BADM PROJECT REPORT

# Reduce the Wait Time For Customers at Checkout

Pankaj Sharma - 61310346

Bhaskar Kandukuri – 61310697

Varun Unnikrishnan – 61310181

Santosh Gowda – 61310163

Anuj Bajpai - 61310663

## 1. Business Objective

In a busy supermarket the number of checkout lanes is constant. Not all customers buy in large volumes. Some buy in small quantity but are forced to wait in long queues at the checkout counter. Our objective is to demarcate a few checkout lanes as "fast checkout lanes" which would exclusively serve these customers (who shop in low quantity), thereby lowering their waiting time. The challenge however is to predict an optimum number of fast checkout lanes, such that on one hand they are able to process these customers fast and on the other they do not remain empty.

The fast checkout lanes, in this case will be altered dynamically everyday based on the predicted demand i.e. the percentage of fast checkout lanes will be directly proportional to the percentage of "small baskets". Hence a model is created which predicts the number of customers with small basket size on a particular day of the week. The model should take into consideration the weekly demand cycle as well as the seasonal variation.

**Benefit** – The benefit of optimizing the checkout lanes is that it improves the service levels and improves customer satisfaction by reducing the time spent waiting. The optimization also seeks to balance the load on fast and regular checkout lanes.

## 2. Data Mining Problem

The data mining objective is to predict the percentage of small baskets on a given day. A small basket is defined as a basket which has a quantity of less than 20 units. The total number of checkout lanes that we are assuming is 50. The reason to make the assumption of 50 checkout lanes were two-fold -

- It is not uncommon for retailers such as Total Mall and SPAR to have as many as 50 checkout lanes in their flagship outlets in big cities in India.
- Since the number of fast checkout lanes is to be proportional to the number of small-sized baskets, limiting the checkout lanes to only 10 would not give us a clear distribution of fast checkout lanes. For example, if 25% and 30% of baskets are "small", then number of fast checkout lanes would be 2.5 (which would be

rounded off to 3) and 3 respectively. Therefore, to get a clearer distribution, we assumed a larger total checkout lanes of 50.

**Predicted variable**

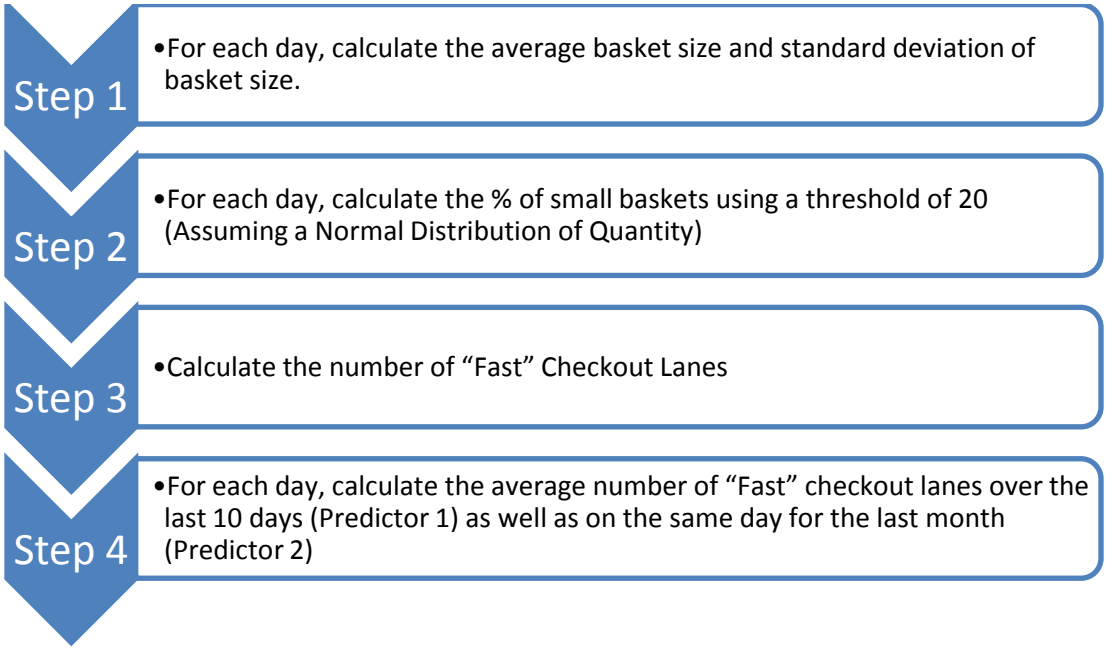
The predicted variable that is the output of the model is the number of fast checkout lanes that is required on a particular day.

**Predictors**

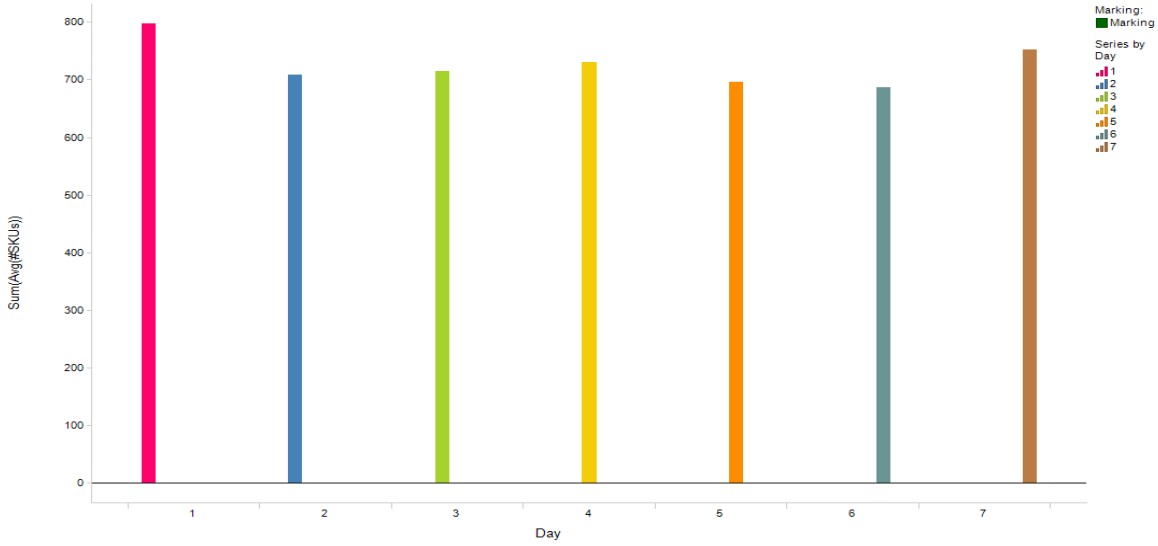
- **Number of "Fast" checkout lanes over the last 10 Days** – This helps to capture the effect of seasonality in the sales.
- **Number of "Fast" checkout lanes over the last 4 weeks on that day** – This helps to capture the weekly trend or the daily trend in sales.

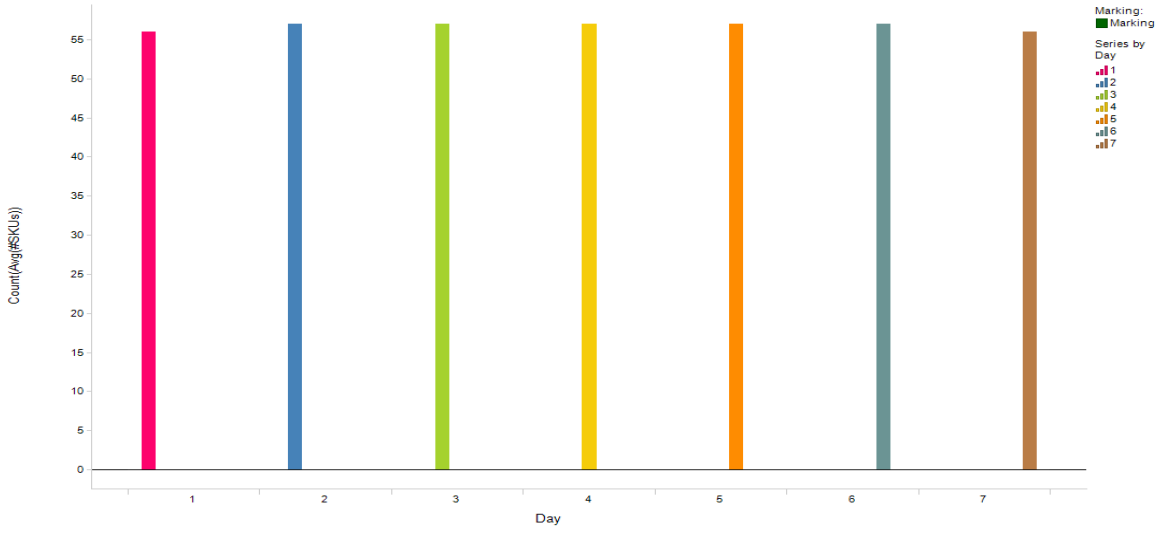
## 3. Data Preparation

To create the model we used 1 Year's worth of daily basket level data. Data (all rows) for a particular day were aggregated to create one row for a day. The following steps were followed to prepare the data



Graphical relations in Data:




The above graph shows higher sales on Saturday and Sunday as compared to other days. Also, second graph shows that even the sale of SKUs on weekend is higher but the number of SKUs sold on weekdays is almost equal that on weekdays. This implies that dynamic checkout lanes can be helpful in reducing the time to serve the customers.

## 4. Benchmark

From the data transformation we created in the previous step, we created a pivot table to find the simple average of the average fast checkout lanes required using the data for 365 days. The findings are as below. This is used as the benchmark for our predictions.

| Day of the week | Average Fast Checkout lanes |
|---|---|
| 1 | 13.9 |
| 2 | 15.4 |
| 3 | 15.3 |
| 4 | 14.9 |
| 5 | 15.5 |
| 6 | 15.5 |
| 7 | 14.7 |

## 5. Methods

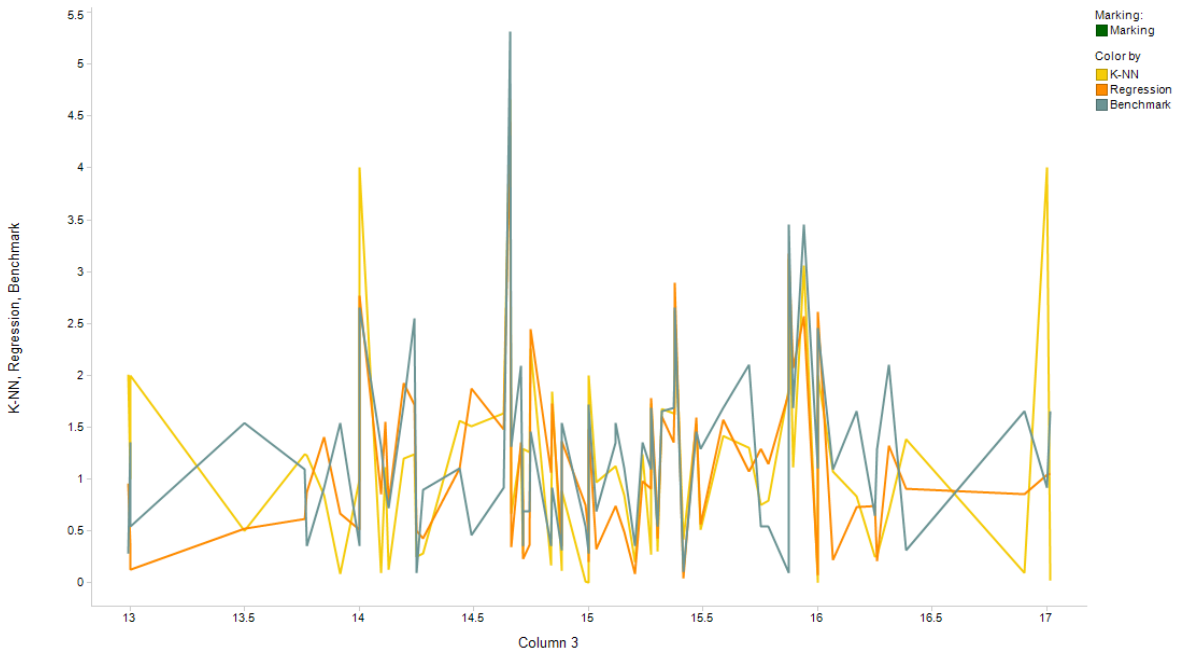
The two methods we felt that will best help predict the number of fast checkout lanes were Multiple Linear Regression and K-Nearest Neighbour. Although we knew were playing with just a year's data that could limit the effectiveness of the KNN algorithm, we proceeded with it to check its prediction accuracy vis-à-vis the Regression model. The data set for both models was divided into Training, Validation and Test in the ratio of 50:30:20.

## 6. Evaluation

To evaluate the prediction accuracy of our models i.e. Multiple Linear Regression and KNN models, we first calculated the average difference between the number of fast checkout lanes as predicted by our benchmark and the actual number of fast checkout lanes as given by the test set. This turned out be 1.24. We then repeated this exercise for number of fast checkout lanes as predicted by both our models. The following table summarizes the three results -

| Method | Average Difference |
|---|---|
| Actual vs. Benchmark | 1.24 |
| Actual vs. Multiple Linear Regression | 1.105 |
| Actual v.s KNN | 1.16 |

As you can see, both our models, Multiple Linear Regression and KNN both have lower average difference between actual and predicted fast checkout lanes than the benchmark comparison. The Regression model with average difference of 1.105 displays a 11% improvement in prediction accuracy over the Benchmark.




The graph depicted that the ability of models to surpass the prediction by the benchmark while predicting values. The K-NN and Regression are able to beat the benchmark in most of the cases.

## 7. Insights

In the step 2 of Data Preparation, we assumed the quantity to be normally distributed. The basket size on any given day in the year ranged from 0 to 960. Hence normal distribution was not a bad approximation. Also as part of our analysis, we created a pivot table to calculate the percentage of small baskets using the threshold as 20 for each day. The average baskets required was little different than the one used using the normal distribution approximation but the prediction accuracy was more or less the same among the KNN method, Multiple linear regression and the benchmark model. The only reason we went with the normal distribution approximation is that the model

is more scalable in terms of identifying the "Fast" checkout lanes required for different thresholds.

Based on the results from our data mining exercise, we found that the Regression model gave us more accurate predictions. However, this was the case because the amount of data available was very less. In case there is more data available, the KNN model would work much better and should be used in lieu of the Regression model.

## 8. Challenges/Problems faced

- We needed daily data for prediction, and as there was only 1 year data we had very less number of records (only 365 rows of data). It is expected that K-NN will be able to provide better results if the rows of data would have been more.
- The data set had limited number of columns thus we have to make certain assumptions such as number of total check-out lanes and number of fast check out lanes. If the data related to this has been provided the predictions would have been better. Similarly, hourly break up of sales data can be used to create a better model where stores can dynamically change the fast check-out lanes each hour based on expected sales.

## 9. Appendix

### Validation error log for different k

| Value of k | Training RMS Error | Validation RMS Error | |
|---|---|---|---|
| 1 | 0.613402499 | 1.740604351 | |
| 2 | 0.613402499 | 1.550421741 | |
| 3 | 0.613402499 | 1.513258118 | |
| 4 | 0.613402499 | 1.499647648 | |
| 5 | 0.613402499 | 1.487517528 | |
| 6 | 0.613402499 | 1.426965503 | |
| 7 | 0.613402499 | 1.43057689 | |
| 8 | 0.613402499 | 1.415198173 | <--- Best k |

### Training Data scoring - Summary Report (for k=8)

| Total sum of squared errors | RMS Error | Average Error |
|---|---|---|

| | | |
|---|---|---|
| 74.5 | 0.613402499 | -1.0101E-08 |

## Validation Data scoring - Summary Report (for k=8)

| Total sum of squared errors | RMS Error | Average Error |
|---|---|---|
| 238.3315185 | 1.415198173 | -0.17563305 |

## Test Data scoring - Summary Report (for k=8)

| Total sum of squared errors | RMS Error | Average Error |
|---|---|---|
| 181.5234865 | 1.506334485 | 0.245993075 |

## The Regression Model

| Input variables | Coefficient | Std. Error | p-value | SS |
|---|---|---|---|---|
| Constant term | 2.41092038 | 1.7997483 | 0.18193854 | 44910.72656 |
| Predictor 1 | 0.55210441 | 0.13465534 | 0.00006053 | 71.3438797 |
| Predictor 2 | 0.28802398 | 0.10329486 | 0.00582214 | 13.0338726 |

## Training Data scoring - Summary Report

| Total sum of squared errors | RMS Error | Average Error |
|---|---|---|
| 326.8949702 | 1.284906494 | -3.5535E-08 |

## Validation Data scoring - Summary Report

| Total sum of squared errors | RMS Error | Average Error |
|---|---|---|
| 198.9360112 | 1.292954441 | -0.14501853 |

## Test Data scoring - Summary Report

| Total sum of squared errors | RMS Error | Average Error |
|---|---|---|
| 147.3655091 | 1.357228375 | 0.176801001 |