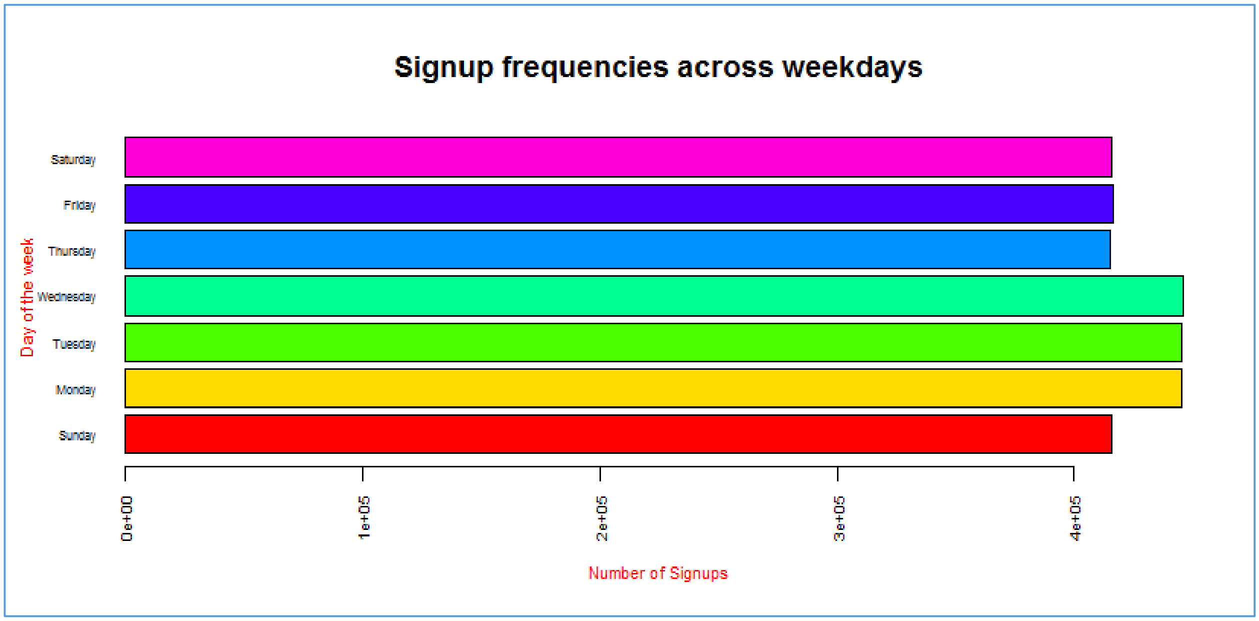
Living Social Data Science Challenge

Pavani Satish Kalepalli

Email: sat37@outlook.com Phone: 716-289-6839

Problem 1: Data Analysis

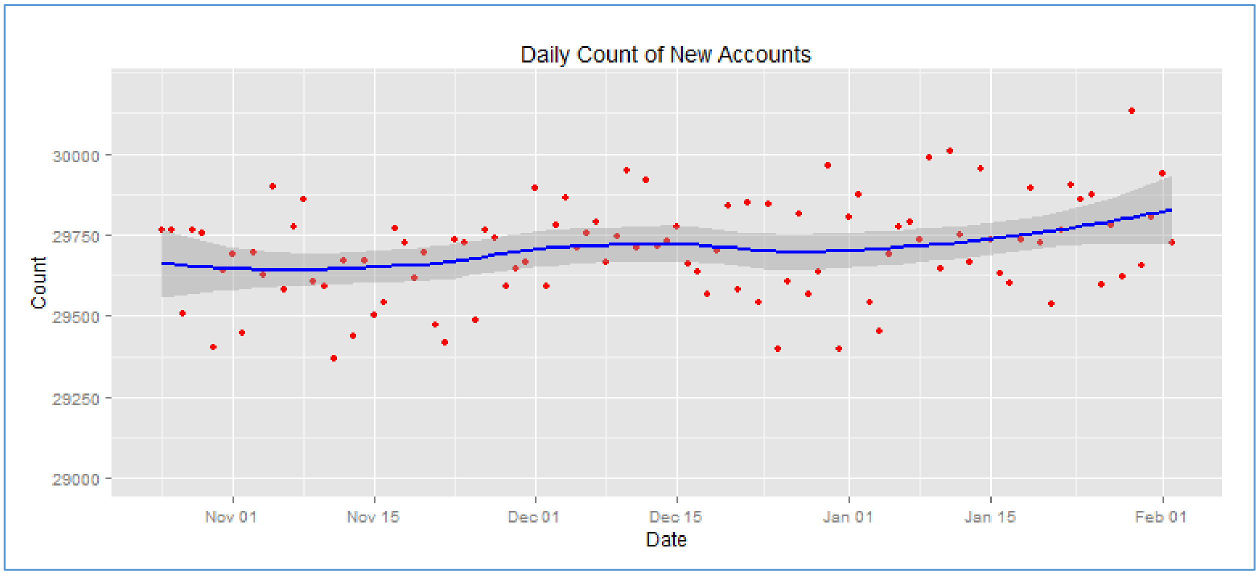
- The given data set was analyzed using R-Studio and the observations and results of various experiments were obtained.
- Refer to the R scripts for the implementation details.
- The metrics, plots and inferences are detailed in the following sections:



Most number of accounts were created on the following days:

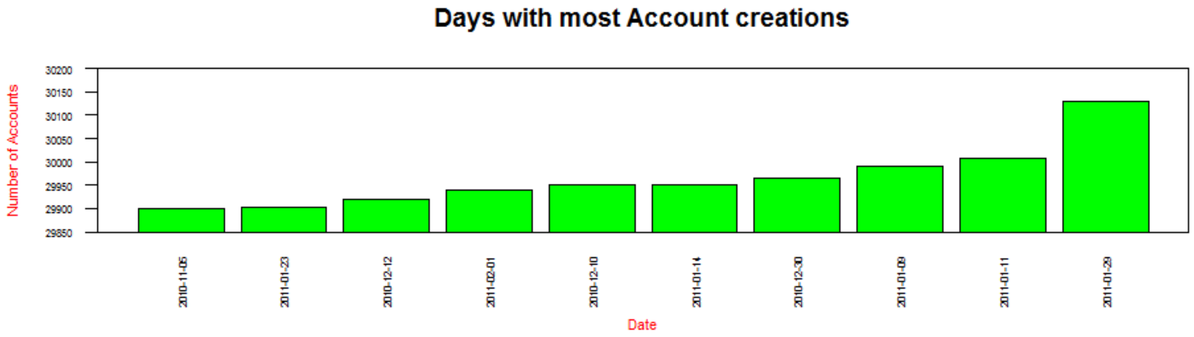
Wednesday – 445760 Tuesday- 445442 Monday- 445452

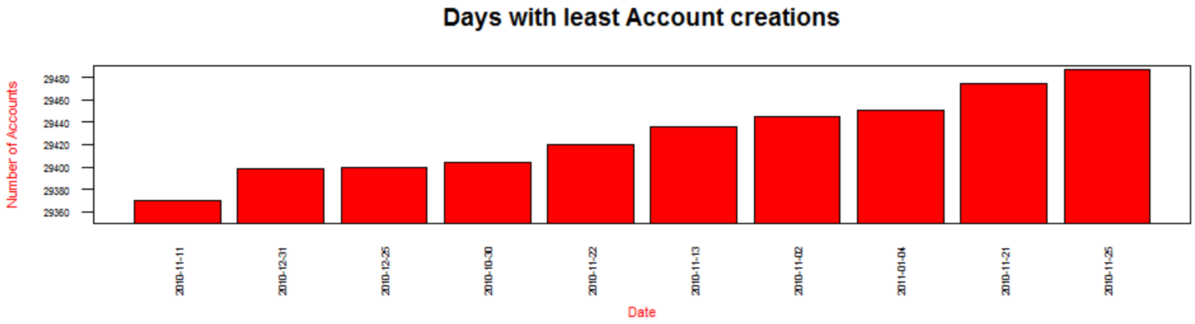
Possible trend and Inference: Customers are planning their weekends in advance!



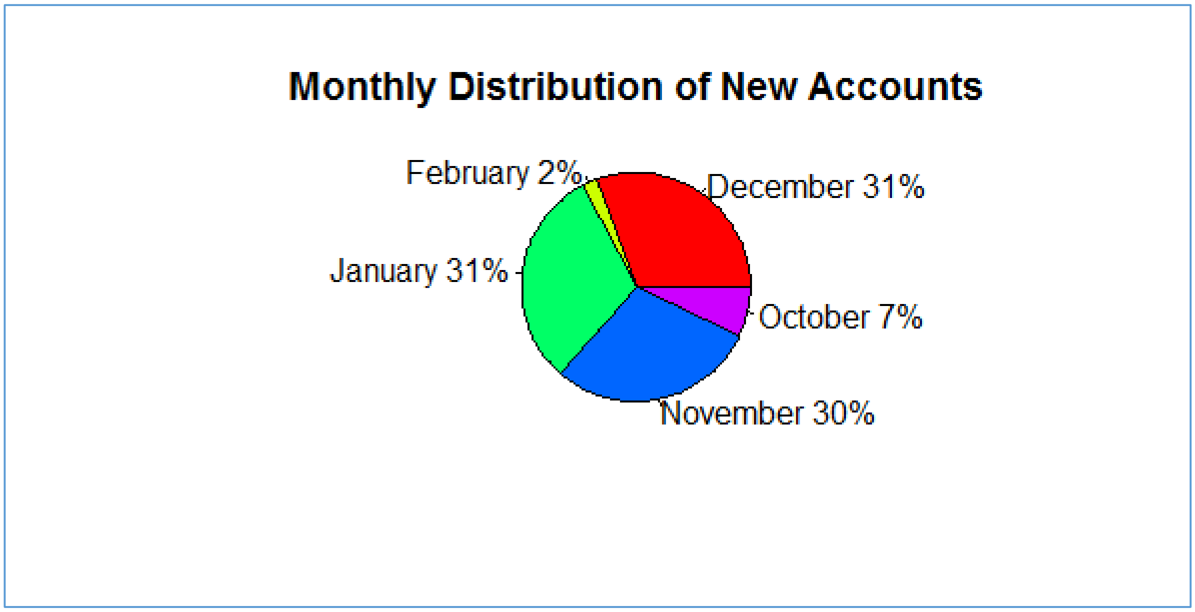
During the given period, the number of new accounts created per day range from 29,370 to 30,131

Possible trend and Inference: There are no spikes or valleys in the above curve, indicating that there has been a slow and gradual increase in the number of customers and there are no special seasonal changes during the given period.

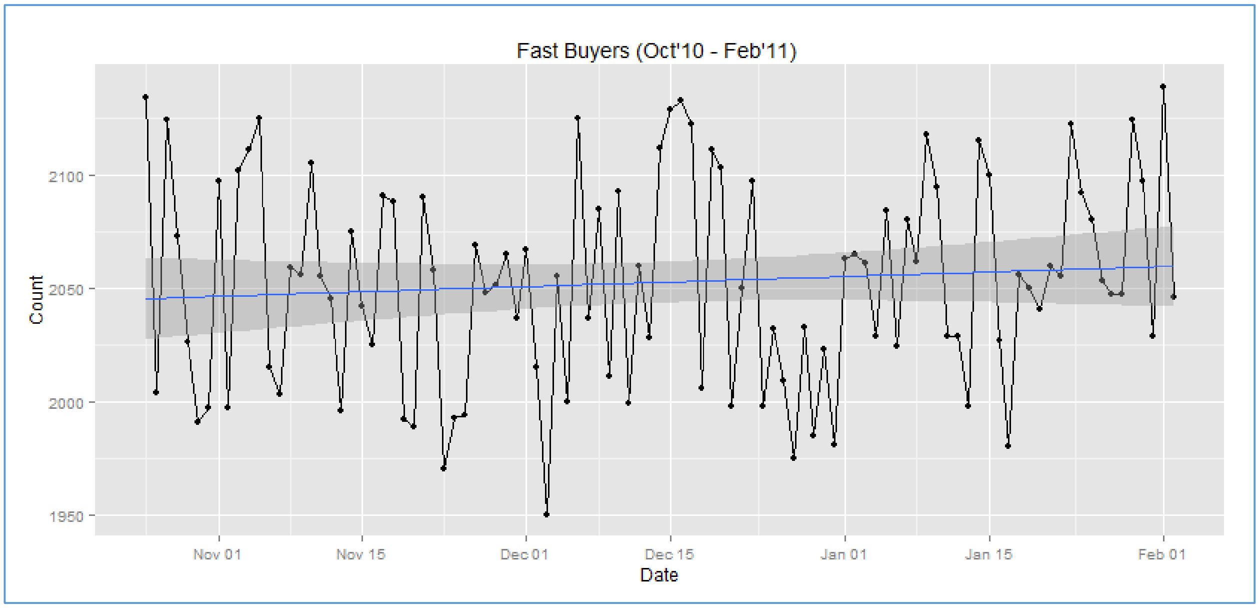




Highest number of customers signed up on 01-29-2011 and the lowest number of customers signed up on 11-11-2010, indicating that these are special days in terms of offerings, discounts or festive occasions.

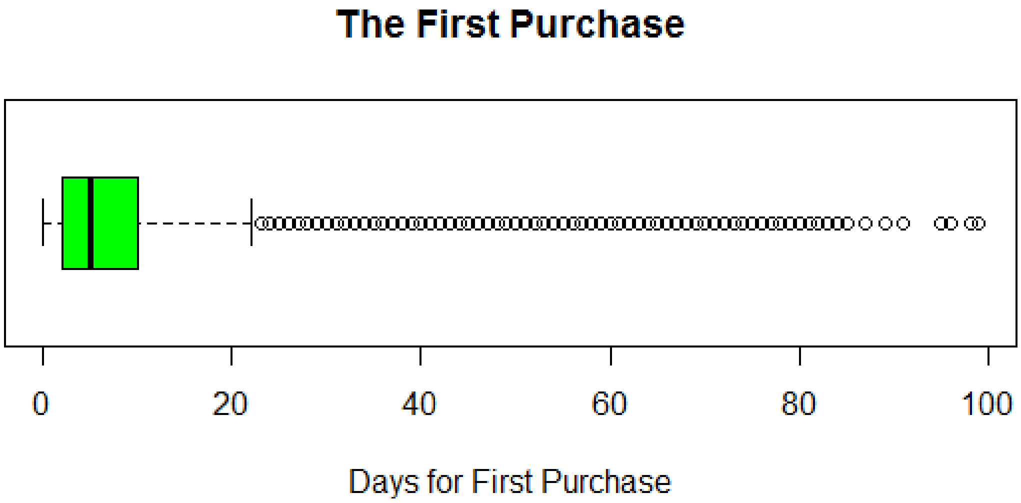


Inference: The number of new customers in all the months are evenly distributed. The percentages in February and October are low only because the data does not include all the dates of those two months. The pie chart corroborates the fact there are no seasonal changes during the given period and the trend of new customers is constant across all seasons.



The above trend depicts the number of users who buy products on the same day of account creation. These are the fasted buyers – The lowest count of fastest buyers is 1950, recorded on 12/03/2010 and the highest count of fastest buyers is on 02/01/2011 which is 2139.

Inference: The number of customers who buy on the same day of account creation is always less than 10% of the number of accounts created per day: This indicates that the trend of impulsive buying is very low during the given time period. This also indicates that there are no attractive offers or discounts during this period that encourage impulsive buying.



Number of Days for making the First Purchase					
Minimum	**1st Quartile**	**Median**	**Mean**	**3rd Quartile**	**Maximum**
0	2	5	6.99	10	99

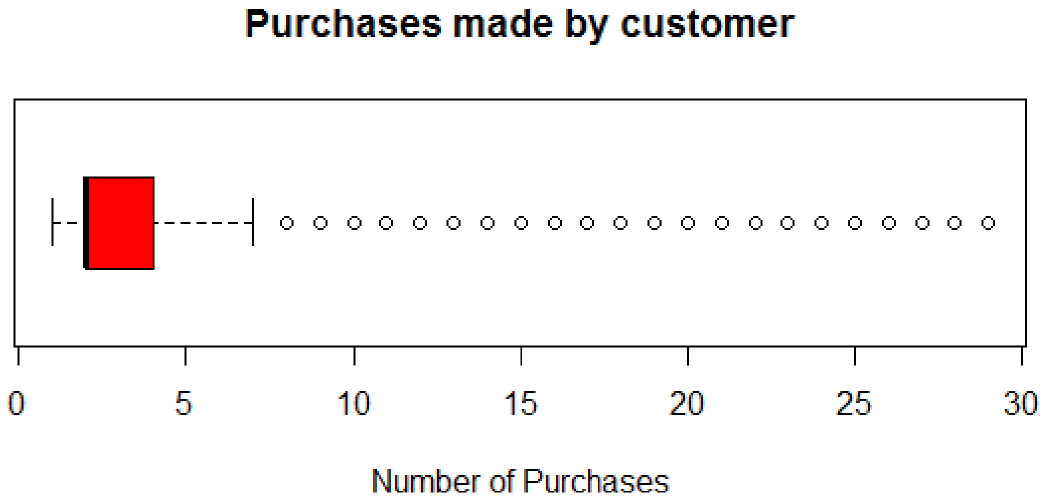
Days for FIRST PURCHASE	CUSTOMER COUNT
0	207311
1	370706
2	322232
3	280065
4	242125
5	210217

Inferences:

- From the boxplot and the distribution tables above, it can be observed that the most number of users took at least one day ONE day to make their first purchase.
- This data also corroborates the previous facts that the rate of impulsive buying is very low
- The boxplot confirms that there are very few users who took more than 10 days to make their first purchase.
- Most number of customers took ONE day to make their first purchase. This can be inferred as either of the following: The customers are very thoughtful in making their selection (or)

the site is not very user-friendly in terms of enabling the customer to make the payment on the same day of signup.

- The general trend indicated by the above data is that the customers create a new account only if they desire to buy goods immediately and do not take a lot of time for making the first purchase after the Sign-up.

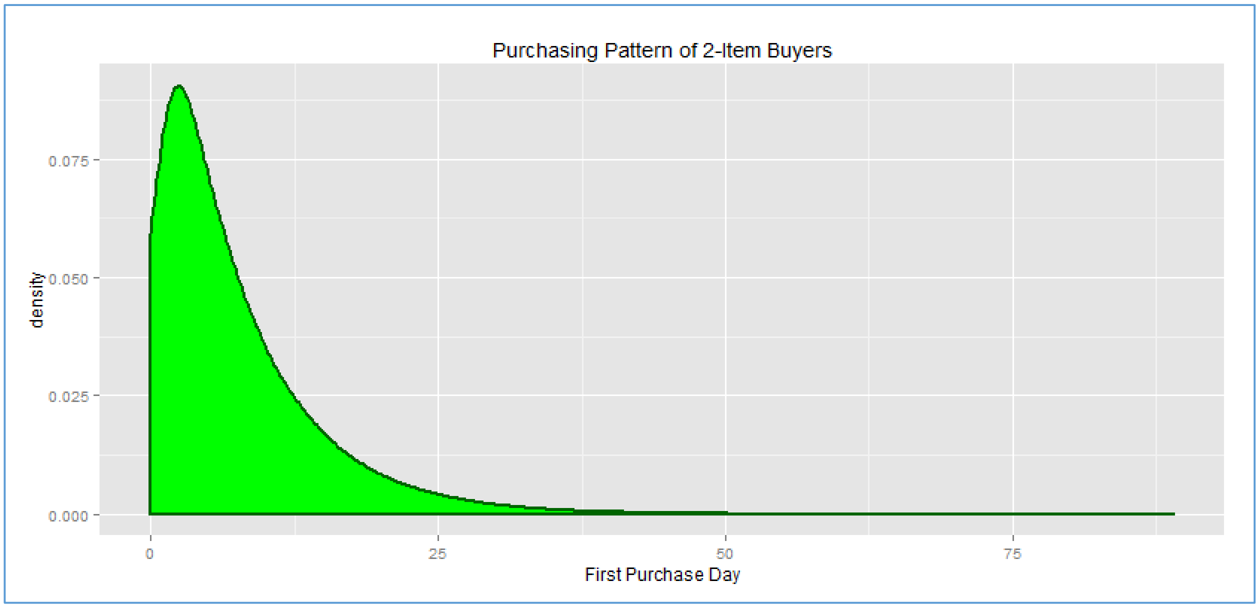


Count of Purchases made by a customer					
Minimum	**1st Quartile**	**Median**	**Mean**	**3rd Quartile**	**Maximum**
1	2	2	2.97	4	29

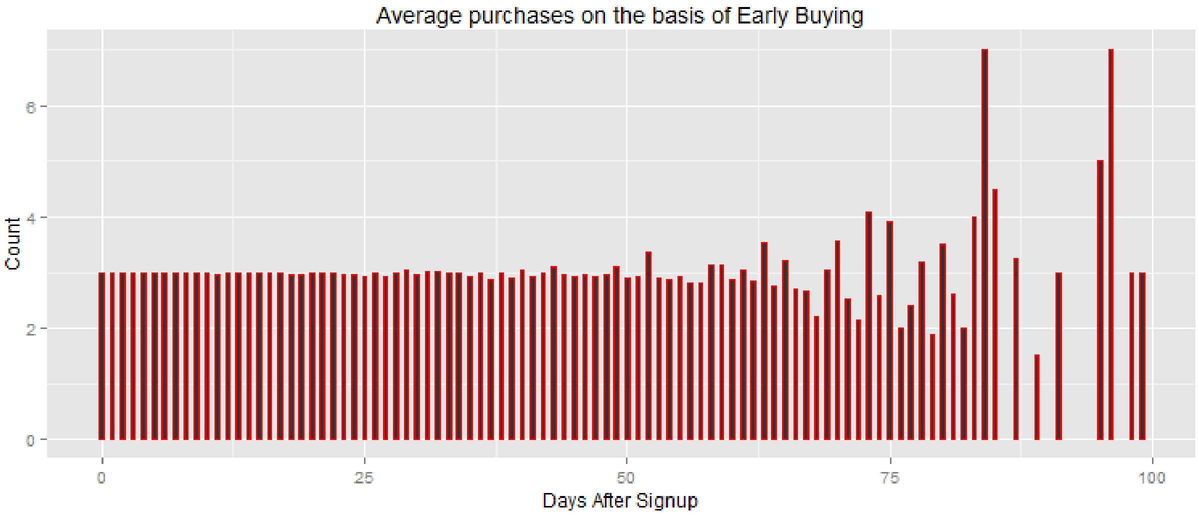
Purchases Made	Customer Count
1	663857
2	918932
3	557538
4	338199
5	205447
27	5
28	2
29	3

Inferences:

- From the boxplot and the distribution tables above, it can be observed that the most number of users purchased TWO items and followed by ONE item.
- Very few users purchases more than FOUR items and only 3 users purchased 29 items which is the highest number of purchases made by any given user
- The count of customers who purchased TWO items is approximately 1.5 times that of the users who bought ONE item indicating that there is a special 'Buy 1-Get 1' or a discount on TWO purchases
- The purchasing pattern of TWO-item buyers can be plotted as below:



- The above plot reaffirms that most users <u>did not</u> buy on the <u>same day</u> of their Sign-up and they made their first purchase <u>within TWO days</u> from the day of sign-up. This plot also corroborates the previous inference that there might have been a special discount for buying the second item after the first purchase.



- The above plot summarizes the purchasing pattern of customers on the basis of speed of their first purchase.
- It can be inferred that the average number of items purchased by a customer is THREE and it <u>remains a constant irrespective</u> of the speed of the first purchase. It indicates that the customers buying immediately after their Signup <u>do not</u> actually buy more items. Similarly, the customers buying after a long time after signup <u>do not</u> buy fewer items than the ones buying early.
- However, the peaks and valleys at the rear end of the plot <u>are only because there are fewer data points in the region</u> 'first purchase after 75 days' and they <u>do not</u> indicate any abnormalities in the trend which is a constant one.

Problem 2: Building a Model

Option 2: Deal Targeting

Objectives and Motivation

- The problem statement states that Takeout and Delivery services are being used by LivingSocial users of various cities and the aim of this experiment is to describe a model that predicts whether a LivingSocial user living in another city will be interested in opting for a similar service or not.
- This a classification problem which aims to classify the LivingSocial users in the new cities into THREE classes namely – Takeout ,Delivery and Dine-in (Takeout and Delivery are treated as separate classes as they are different services in the real world)
- The key to designing this model lies in the rich transactional data obtained from the customers' orders in the cities where the service is currently being offered. This transactional data along with customers' profile information reflect their preferences. Thus, the proposed classifier model learns parameters from these preferences to classify the population into the above mentioned categories.
- The most vital assumption is about the service and customers not being regional. In case, the services and customers are very regional, every city would require a different classifier to distinguish the customers' preferences.

The Dataset

- The data is collected from the user actions and profiles of customers living in cities where the services are being offered.
- Each row in the dataset corresponds to a combination of user preferences and the class label is one of these three classes: Takeout, Delivery and Dine-in.
- The feature vector is a combination of many features(a finite number) , a few of them can be elaborated as below(assuming that the service is related to food):

Nature of Food: Fast Food, Buffet , Desserts, Festive foods etc.
Size of the food: suitable for takeout and delivery or not?
Size of Restaurant: Small, Medium, Large etc.
Ambience: Noisy, Peaceful, Posh, simple etc.
Like/Dislike preferences
Rating given by the customer
Number of Views/ View Details by the user
Comments provided or not?
Age of the customer : Younger generation prefers takeout and delivery while older generation prefers dine in.
Time of placing the order: Morning, Noon, Evening, Night, Late night etc.
Turnaround time at service location
Lack of time to cook: can be obtained from the professional details section of profiles
Number of Kids and Children

The Model

- The proposed model uses Naïve Bayes classification to learn the feature vectors and estimate the parameters for class prediction
- Naïve Bayes classification, as the name implies, is easy to implement, fast to train, and handles both real and discrete data. We can compute all the probabilities with a single scan of the feature vectors and store them in a small table. It is space efficient.
- Moreover, it is not sensitive to irrelevant features. For example, the number of views of an item may not be directly linked to whether a customer places a takeout order or prefers dine in but the inclusion of this feature assumes that we have a good enough estimate of the probabilities. Hence, the more data the better. It can also handle streaming data.
- It is a probabilistic classifier that assumes independence of features and computes the maximum likelihood of each feature for a given class.
- Another advantage of using Naïve Bayes classifier is that each feature can be assumed to have been generated from a different probability distribution like Bernoulli, Gaussian and Binomial depending on the nature of that particular feature(whether continuous or discrete) and the final estimate is a joint probability estimate of all the individual probabilities of features for a given class.
- There are four kinds of probability estimates: using prior, maximum likelihood, posterior or the Bayesian averaging method of estimation.
- If only a prior is used, the model becomes too simple and doesn't make use of the training data which is incorrect. The Maximum likelihood option provides a good estimate but is prone to the Black swan problem if all the training examples are of the same class. The posterior estimate is a revised estimate that makes use of both maximum likelihood and the prior. It is a point estimate while the Bayesian averaging estimate is the best one as it is a weighted estimate of all the probabilities.
- Another reason for choosing the Naïve Bayes over other classifiers is because, for training sets that are small, the Bayes classifier converges quickly where as other classifiers like logistic regression and k-NN are prone to over fitting with the increase in number of features.
- The other models like SVM, logistic regression and large margin classifiers may have higher accuracy but they are not necessary as the given problem is not a risk-oriented cancer prediction problem where accuracy is preferred. Moreover, that kind of accuracy comes with a cost of over fitting and they are not easily adaptable to new training data.
- In a typical Naïve Bayes Model, the first step is to define the probabilistic discriminant functions: E.g. $p(y= \text{Takeout}/x)$, $p(y=\text{Delivery}/x)$, $p(y=\text{Dine in}/x)$
- Then, the class conditional densities of each feature is estimated: $p(x/y=\text{Takeout})$, $p(x/y=\text{Delivery})$,$p(x/y=\text{Dine in})$
- All input attributes are conditionally independent of each other given the class. Therefore, $p(x, y) =p(x/y)\ p(y)$ where $p(x/y)$ is the product of the class conditional densities of individual features.
- After computing the Likelihood and posterior estimates, we choose the class that explains the input data better (MLE) or the class with better posterior probability i.e. $p(y=\text{Takeout}/x^*) > p(y=\text{Delivery}/x^*) > p(y=\text{Dine in}/x^*)$ or vice versa.

Evaluation of Model

- In the simpler method, we split the dataset into training data and test data (using the 70-30) ratio.
- The model learns using the training data and predicts on the test data.
- The test errors give an honest estimation for future cases i.e. the cases belonging to the new cities where LivingSocial is planning to start the services.
- During the evaluation process, we can avoid the train/test bias i.e. we can avoid using one split of train/test as it involves a luck factor
- Instead, we can choose a much better(less-biased) option of using multiple train/test splits and averaging the test errors obtained on these splits.
- The splitting of dataset can be done using two methods: Random subsampling or cross-fold validation.
- In random subsampling, we choose the train and test set ‘n’ times randomly. However, the cross-fold validation is better: It partitions the data into ‘n’ disjoint subsets and training is done on n-1 subsets and testing on the remaining one.
- For the test data set, we can build the precision-recall matrix (or) the confusion matrix. The confusion matrix is generally used with binary targets but can also be extended to our problem of three-class classification.
- In the multiclass classification setting, the notions of precision, recall and F-measure can be applied to each label independently. There are a few ways to combine the results across metrics, defined by the type of averaging method used: micro (or) weighted. "Micro" gives each sample-class pair an equal contribution to the overall metric. Rather than summing the metric per class, this sums the dividends and divisors that make up the per-class metrics to calculate an overall quotient. Micro-averaging method is preferred in multi-class setting where a majority class is to be ignored.
- In the micro-averaging case, the multiclass setting will produce an equal precision, recall and F-score while the “weighted” averaging may produce an F-score that is not in between precision and recall.
- Other metrics that can be used for evaluation of multiclass setting include Sensitivity, Specificity, Positive predictive value and Negative predictive value. Another metric is the ROC which can be extended to a multiclass problem.
- The other metrics are the misclassification error metrics based on zero-one loss functions and general loss function (a more weighted one).

Things to watch out for in Naïve Bayes Classification

- A subtle issue with Naïve-Bayes is that if there are no occurrences of a class label and a certain attribute value together, then the frequency- based probability estimate will be zero. Given the conditional independence assumption, when all the probabilities are multiplied, we get a zero and this affects the posterior probability estimate. The problem occurs when the drawn samples do not represent the entire population. This can be avoided by using Lagrange correction.
- - Number of user examples to be collected: The ideal size of the data to be collected depend on the training and cross-validation error estimates. If the test error is much higher than the training error, and it reduces as we increase the number of examples, it suggests a scope for improvement and the need for more training examples. If the difference between the test error and training error is a constant even after a significant increase of examples, it means

that additional data is not required. If more data is provided at this stage, it only adds redundancy of features. However, Naïve-Bayes uses the redundant and irrelevant features for a better probability estimation.