo Simulations. This provides probabilistic forecasts rather than just point estimates, offering a more comprehensive picture of future outcomes. It supports various distributions for observations, including Poisson, Normal, binomial, and negative binomial

DATASET

This dataset features time-series sales data from a superstore in Ecuador, sourced from Kaggle[5], covering the years 2014 to 2017. It includes essential details like transaction dates, store numbers, product categories, sales figures, and the quantity of promoted items. The dataset records daily sales for each store throughout the three-year span. Importantly, promotional information is absent before 2016, leading to the exclusion of pre-2016 data from the analysis.

	id	date	store_nbr	family	sales	onpromotion	Classified_promo_3	d
2966524	2966524	2017-07-27	44	LIQUOR,WINE,BEER	341.0	4	medium	
2968306	2968306	2017-07-28	44	LIQUOR,WINE,BEER	439.0	5	medium	
2970088	2970088	2017-07-29	44	LIQUOR,WINE,BEER	665.0	5	medium	
2971870	2971870	2017-07-30	44	LIQUOR,WINE,BEER	378.0	6	medium	
2973652	2973652	2017-07-31	44	LIQUOR,WINE,BEER	194.0	4	medium	
2975434	2975434	2017-08-01	44	LIQUOR,WINE,BEER	266.0	5	medium	
2977216	2977216	2017-08-02	44	LIQUOR,WINE,BEER	307.0	4	medium	
2978998	2978998	2017-08-03	44	LIQUOR,WINE,BEER	352.0	4	medium	
2980780	2980780	2017-08-04	44	LIQUOR,WINE,BEER	401.0	4	medium	
2982562	2982562	2017-08-05	44	LIQUOR,WINE,BEER	521.0	5	medium	
2984344	2984344	2017-08-06	44	LIQUOR,WINE,BEER	307.0	3	low	
2986126	2986126	2017-08-07	44	LIQUOR,WINE,BEER	175.0	3	low	
2987908	2987908	2017-08-08	44	LIQUOR,WINE,BEER	225.0	7	medium	
2989690	2989690	2017-08-09	44	LIQUOR,WINE,BEER	260.0	4	medium	
2991472	2991472	2017-08-10	44	LIQUOR,WINE,BEER	387.0	6	medium	
2993254	2993254	2017-08-11	44	LIQUOR,WINE,BEER	560.0	10	large	
2995036	2995036	2017-08-12	44	LIQUOR,WINE,BEER	454.0	8	large	
2996818	2996818	2017-08-13	44	LIQUOR,WINE,BEER	252.0	8	large	
2998600	2998600	2017-08-14	44	LIQUOR,WINE,BEER	186.0	6	medium	
3000382	3000382	2017-08-15	44	LIQUOR,WINE,BEER	206.0	7	medium	

Figure 1: Snippet of the Dataset

In our analysis, we narrow our focus exclusively to the 'LIQUOR, WINE AND BEER' sales category for store 44. This subset of the dataset allows us to effectively illustrate the proof of concept for the Bayesian solution framework. By concentrating solely on this specific category and store, we aim to streamline and showcase the application of our Bayesian approach in a targeted manner.

METHODOLOGY

To design the solution framework, we use the following approach:

- Data preprocessing
- Exploratory data analysis
- Population selection
- Frequentist method to choose the best weekday
- Bayesian method to choose the best weekday
- Forecast sales on the best weekday

Let's have a deep dive into each of the above steps:

1. **Data Preprocessing:**

   On wrangling the data, we found the following:

   - We found 1 null value which we removed
   - Data prior to 2016 had no promotional activity, therefore they were excluded from the analysis

2. **Exploratory data analysis:**

   Exploratory Data Analysis (EDA) is a crucial phase in this process, serving as the preliminary examination of a dataset to unearth patterns, trends, and insights. Through visual and statistical techniques, EDA aims to unravel the underlying structure of the data, enabling analysts to form hypotheses, identify outliers, and assess the distribution of variables.

In the course of this analysis, our aim is to comprehend the data by employing Exploratory Data Analysis (EDA), with the specific goal of gaining insights into customer purchasing behavior. We perform the following analysis to gain a better understanding of the data.

Data Analysis on Promotional Activity:

We observed that the minimum number of products on promotions was 0 and maximum number of products on promotions was 14. We take the frequency plot of the products on promotions to understand how promotions were rolled out in the past year:

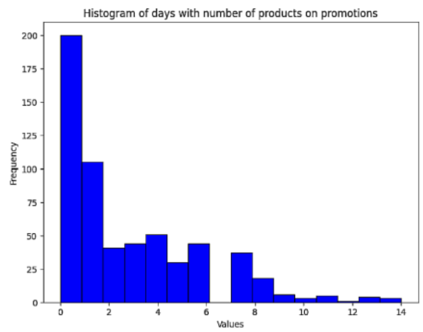


Figure 2: Snippet of frequency distribution of days with respective products on promotions

We observed close to 65% of the days, promotions were rolled out on 0-3 products.

Therefore, based on the spread of number of products on promotions,

$$Promotional\ Flag = \begin{cases} High\ Promotional\ Activity, Number\ of\ Products\ on\ Promotion > 3 \\ Low\ Promotional\ Activity, Number\ of\ Products\ on\ Promotion \leq 3 \end{cases}$$

Figure 3: A bar graph represting difference in average sales based on promotional activity

On analyzing sales with and without high promotional activities, a noteworthy trend emerges: days featuring high promotions exhibit a significant uptick of 38% in average daily sales compared to periods without such promotions. Recognizing the considerable impact of promotional activities, we've strategically chosen to hone our focus on days marked by "High Promotional Activity" to increase the return on investment.

Data Analysis on the Day of the week:

To understand the customer purchasing pattern, we need to understand when and how much do they purchase. We calculate the average daily sales across different days of the week. We also calculate the daily average sales across different days of the week with and without “High Promotional Activity”.

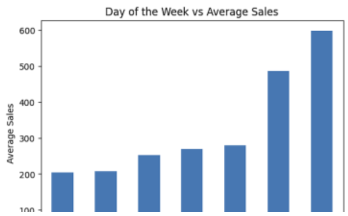


Figure 4a Bar graph representing average spend across different days of the week

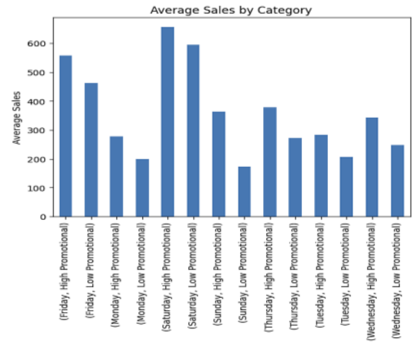


Figure 4a and 4b Are Bar graphs representing average spend across different days of the week

```
day_of_the_week
Monday        20
Wednesday     22
Thursday      23
Tuesday       25
Sunday        27
Friday        32
Saturday      36
```

Figure 5c Represents the number of promotions across days of the week

Our analysis reveals distinct patterns in sales throughout the week. Specifically, Monday through Thursday consistently exhibit lower sales figures in contrast to the notably higher sales observed on Friday, Saturday, and Sunday. In Figure 2, we can observe the similar number of high promotional activities across Monday through Thursday and an increased amount of higher promotional event across Friday through Sunday. Furthermore, in figure 3, it is noteworthy that increased promotional activities correlate with higher average sales, even during weekdays.

Upon closer examination, Wednesday and Thursday emerge as weekdays with notably higher average sales, suggesting that these days hold particular significance in terms of revenue generation.

In light of these findings, we can confidently conclude that weekdays consistently yield significantly lower average sales when compared to weekends. This insight positions weekdays as opportune periods for targeted efforts and strategies aimed at boosting sales performance.

Holiday Analysis:

In our exploration of average daily sales, we specifically examined the influence of holidays on revenue. Our findings indicate that holidays consistently yield a substantial impact, boasting a median sales figure that is 10% higher than that of non-holidays.

Recognizing that holidays often coincide with increased promotional activities and generally exhibit higher-than-normal sales, we made a deliberate decision to exclude them from our analysis. This strategic choice ensures that our insights are focused on the broader trends, independent of the potentially skewed effects associated with promotional peaks on holidays.

By excluding holidays from our analysis, we aim to provide a more accurate representation of the underlying patterns and trends in daily sales, allowing for a clearer understanding and actionable insights for strategic decision-making.

3. Selection of Population:

From Exploratory Data Analysis, we can select the target population with the following characteristics:

- We consider weekdays Monday, Tuesday, Wednesday, and Thursday. We exclude Friday, Saturday, and Sunday for the analysis
- Select days with “High promotional Activity” and exclude days with “Low promotional Activity”

- Exclude holidays from the population to ensure the holidays don't skew the analysis

4. **Frequentist method to choose the best weekday:**
Now that we have narrowed down the population to our interest, we test the hypothesis:

H_0: *"There is no difference in the average sales among Monday, Tuesday, Wednesday, and Thursday"*

$$H_0{:}\mu_M{=}\mu_T{=}\mu_W{=}\ \mu_{Th}$$

H_1: *"There exists at least one such pair of days whose average sales is significantly different from one another"*

$$H_1{:}\ (\mu_M \neq\ \mu_T\ )\ or\ (\mu_M \neq\ \mu_W)\ or\ \ (\mu_M \neq\ \mu_{Th})\ or\ (\mu_T \neq \mu_W)\ or\ (\mu_T \neq\ \mu_{Th})\ or\ \ (\mu_w \neq\ \mu_{Th})$$

To test this hypothesis, we use the two tailed student's T-test to compare the sales incurred on Monday, Tuesday, Wednesday and Thursday among each other. Below is the P values when they are compared between each other:

Week of the day	Monday	Tuesday	Wednesday	Thursday
Monday	-	0.907746	0.27896185	0.0819091
Tuesday	0.907746	-	0.27045274	0.0674276
Wednesday	0.278962	0.270453	-	0.5574749
Thursday	0.081909	0.067428	0.55747494	-

Figure 6 is a matrix with P-Values of T-test among different days of the week

We cannot reject the null hypothesis because there is no significant difference in mean among any day.

We can see Thursday and Wednesday have higher mean, but they are not significantly differently from any other day.

5. **Bayesian method to choose the best weekday:**
To choose the best weekday to roll out high promotions, we use choose a Beta-Binomial conjugate models. We choose the Beta model as a prior, the Binomial model as the likelihood function, and the posterior function would be a Beta model.

$$\text{Prior: } p(y) \sim Beta(\alpha, \beta)$$

$$\text{Likelihood: } p(x|y) \sim Binomial(n, x)$$

$$\text{Posterior: } p(y|x) \sim Beta(\alpha + x, \beta + n\ -\ x)$$

Choice of prior:

To choose the prior for the Beta function, we choose the Beta(5,5) distribution for the following reasons:

- It is weakly informative in nature and it doesn't strongly favor any particular value between 0 and 1. As we do not have a bias or knowledge on how successful the promotional campaign is going to perform

- The choice of prior $\alpha = \beta$, results in a symmetric distribution centered around 0.5. This reflects a situation where you believe the true probability is equally likely to be on either side of 0.5
- In future, if this framework is implemented, we can update the prior as more data is available and the shift in probability can be observed

Likelihood function:

In order to model the data as per the Beta-Binomial distribution, the likelihood Binomial distribution requires the data to be in the form of success (1's) and failure (0's). Therefore, we use the definition stated in the above section:

$$x = \begin{cases} 1, weekday\ sales\ > Average\ weekday\ Sales\ (Lift\ in\ sales) \\ 0, weekday\ sales\ \leq Average\ weekday\ Sales\ (No\ lift\ in\ sales) \end{cases}$$

Therefore, in simple terms, when we observe a lift in sales over the daily average, we consider the impact of the high promotional activity as a success.

Posterior function:

We use the prior and the likelihood function to estimate the posterior probability. There are 2 ways to obtain the posterior probability:

I. Using the Beta-Binomial conjugacy property
II. Using the Markov Chain Monte Carlo Simulation

Each of the above methods will be discussed in the result section

6. **Forecast sales for the next 2 weeks:**

To forecast the sales for the next two weeks, we use the PyBATS model. The PyBATS model uses Dynamic Generalised Linear Models coupled with Bayesian inference to incorporate causal impact of holidays and promotions. This model is known to be easy to use and infer results from.

In this analysis, we consider the time series data after 01-01-2016. We use the Poisson regression with the DGLM model in the base. We then convert the data to integer such that it is usable by the Poisson Regression model.

The PyBATS model needs input of the seasonality aspect of the data and the number of days to be considered to automatically assign a prior value to the prior function. To understand the seasonality aspect, we use ACF plot:

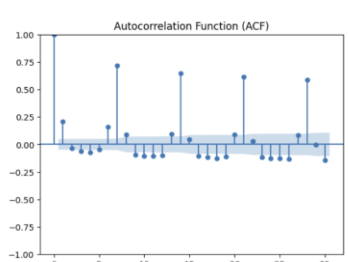


Figure 7ACF plot with 30 lags

From the ACF plot, we can observe seasonality after 7 lags. We use this information in our PyBATS model by initializing seasPeriod=[7].

We ask the model to select the prior considering 7 days of data in the past, by initializing prior_length=7. We also ensure the model considers the first harmonic component of the seasonality by initializing seasHarmonisingComponent=[[1]]. We add random effect “Rho” of 0.055. The below are the model parameters:

```
In [176]:
prior_length = 7 # Number of days of data used to set prio
k = 7              # Forecast horizon
rho = 0.055          # Random effect discount factor to increase variance of forecast distribution
forecast_samps = 100000  # Number of forecast samples to draw
# forecast_start = pd.to_datetime('2016-01-01') # Date to start forecasting
# forecast_end = pd.to_datetime('2017-08-01')   # Date to stop forecasting
forecast_start =300 # Date to start forecasting
forecast_end = 500  # Date to stop forecasting

mod, samples, model_coef = analysis(df_timeseries_new['sales'].values, df_timeseries_new[['promo_flag_new', 'holiday_flag']]
                        k, forecast_start, forecast_end, nsamps=forecast_samps,
                        family='poisson',
                        seasPeriods=[7], seasHarmComponents=[[1]],
                        prior_length=prior_length, dates=df_timeseries_new.index,
                        rho=rho,
                        deltrend=1,                      # Discount factor on the trend component (the intercept)
                        delregn=1,
                        delseas = 1, # Discount factor on the seasonal component
                        delhol = 0, # Discount factor on the holiday component
                        ret = ['model', 'forecast', 'model_coef'])

beginning forecasting
```

Figure 8 Code for PyBATS Model initiation

RESULTS

To help determine the best possible day to roll out promotions, we use posterior probability by using the Beta(5,5) as prior and the data observed is used for the binomial distribution.

Using Markov Chain Monte Carlo Simulation to get the posterior probability:

We use the PyMC3 package to help us get the posterior probability. The PyMC3 package requires us to pass a prior to which, we pass a Beta(5,5)

For the likelihood function, we specify the binomial distribution and pass the number of success to the “observed” parameter and pass the number of experiments to the “n” parameter. We use 6000 simulations with 1000 as the burn in samples. Below is the attached image of the MCMC simulation using PyMC3 package:

```
[81]:
# Generate some example data
np.random.seed(42)


# Define the PyMC3 model
with pm.Model() as beta_binomial_model:
    # Prior distribution for the success probability (Beta distribution)
    prior_success_prob = pm.Beta('prior_success_prob', alpha=5, beta=5)

    # Likelihood (Binomial distribution)
    likelihood = pm.Binomial('likelihood', n=len(thursday_array), p=prior_success_prob, observed=thursday_count_positive)

# Perform sampling
with beta_binomial_model:
    thursday_trace = pm.sample(5000, tune=1000, cores=1)  # You may need to adjust 'cores' based on your system

# Plot the posterior distribution of the success probability
pm.plot_posterior(thursday_trace, var_names=['prior_success_prob'])
plt.show()

C:\Users\madha\.conda\envs\bayesian_2\lib\site-packages\deprecat\classic.py:215: FutureWarning: In v4.0, pm.sample will return an `arvi
  return wrapped_(*args_, **kwargs_)
Auto-assigning NUTS sampler...
Initializing NUTS using jitter+adapt_diag...
Sequential sampling (2 chains in 1 job)
NUTS: [prior_success_prob]
100.00% [6000/6000 00:19<00:00 Sampling chain 0, 0 divergences]
C:\Users\madha\.conda\envs\bayesian_2\lib\site-packages\theano\scalar\basic.py:2851: RuntimeWarning: divide by zero encountered in true
  return np.float32(1.0) / x
C:\Users\madha\.conda\envs\bayesian_2\lib\site-packages\numpy\core\fromnumeric.py:86: RuntimeWarning: invalid value encountered in redu
  return ufunc.reduce(obj, axis, dtype, out, **passkwargs)
C:\Users\madha\.conda\envs\bayesian_2\lib\site-packages\theano\tensor\elemwise.py:826: RuntimeWarning: invalid value encountered in imp
  variables = ufunc(*ufunc_args, **ufunc_kwargs)
C:\Users\madha\.conda\envs\bayesian_2\lib\site-packages\theano\scalar\basic.py:2893: RuntimeWarning: divide by zero encountered in log
  return np.log(x)
100.00% [6000/6000 00:22<00:00 Sampling chain 1, 0 divergences]
Sampling 2 chains for 1_000 tune and 5_000 draw iterations (2_000 + 10_000 draws total) took 42 seconds.
Got error No model on context stack. trying to find log_likelihood in translation.
C:\Users\madha\.conda\envs\bayesian_2\lib\site-packages\arviz\data\io_pymc3_3x.py:98: FutureWarning: Using `from_pymc3` without the mod
  warnings.warn(
```

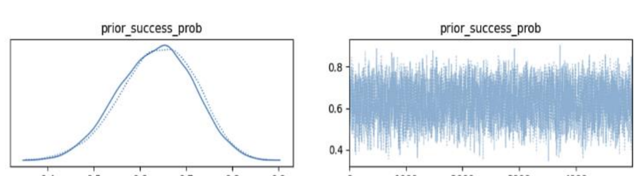


Figure 9 Code and Trace Plot to simulate the posterior distribution for the Beta distribution

We use the MCMC simulation to estimate the posterior probability for all the four days and the results are as follows:

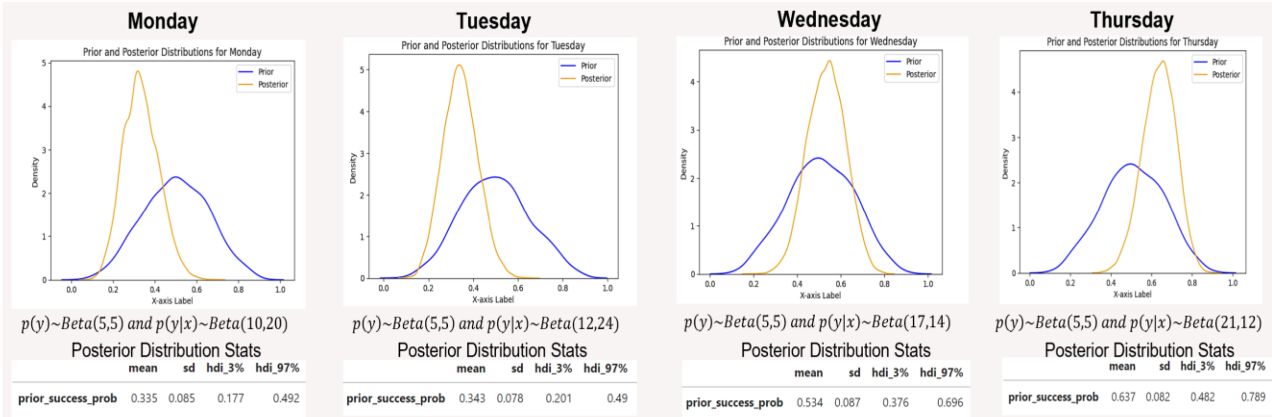


Figure 10 Comparing the prior Beta(5,5) with the posterior distributions for the weekdays

We can observe from the above results that the posterior mean for Monday and Tuesday have a negative shift from the prior mean. This suggests, reduced probability of success if promotions were rolled out on these days. It can also be considered that the probability of success for these days in around 35%.

We observed a positive shift in the mean for Wednesday and Thursday. Wednesday has close to 54% chance of success while on Thursday we observe 63% chance of success. From the results, Thursday seems to be the most favorable day to roll out promotions, with Wednesday being the second most favorable. We can also estimate the posterior probability by counting the number of successes for each day update the probability accordingly.

Forecast the sales for the next 2 weeks using PyBATS:

To forecast the sales for the next 2 weeks, we fit the PyBATS model over 6 months to train the the model:

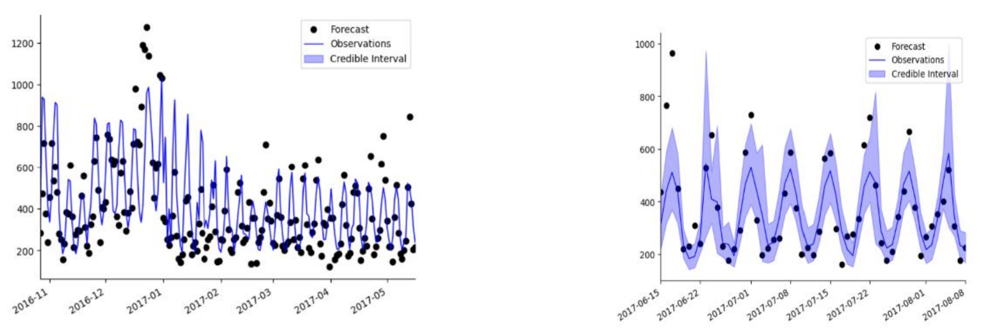
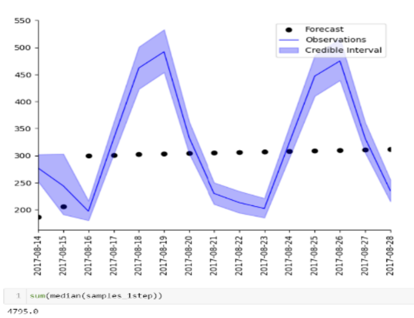


Figure 11 Is the train and the test fir using the PyBATS model

We got a train MAPE of 34% and later we tested it out on the last 1 month of data, which yielded us with a test MAPE of 21%. This is not a great fit but is acceptable as there is high fluctuations in a daily sales data.

We then forecast the result with promotions added on the next two Wednesdays and Thursdays separately and forecast the result:



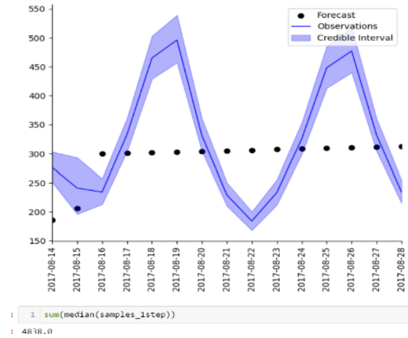


Figure 12 is the plot for forecast for the next 2 weeks for Wednesday and Thursday

We observe that the total revenue generated if the promotion was rolled out on Wednesday would be $4795 and would be $4838 if it were rolled out on a Thursday.

CONCLUSION

Utilizing the Bayesian framework outlined above, we have successfully identified optimal days for launching promotions within the "Liquor, Wine, and Beer" category. Thursday emerges as the most favorable day, presenting a substantial 63% likelihood of a sales boost. Despite this, the sales forecast does not exhibit a notable variance.

To refine our strategy and gain deeper insights, we propose employing an AB testing approach. This involves implementing promotions on both Wednesday and Thursday over two-month intervals, with continuous updates to posterior probabilities. By adopting this methodology, we can effectively measure and adjust our strategies based on real-time results.

This method has enabled us to quantify the probability of success for different days, empowering stakeholders with robust evidence to make informed decisions. Notably, our Bayesian framework stands out in comparison to frequentist methodologies, as it revealed no significant disparities in the mean of the data.

References:

[1] https://en.wikipedia.org/wiki/Beta_distribution

[2] Reading 15a: Conjugate Priors: Beta and Normal (mit.edu)

[3] **https://link.springer.com/chapter/10.1007/978-3-658-32182-6_9**

[4]**https://github.com/lavinei/PyBATS_nbdev/blob/master/examples/Poisson_DGLM_In_Depth_Example.ipynb**

[5] https://www.kaggle.com/competitions/store-sales-time-series-forecasting/data

[6] KAY H. BRODERSEN, FABIAN GALLUSSER, INFERRING CAUSAL IMPACT USING BAYESIAN STRUCTURAL TIME-SERIES MODELS, Google.