CS 6375.501 Machine Learning

Project Report

Santander Customer Satisfaction

Submitted By

George Koshy (gxk140830)
Kevin George (kag140330)
Siddharth Jhaveri (spj140030)

1. Introduction:

The dataset chosen for the Machine Learning final project by our team is the Santander Customer Satisfaction dataset which is an ongoing competition in Kaggle.
Kaggle link: https://www.kaggle.com/c/santander-customer-satisfaction/
Project github link: https://github.com/gkoshyk/Kaggle-SantanderCustomerSatisfaction

2. Problem Definition and Algorithms:

2.1 Task Definition:

Customer satisfaction is a key measure of success. Unhappy customers tend to churn. But unhappy customers rarely voice their dissatisfaction before leaving.

The aim is to identify dissatisfied customers early in their relationship. Doing so would allow Santander to take proactive steps to improve a customer's happiness before it's too late.

We would use hundreds of anonymized features to predict if a customer is satisfied or dissatisfied with their banking experience.

2.2 Description of the Dataset:

1. The number of instances in the dataset is 76020.
2. There are a total number of 370 attributes.
3. The attributes have a mix of both binary and continuous values.
4. Unfortunately Santander has not disclosed the meaning and description of each of these attributes, they are anonymized. So it doesn't help us in extracting semantic features.
5. Type of the output variable is Class Value i.e. whether the bank customer would churn or not.

2.3 Algorithm Definition:

We will be using the following Algorithms:

1) Dimensionality Reduction:

   a) Removing features with low variance Variance Threshold (Low Variance Filter):

   Data columns with little changes in the data carry little information. Thus all data columns with variance lower than a given threshold are removed. A word of caution: variance is range dependent; therefore normalization is required before applying this technique.

2) Classification:

   a) Naive Bayes (Primary)

   The Naive Bayesian classifier is based on Bayes' theorem with independence assumptions between predictors. A Naive Bayesian model is easy to build, with no complicated iterative parameter estimation which makes it particularly useful for very large datasets. Despite its simplicity, the Naive Bayesian classifier often does surprisingly well and is widely used because it often outperforms more sophisticated classification methods.

   b) SVM with sigmoid kernel function (Primary)

   A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

   c) Boosting (AdaBoost and Gradient Boost) :

   Boosting is a machine learning ensemble meta-algorithm for primarily reducing bias, and also variance in supervised learning, and a family of machine learning algorithms which convert weak learners to strong ones.

AdaBoost

AdaBoost gives a higher performance when used along with many other algorithms. The output of the other learning algorithms ('weak learners') is combined into a weighted sum that represents the final output of the boosted classifier.

Gradient Boost

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function

d) Other Classifiers

Other classifiers taught in the class like Bagging, Neural Networks, Decision Tree and Perceptron.

3. Experimental Evaluation:

3.1 Methodology:

1) We have used PCA and Variance Threshold for dimensionality reduction. Once the Analysis is done, the number of attributes is reduced from 370 to14.
2) All the classifiers use accuracy and MSE as the evaluation criteria.

3.2 Results:

After applying PCA

Predicted class labels of the order as below

```
array([[ 0.94135716,  0.05864284],
[ 0.94135716,  0.05864284],
[ 0.96494056,  0.03505944],
...,
[ 0.96494056,  0.03505944],
[ 0.96494056,  0.03505944],
[ 0.94135716,  0.05864284]])
```

(Only included few as there are 75k instances)

Results in terms of accuracies after applying various dimensionality reduction techniques

Number of instances = 76020
Number of attributes after dimensionality reduction = 14

Dimensionality Reduction Technique	SVM Accuracy	Naïve Bayes Accuracy	Decision Tree Accuracy	K - nearest Accuracy	Bagging	Neural Network	Ada Boost
PCA	0.934901	0.963036	0.930282	0.962904	0.950802	0.963036	0.962641
SVD	0.963036	0.963036	0.930018	0.962641	0.962641	0.963036	0.962904
Low variance Threshold	0.962094	0.963036	0.930018	0.962641	0.962641	0.963036	0.962904

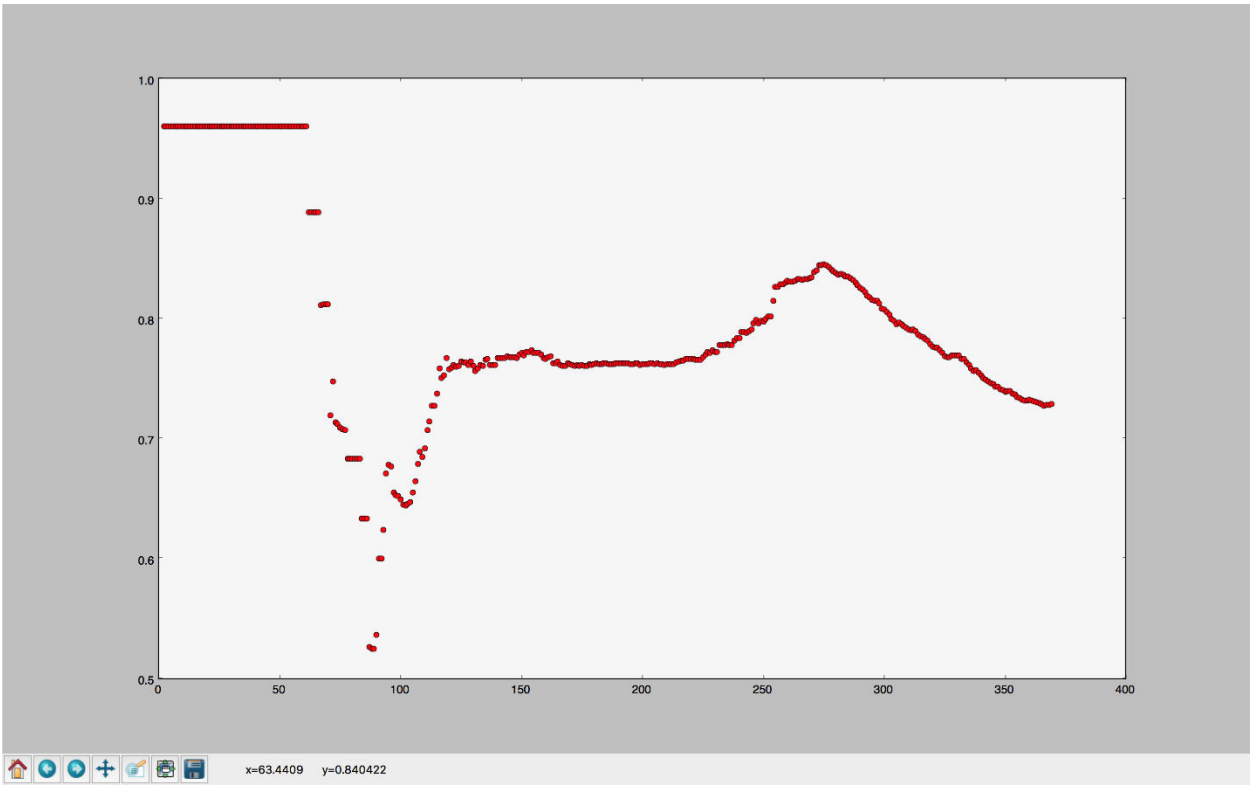
Programmatically finding the optimum number of features to select for better accuracy with Naive Bayes using test-train split

The accuracy of the results varied with selecting the number of attributes. Accuracy was maximum and was 96% when selecting attributes less than 60, while accuracy was lowest while selecting attributes between 88 – 90. (~52%)

Performing above thing with k fold cross validation, instead of train-test split

The accuracy of the results varied with selecting the number of attributes. Accuracy was maximum(~95%), mean squared error was minimum (~0.04) when attributes were less than 60, while accuracy was lowest while selecting attributes between 87 – 90 and MSE was highest (0.46).

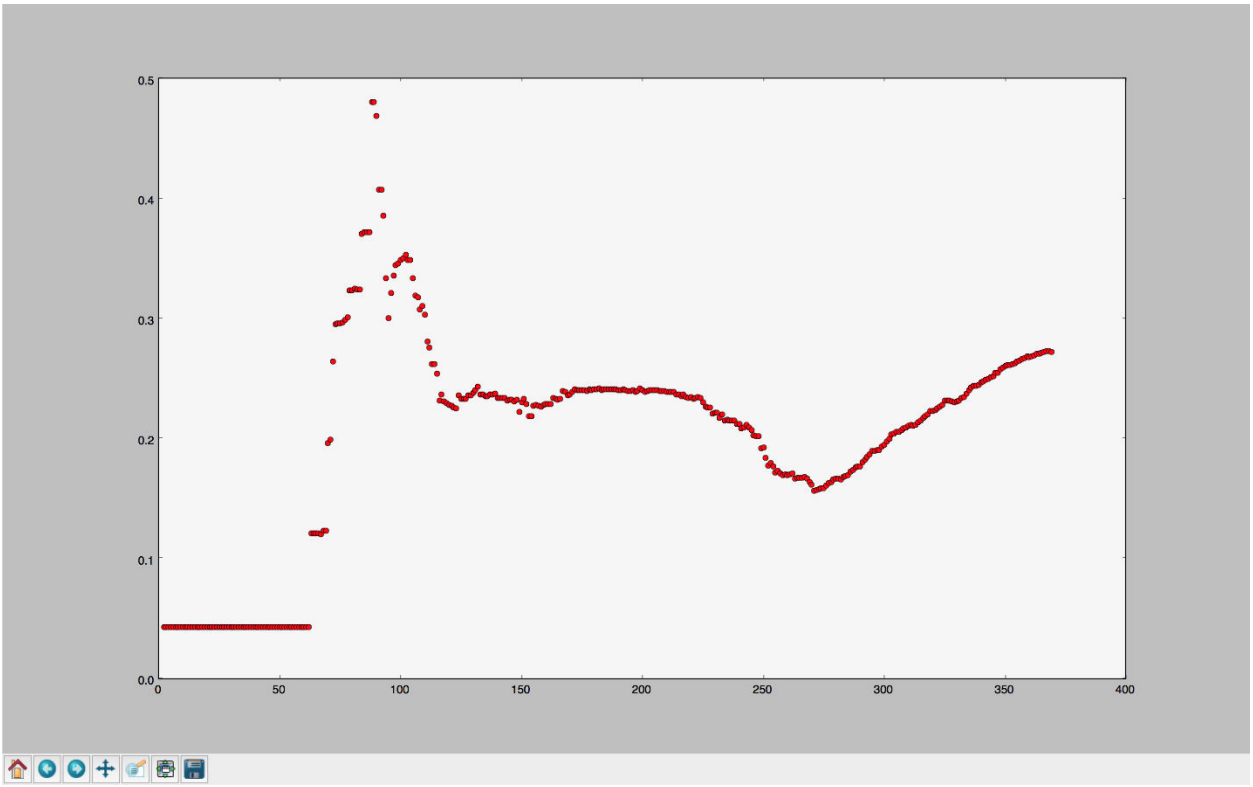
Graph Plots



Plot for accuracy vs number of attributes

The accuracy is 96 % when the number of attributes is less and is consistent until attributes is less than approximately 60. The accuracy starts to go down as number of attributes increases and is the lowest when attributes are close to 90. Then accuracy

shows a gradual increase and finally comes down to the range of 75 % when attributes are 370.



Plot of Mean squared Error vs Number of attributes

Mean squared Error is lowest when the number of attributes is less. Its graph is just the inverse of plot of accuracy vs number of attributes

3.3 Discussion:

1) Having a data with a lot of Dimensions will not give proper and accurate results (Curse of Dimensionality is Real !!).
2) Reducing the dimension with proper analysis and Dimension Reduction techniques (getting the uncorrelated principal variables) makes classification models more accurate.
3) Whereas, blindly reducing the dimensions can be counter- productive. That is, the generated model might have worse accuracy.
4) The classifiers like SVM with sigmoid kernel function, AdaBoost and Gradient Boost predicts with a high accuracy (~ 96% accuracy).

4. Related Work:

Customer Satisfaction can be judged based on their behavior and usage of a product and service. Patterns can be identified by clustering similar users and marking them as important to the company providing the service, so that these valuable customers (people identified as power users) don't churn at all.

5. Future Work:

1. To improve the result we can write a custom kernel for doing probabilistic modeling on the dataset.
2. Continue with the idea of probabilistic modelling to represent complex conditional probability graphs with Bayesian Networks.
3. Submit the results to Kaggle and be active in the competition till its completion.

6. References:

1. en.wikipedia.org
2. http://www.kdnuggets.com/2015/05/7-methods-data-dimensionality-reduction.html
3. http://www.saedsayad.com/naive_bayesian.htm

7. Library Used:

1. Pandas
2. Numpy
3. Sci-Kit learn

IDE used

Jupyter

8. Conclusion

To conclude, computing vectors for the input data of huge datasets and calculating probabilities can be a difficult task and will lead to longer running times while creating the model. With this project, it has been demonstrated that better results can be obtained using lesser attributes and much lesser mathematical computations. We demonstrated this by comparing various dimensionality reduction techniques. Also it is clear that accuracy is increased when correlated attributes are removed.