ITCS 6156: Machine Learning
Assignment 4: Support Vector Machines

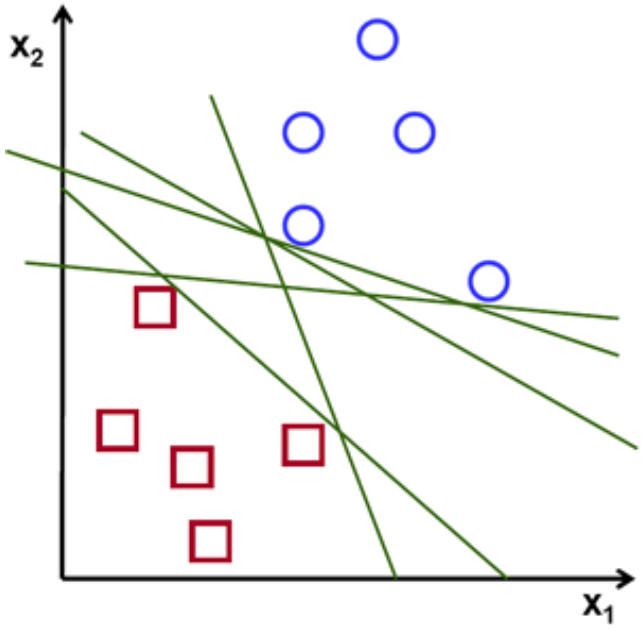
Submitted By: Ashwin Venkatesh Prabhu
UNCC ID: 800960400
Email: avenka11@uncc.edu

Collaborated with: Febin Zachariah
UNCC ID: 800961027
Email: fzachari@uncc.edu

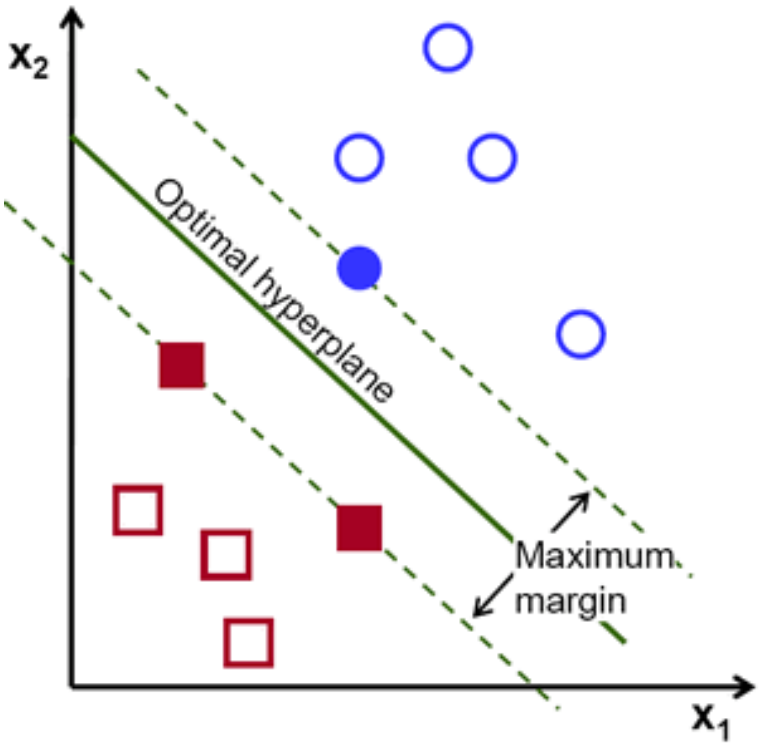
Support Vector Machines Algorithm

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. It is a supervised machine learning algorithm which can be used for both classification and regression problems, but is mainly used for classification problems. In this algorithm, we plot each data item as a point in the n-dimensional space, where n is the number of features, with the value of each feature being the value of the predictor coordinate. Then we perform classification by finding the hyperplane that differentiate the two classes very well.

For example, in the below figure, we deal with lines and points in the Cartesian plan instead of hyperplanes and vectors in the higher dimensional space. This is a simplification problem for a linearly separable set of two-dimensional points which belong to one of the two classes, but the same concepts can be applied to tasks where the examples to classify lie in a space whose dimension is higher than two.



In the above picture, we can see that there are multiple lines which offer solution to the problem. But the question here is, which one of them is better than others OR which one is best suited for the set of points in the plane. A line is considered bad if it passes too close to the points because it will be noise-sensitive and it will not generalize correctly. Therefore, our goal should be to find the line passing as far as possible from all points. Then the operation of support vector machines algorithm is based on finding the hyperplane that gives the largest minimum distance to the training examples. The optimal separating hyperplane maximizes the margin of the training data.



I have used "e1071" library in R to create an SVM model. The method used is svm(). The different relevant parameters for the method is given as below:

1) formula: the symbolic description of the formula to be fit
2) data: the data frame containing the variables in the model
3) type: Support Vector Machines can be used as a classification machine, or as a regression machine, or for novelty detection. SVM use hyperplanes to perform classification. While classifying using SVM, there are two types of SVM – C SVM and nu SVM. C and nu are regularization parameters which help implement a penalty on the misclassifications that are performed while separating the classes. Thus, helps in improving the accuracy of the output. C ranges from 0 to infinity and can be a bit hard to estimate and use. A modification to this is nu, which operates between 0 to 1 and represents the lower and upper bound on the number of examples that are support vectors that lie on the wrong side of the hyperplane. The default setting for the type is C-classification or eps-regression (depends on the whether the y is a factor or not), but can be explicitly overwritten by setting an explicit value. The available options are:
    a. C-classification
    b. nu-classification
    c. one-classification (for novelty detection)
    d. eps-regression
    e. nu-regression
4) kernel: used in training and predicting. Change the value of the following parameters depending on the kernel type
    a. linear
    b. polynomial

    c. radial basis
    d. sigmoid
5) gamma: parameter needed for all kernels except “linear”. The gamma parameter defines how far the influence of a single training example reaches with low values meaning far outreach and high values meaning closer outreach. The gamma parameters are the inverse of the radius of influence of samples selected by the model as support vectors. If the gamma is too large, the radius of the area of influence of the support vector is only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting. When gamma is very small, the model is too constrained and cannot capture the complexity or “shape” of the data. The region of influence of any selected support vector would include the whole training set.
6) cross: if an integer value k>0 is specified, a k-fold cross validation on the training data is performed to assess the quality of the model: the accuracy rate for classification and mean squared error for regression
7) cost: cost of constraints violation (default: 1) – this parameter tells SVM optimization how much you want to avoid misclassifying each training example. For large values of C, the optimization will choose smaller – margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger margin separating hyperplane, even if that hyperplane misclassifies more points. For very tiny values of C, should get misclassified examples, even if the hyperplane is linearly separable. In summary, small C makes the cost of misclassification low, and large C makes the cost of misclassification high.

Optical Recognition of Handwritten Digits

Implementation Details:

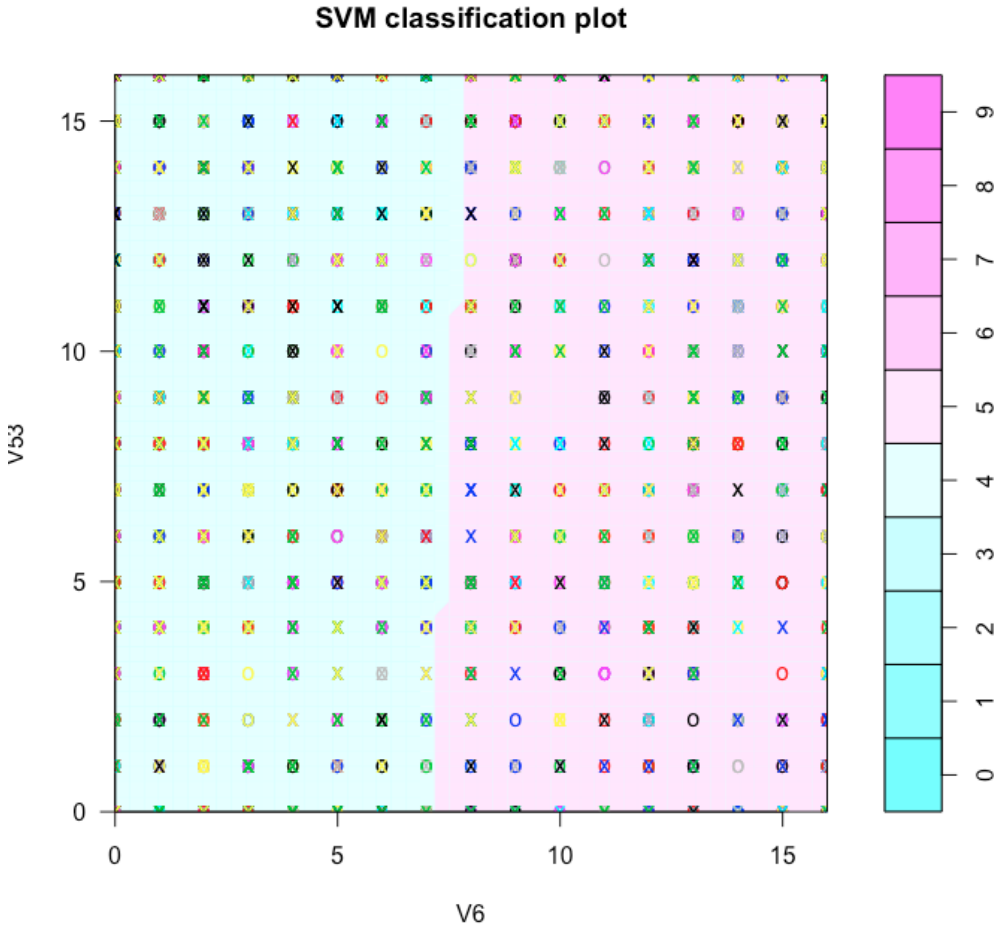
1) The program is written using R programming language
2) The program using “svm” method from the “e1071” library to execute support vector machine algorithm on the dataset.
3) The dataset for train and test data does not have a header. So, the headers for the feature variables are names V1 to V64. The header for class variable is named V65.
4) The train data is imported and stored in “train_data” variable
5) The test data is imported and stored in “test_data” variable
6) “svm” method is used execute support vector machine algorithm. As parameters to the method, the formula, train dataset, kernel (values can be linear, polynomial, etc.), type (values can be C-classification, nu-classification, etc.), cross (value is an integer k which specified the number of folds for cross validation), gamma (value is 1 / (data dimension))
7) Once the model is created, predict function is used to predict the value of class variable on the test dataset

Results:

I tried different values for the parameters, while creating an svm model. The value for cross is 10, which means the undergoes 10-fold cross-validation. The accuracy achieved for different models are given below:

Type	Kernel	C	Gamma	Model Accuracy	Accuracy
C-classification	linear	1	0.015625	97.88125	96.49416
C-classification	linear	100	10	97.82893	96.49416
C-classification	linear	100	100	98.01203	96.49416
C-classification	linear	0.1	100	97.95972	96.54981
C-classification	linear	0.0001	100	97.01805	95.21425
C-classification	linear	0.0001	1000	97.07036	95.21425
C-classification	linear	1000	1000	98.01203	96.49416
C-classification	polynomial	1	0.015625	99.05833	97.88536
C-classification	polynomial	100	10	98.92754	97.88536
C-classification	polynomial	100	100	98.87523	97.88536
C-classification	polynomial	0.1	100	98.90139	97.88536
C-classification	polynomial	0.0001	100	99.05833	97.88536
C-classification	polynomial	0.0001	1000	99.08449	97.88536
C-classification	polynomial	1000	1000	98.92754	97.88536
C-classification	sigmoid	1	0.015625	7.925713	10.18364
C-classification	sigmoid	100	10	8.37039	10.18364
C-classification	sigmoid	0.0001	100	8.37039	10.12799
nu-classification	linear	1	0.015625	94.97777	92.65442
nu-classification	linear	100	10	95.03008	92.65442
nu-classification	linear	100	100	94.97777	92.65442
nu-classification	linear	0.1	100	95.10855	92.65442
nu-classification	linear	0.0001	100	95.00392	92.65442
nu-classification	linear	0.0001	1000	94.89929	92.65442
nu-classification	linear	1000	1000	95.00392	92.65442
nu-classification	polynomial	1	0.015625	95.89328	93.4335
nu-classification	polynomial	100	100	95.91943	93.4335
nu-classification	polynomial	0.0001	1000	95.91943	93.4335
nu-classification	polynomial	1000	1000	95.78865	93.4335
one-classification	linear	100	0.015625	50.03924	10.68447
one-classification	linear	0.0001	1000	49.96076	10.68447

1) As we can see from the above table, the highest accuracy achieved is 99.08449%, which is for the model with **type = C-classification, kernel = polynomial, Cost = 0.0001, gamma = 1000**. Since, this is not a binary classification, it would be very hard to achieve a linearly separable model here, which is clear from the results, that a polynomial function is needed for accurate prediction for this dataset.
2) The lowest accuracy is for type = C-classification, kernel = sigmoid.
3) I expected nu-classification to give a better accuracy, as it was a more optimized classifier than the C-classification, but nu-classification gave a lower accuracy than the C-classification, even though 95% is still a good accuracy.
4) Accuracy is the least with type = one-classification, as seen in the above table
5) To visualize the data better, I conducted correlation analysis on the data to find the features which are highly correlated. On generating correlation matrix, I found two features which were highly correlated to the class variable, V65. They were, V6 and V53. Using these features, a graph was plotted to visualize the SVM classification.



The above plot is a scatter plot of the classification model by highlighting the classes and support vectors. The 'o' symbols in the plot represent the data and the 'x' symbol represent the support vectors.

Amazon Baby Product Review dataset

Implementation Details:

1) The program is written using R programming language
2) The program using "svm" method from the "e1071" library to execute support vector machine algorithm on the dataset.
3) The dataset has three headers namely, "name", "review", "rating".
4) Used Vader sentiment library from python to calculate the sentiment score for the train and test dataset.
5) The SentimentIntensityAnalyzer method from Vader sentiment library takes the text review as input and output four values, namely, "pos" (positive score of the review), "neg" (negative score of the review), "neu" (neutral score of the review), "compound" (normalized value of the sum of all sentiment scores)
6) After calculating the sentiment analysis for the train dataset (stored in "sentiment_train.csv"), and test dataset (stored in "sentiment_test.csv"), the new files are used as input for predictive analysis.
7) The train data is imported and stored in "train_data" variable
8) The test data is imported and stored in "test_data" variable
9) "svm" method is used execute support vector machine algorithm. As parameters to the method, the formula, train dataset, kernel (values can be linear, polynomial, etc.), type (values can be C-classification, nu-classification, etc.), cross (value is an integer k which specified the number of folds for cross validation), gamma (value is 1 / (data dimension))
10) Once the model is created, predict function is used to predict the value of class variable on the test dataset

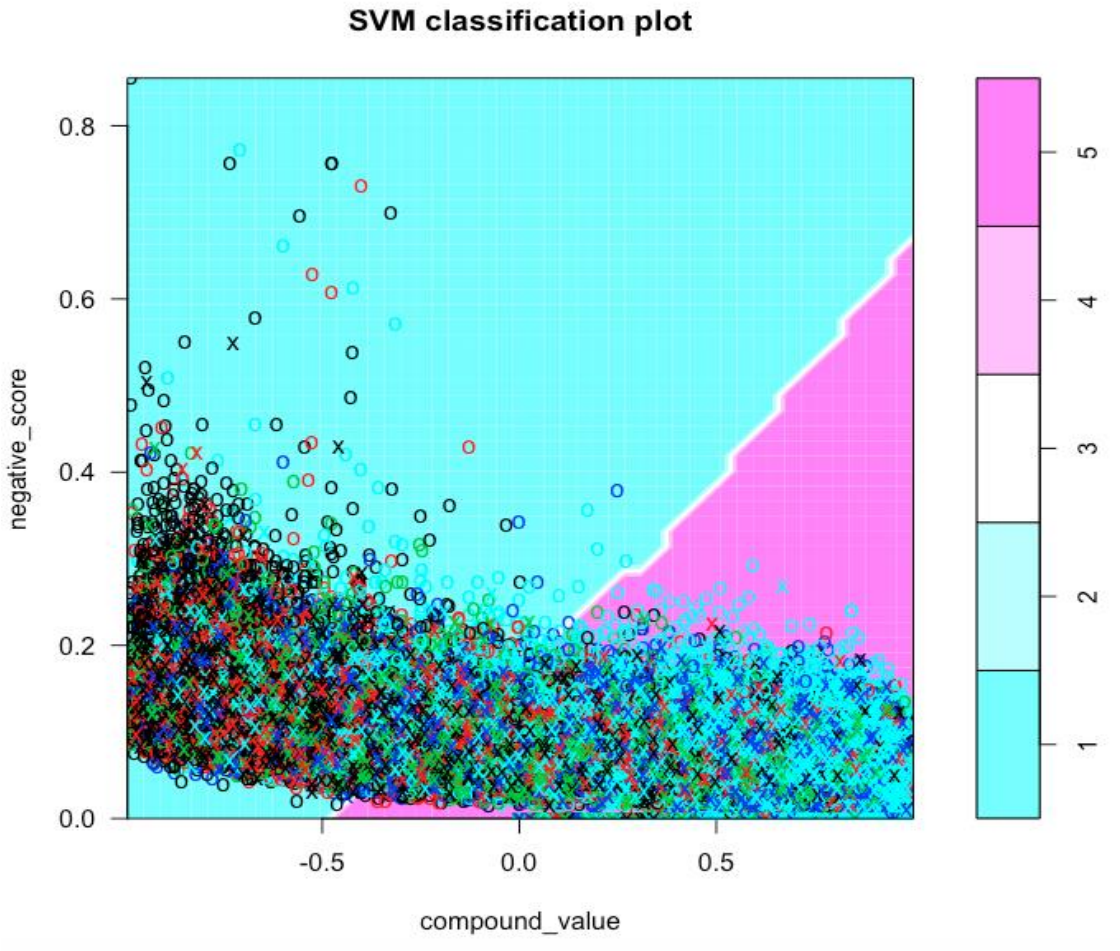
Results:

I tried different values for the parameters, while creating an svm model. The value for cross is 10, which means the undergoes 10-fold cross-validation. The accuracy achieved for different models are given below:

Type	Kernel	C	Gamma	Model Accuracy	Accuracy
C-classification	linear	1	0.25	60.62699	60.07573
C-classification	linear	0.1	10	60.74558	60.09208
C-classification	linear	0.01	100	60.75467	60.09208
C-classification	linear	0.001	100	60.72742	60.07846
C-classification	linear	0.0001	1000	58.75675	57.99711

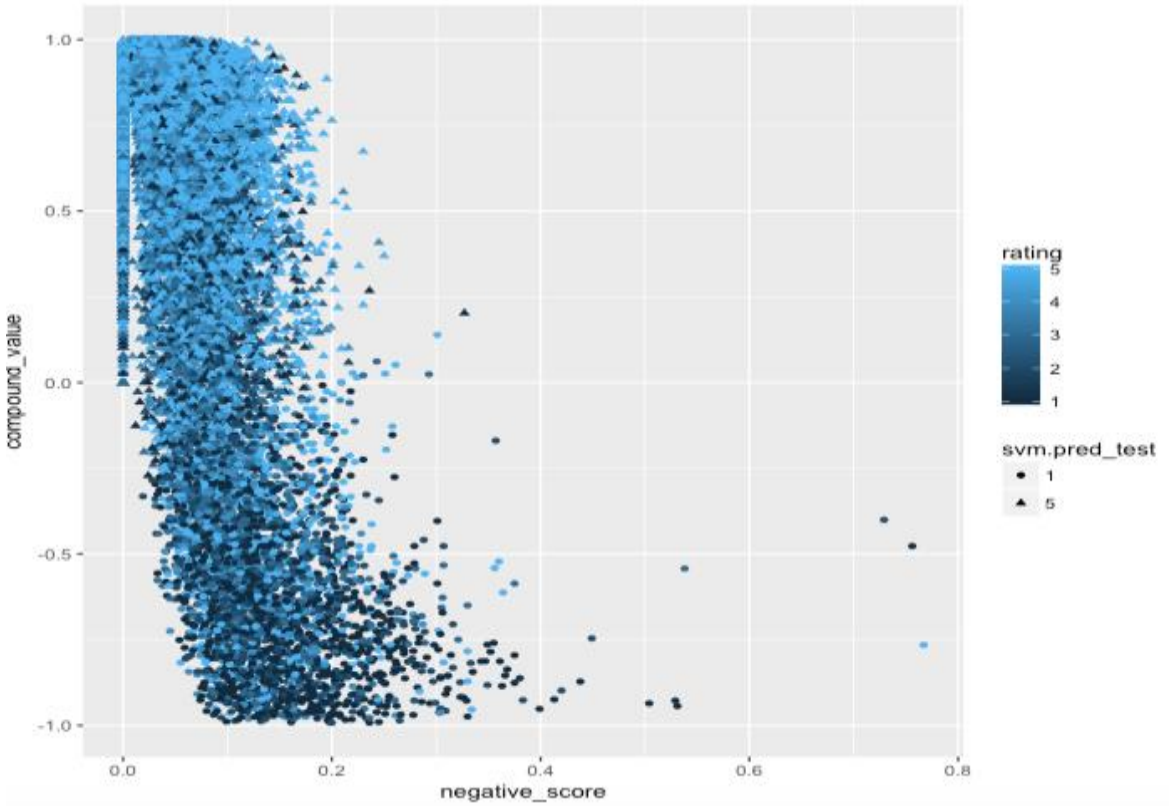
C-classification	polynomial	1	0.25	60.64569	60.0049
C-classification	sigmoid	1	0.25	50.40639	52.25434

1) As we can see from the above table, the highest accuracy achieved is 60.75 %, which is for the model with type = C-classification, kernel = linear, Cost = 0.01, gamma = 100.
2) While trying to run a nu-classification for kernel types linear, polynomial, sigmoid, I got an error which read *"specified nu is infeasible"*. Nu-SVM is a constrained formulation of SVM, equivalent to the original up for re-parametrization, which poses a hard bound on the allowed misclassification. If the bound cannot by satisfied, then the associated convex optimization problem becomes infeasible.
3) To visualize the data better, I conducted correlation analysis on the data to find the features which are highly correlated. On generating correlation matrix, I found three features which were highly correlated to the class variable, rating. They were, negative_score, positive_score and compound_value. Using these features, a graph was plotted to visualize the SVM classification based on negative score and compound value.

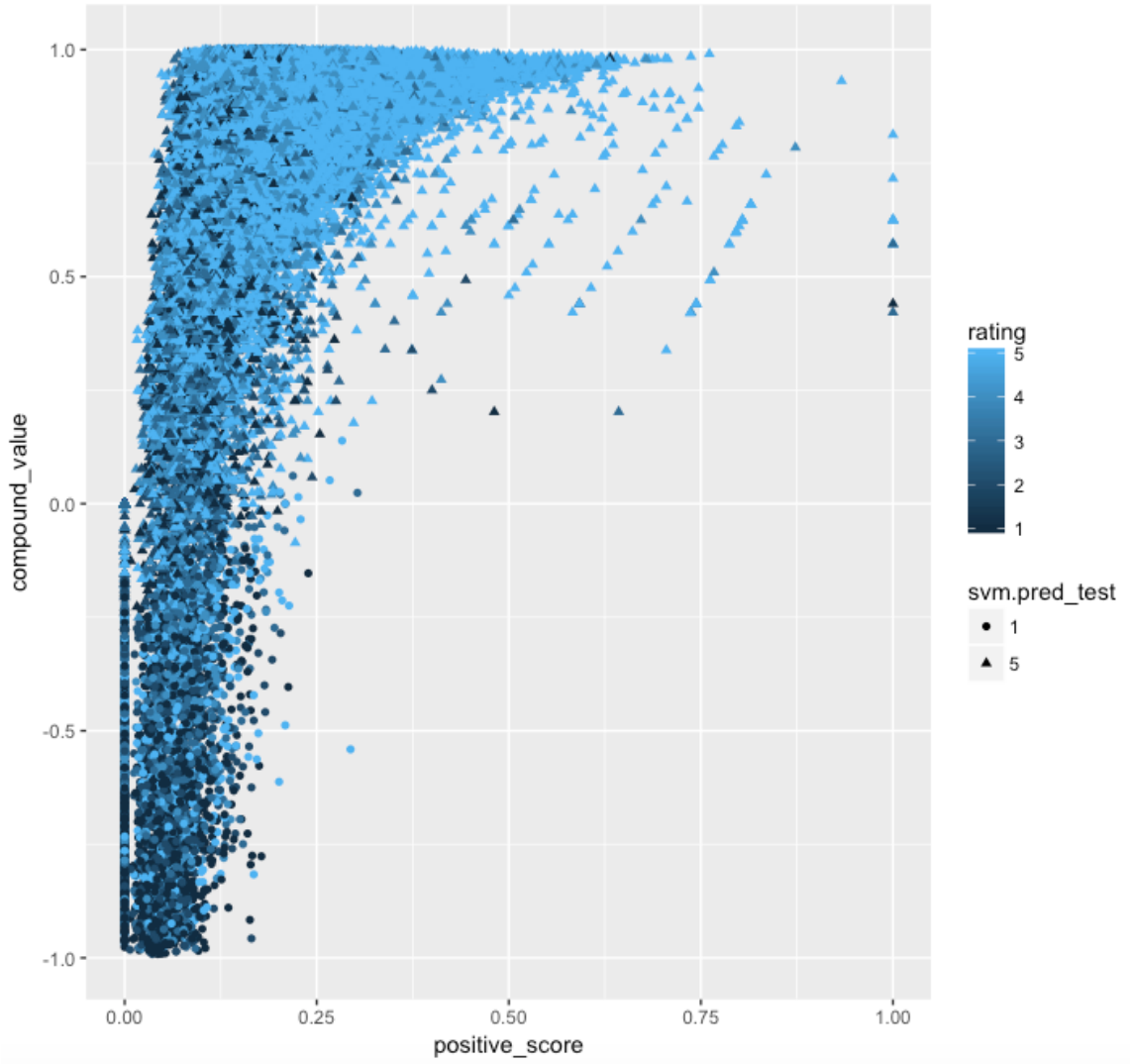


4) The below two graphs visualize the prediction done by the model. The model considers the ratings 2, 3, 4 as rating 5. Hence, classifying the ratings into two categories (high rating – rating 5 and low rating – rating 1). Our dataset has enormous number of rating 5, so this misclassification does not add much to the misclassification rate.

Negative_score vs compound_value



Positive_score vs Compound_value:



References:

1) http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
2) http://vassarstats.net/matrix2.html
3) https://cran.r-project.org/web/packages/e1071/index.html
4) https://escience.rpi.edu/data/DA/svmbasic_notes.pdf