MODEL BASED FEATURE SELECTION FOR VIEW MAINTENANCE MODELS OF DATA WAREHOUSE

Rolly Gupta[a], Sangeeta Sabharwal [b], Anjana Gosain [c]

[a]Research Scholar, Department of COE, Netaji Subhas Institute of Technology, Delhi University, Delhi, India
[b]Professor, Department of COE, Netaji Subhas Institute of Technology, Delhi University, Delhi, India
[c]Professor, University School of Information Technology,GGS Indraprastha University, Delhi, India

***Abstract*—** A data warehouse consists of multiple views that can be accessed by the means of queries. One of the vital decisions in designing a data warehouse is the selection of materialized views for the purpose of an efficient decision making. The search space for the selection method of materialized views is exponentially very large enough; therefore, heuristics have been used for searching a small fraction of the space to get into a near optimal solution. Feature selection is of immense importance in the field of statistical data analysis. The dimensionality of data makes the testing and training process of general classification methods more difficult. Materialization on the reduced set of attributes reduces the computation time and also helps to make the patterns easier to understand. In this paper, we will consider some classification and regression approach to explain the importance of model-based feature selection. Finally, all the different classifiers used for feature selection are also compared and accordingly best classifier for respective datasets is observed.

Keywords—Dimension reduction, Classification, Regression

I. INTRODUCTION

A Data Warehouse (DW) can be defined as a repository of integrated information for querying and analysis of data. Data Warehouse accounts for integration of data from various information sources, possibly large, multiple and distributed heterogeneous databases. Materialized view selection is one of the issues related to DW [18]. The related work is presented in [17], [8], [2], [7], [l0], [13]. The problem is NP-complete [8]. In this approach, it is used to extract and integrate information of relevance from each source and then store them in a centralized repository. When a user query is evaluated, the intermediate virtual result is generated; called as view. When the view is stored (in advance) in repository, it is termed as called materialized view. This arises following situations:

- To materialize all the views in the DW; high performance and high maintenance cost.
- Maintain all views virtually; poorest performance and lowest maintenance cost;
- Maintain some materialized views and some virtual views; optimal performance and maintenance.

Our problem is to materialize views based on model-based feature importance for multiple queries, in order to achieve minimization of query and maintenance cost. Database researchers had made use of heuristics for materialized views selection problem [9], [7], [6], [8], [3], [5], [16]. In this paper, we explore the use model-based feature importance to a number of data sets in order to minimize cost of queries and maintenance.

We are performing the selection and classification of data in order to remove the 'curse of dimensionality'[5–7] in data analysis, especially in data warehouse as they are characterized by relatively few instances and are presented in a high-dimensional feature space. The irrelevant features lead to insufficient classification accuracy and add extra difficulties in finding potentially useful knowledge [8,9] etc. Therefore, appropriate model-based feature selection can reduce the requirements of measurement and storage for minimizing the cost in database storage and management [10,13].

In terms of classification, the main aim of model based feature selection is to search for an optimal feature subset from the initial feature set. This selection will lead to improved classification performance and efficiency for generating classification model. During the past years, extensive research has been conducted from multidisciplinary fields including pattern recognition, machine learning, statistics, and data mining [14,15]. Many feature selection methods have been developed during the past years. This methods calls for searching significant features by taking into account the characteristics of each individual feature. It uses an independent test such as the information entropy and statistical dependence test.

Moreover, section 2 in the paper briefs about the model-based approach for feature selection. The proposed approach is described and implemented in section 3 followed by the classifier comparison of results and then the conclusion in section 5 respectively.

II. Model-based feature selection

A feature selection technique named as forward feature selection, which basically extracts the most important features required for the optimal value was discussed in our previous paper [5]. But, it had one caveat though—i.e. large time complexity. In order to circumvent the issue, feature importance can directly be obtained from the used model being trained. This techniques extracts important features from the model parameters. In this paper, we will consider some classification and regression algorithms to explain model-based feature importance. Different classification techniques are compared using datasets from University of California, Irvine (UCI) Machine Learning Repository. Accuracy of result by each classifier is observed. Finally, all the different classifiers used for feature selection are also compared with each other and accordingly best classifier for respective datasets is observed for interpretation. The workout done as is illuminated as:

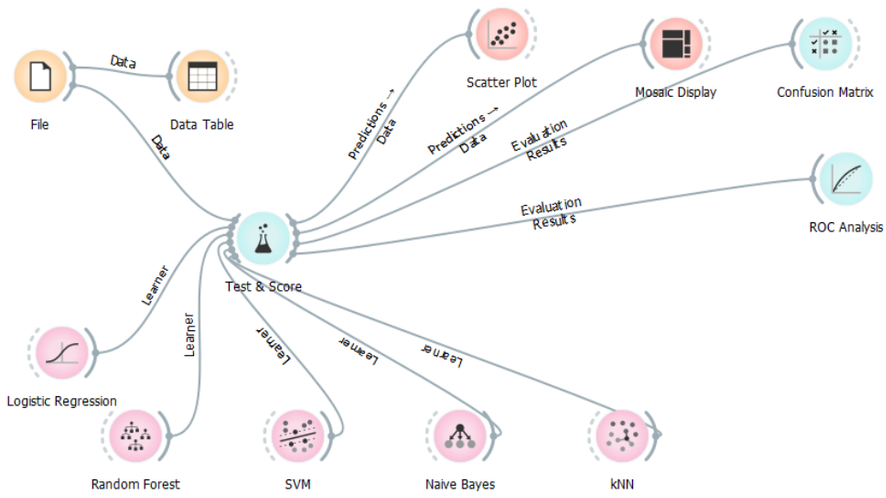


Figure 1: Workout Plan

III. CLASSIFICATION MODELS AND THEIR RESULT SET

3.1. Logistic Regression

An inherently binary classification algorithmic rule, it tries to seek out the simplest hyperplane in k-dimensional area that separates the two categories, minimizing provision loss.

$$L(y_i, y_i^*) = \frac{-1}{n}\sum_{i=1}^{n} log\left(1 + e^{-y_i(w^T x_i + b)}\right)$$

y_i : Label of point $i \in \{-1, +1\}$

y_i^* : Model prediction $= w^T x_i + b$

w : Weight vector

x_i : Input vector

b : Bias / Intercept

$$w^T x_i = \sum_{j=1}^{k} w_j x_{ij}$$

Figure 2: Logistic loss expression

The k dimensional weight vector is accustomed get feature importance. massive positive values of w_j signify higher importance of the jth feature within the prediction of positive category. giant negative values signify higher importance within the prediction of negative category. this may be seen from the expression of logistical loss. SGD reduces loss by setting learning massive positive weights for features vital in predicting an information purpose to belong to the positive category and equally for negative category.

This Logistic Regression concept is implemented using UCI dataset, providing the plot and confusion matrix as below:

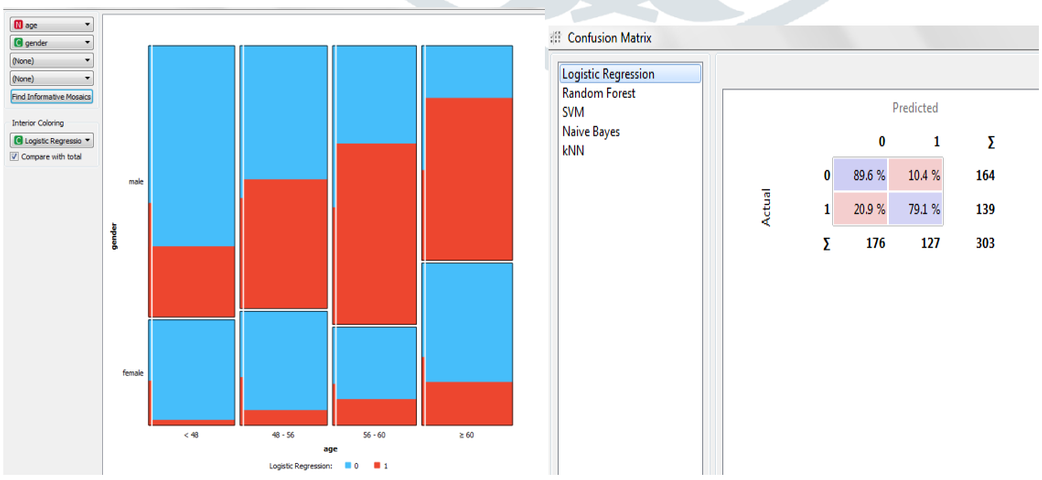


Figure 3: Logistic Regression Result Set

3.2. Random Forest Classifier

Random forest is a model using decision trees as the base learners. The base learners are having high variance, but low bias models. The variance of the overall model can be reduced by aggregating the decisions, taken by all base learners for predicting the response variable. The idea relies that each of base learner learns a different aspect of the data. This is achieved by the row and column sampling. In a classification setting, the aggregation can be done by taking the majority vote.

At each node of the decision tree, the feature used for splitting the dataset is decided on the basis of information gain (I.G.) or more computationally Gini impurity reduction. The feature which maximizes I.G. is selected as splitting feature. Data is then divided among its children according to the value of the splitting feature. If the feature used is categorical, then data belonging to the each category of splitting feature goes, to a separate child. In case of a numerical feature, the best threshold

value (the one used to decide in favour of this feature used as splitting feature) is used to split the data into two parts, each going to the one child.

$$H(D) = -\sum_{i=1}^{k} p_i log(p_i)$$

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

D : Dataset
k : Number of classes
p_i : Probability of a point belonging to class i

$$IG(D, f_j) = H(D) - \sum_{m=1}^{M} \frac{|D_m|}{|D|} H(D_m)$$

M : number of datasets obtained after splitting
D_m : m^{th} Dataset obtained after splitting

Fig 4: Random Forest Equation

- f_j is chosen as splitting feature

Information Gain due to the features summed across all levels of decision tree determines the importance of its feature. This can also be seen that at every node splitting is done on the feature, which maximizes Information Gain. Random forests comprises of multiple decision trees, thus the feature importance of feature j is the normalized sum of I.G. caused by feature j across all the trees.

This Random Forest concept is implemented using UCI dataset, providing the plot and confusion matrix as below:

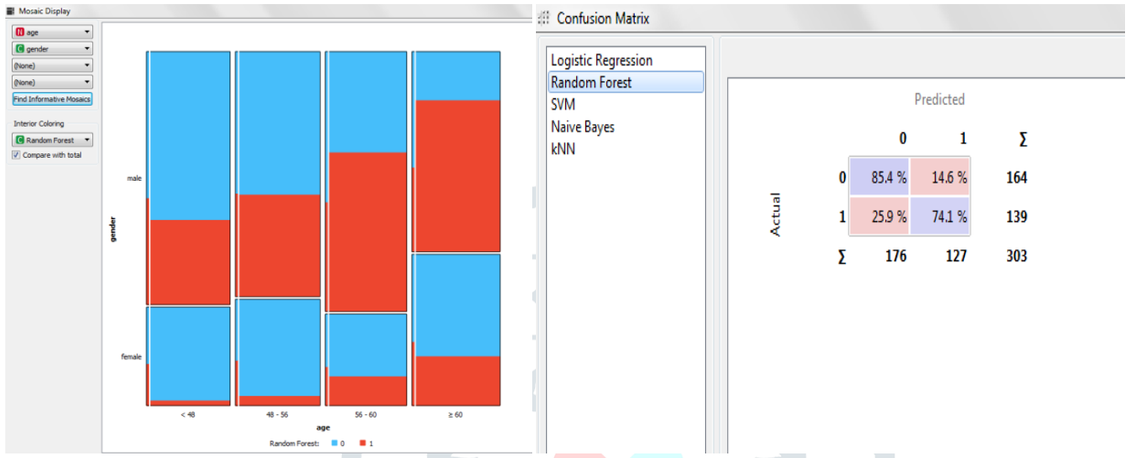


Figure 5: Random Forest Result Set

3.3. Support Vector Machine (SVM)

A Support Vector Machine (SVM) performs the classification by finding the hyperplane which maximizes the margin between the two set of classes. The vectors or (cases) which define the hyperplane are the support vectors.

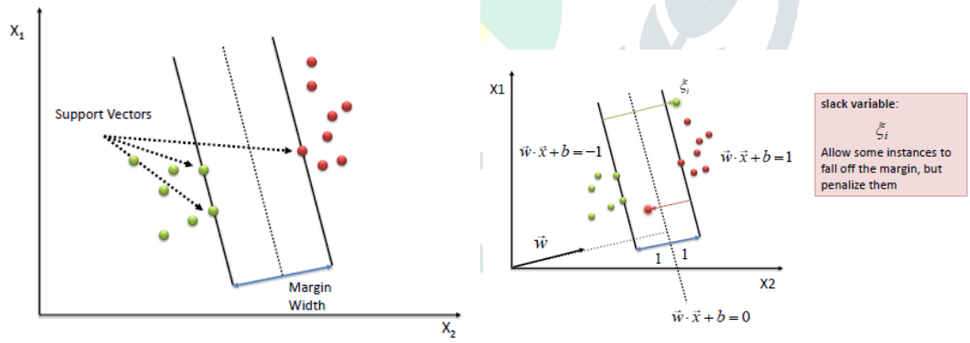


Figure 6: SVM Graph I

The importance of SVM is that if the data is linearly separable, then there is a unique global minimum value. A perfect SVM analysis produces a hyperplane that completely separates the vectors (cases) into two non-overlapping classes/vectors. However, perfect separation is not possible, or may result in a model with many cases which the model does not classify them correctly. In this circumstances, SVM finds the hyperplane which maximizes margin and minimizes misclassifications.

The algorithm helps to maintain the slack variable to zero while maximizing the margin. However, it does not minimizes the number of misclassifications (NP-complete problem), but the total of distances from the margin hyperplanes.

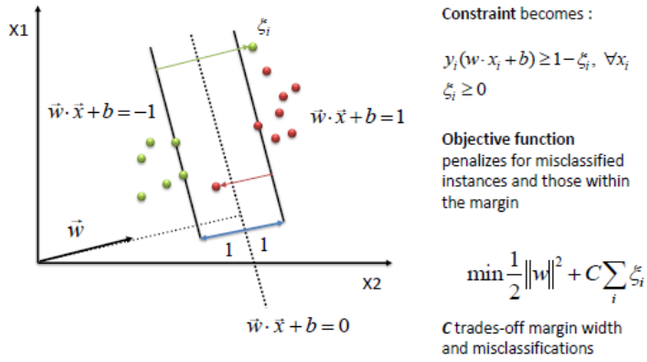


Figure 7: SVM Graph II

The simplest way to separate the two groups of data is with the straight line (1-dimension), flat plane (2 -dimensions) or an N-dimensional hyper plane. However, there are some situations where nonlinear region can separate groups more efficiently. SVM handles this by use of a kernel function (nonlinear) for mapping the data into a different space where a hyperplane (linear) cannot be used for separation. It means a non-linear function is being learned by a linear learning machine in a high-dimensional feature space while the capacity of system is controlled by the parameter which does not depend on the dimensionality of the space. This is referred as *kernel trick* which means the kernel function transforms the data into a higher dimensional feature space, making it possible to perform the linear separation as well.

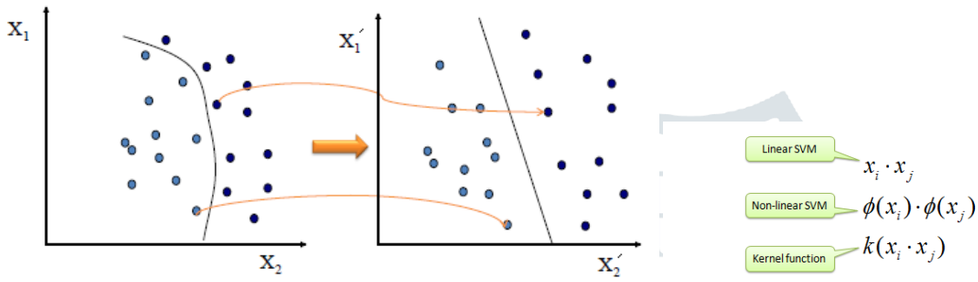


Figure 8: SVM Graph III

This SVM concept is implemented using UCI dataset, providing the plot and confusion matrix as below:

.

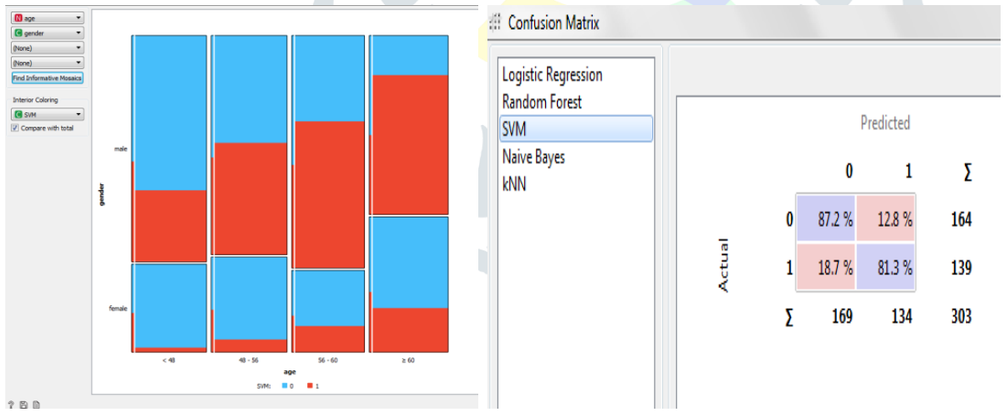


Figure 9: Support Vector Machine Result Set

3.4. Naive Bayesian

The Naive Bayesian classifier is precisely based upon the Bayes' theorem with the assumption of independence between the predictors. A Naive Bayesian model is more easy to build. There is no complicated iterative parameter estimation, which concise it useful for large datasets. The Naive Bayesian classifier often does surprisingly well; and is widely used because it outperforms more sophisticated classification methods.

Bayes theorem gives a way of calculating the posterior probability, $P(c|x)$, from $P(c)$, $P(x)$, and $P(x|c)$.
Naive Bayes classifier assumes that the effect of the value of predictor (x) on given class (c) is independent of values of ther other predictors. This assumption is referred as class conditional independence.

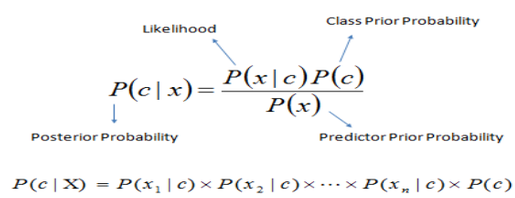


Figure 10: Naïve Bayesian Equation

$P(c|x)$ is posterior probability of the *class* (*target*) given the *predictor* (*attribute*).
$P(c)$ is prior probability of the *class*.

$P(x|c)$ is likelihood which is the probability of the *predictor* given *class*.
$P(x)$ is prior probability of the *predictor*.

In ZeroR model there are no predictors, in OneR model there is only a single best predictor, while naive Bayesian includes all the predictors using Bayes' rule and independence assumptions between predictors. This Naive Bayesian classifier concept is implemented using UCI dataset, providing the plot and confusion matrix as below:

.

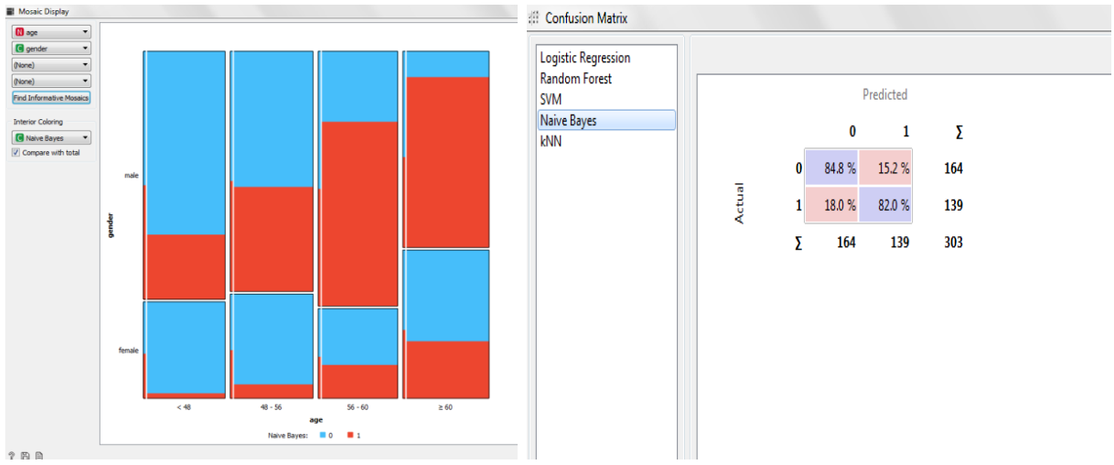


Figure 11: Naïve Bayesian Result Set

3.5. K nearest neighbors

K nearest neighbors is a very simple algorithm. It stores all available cases and then classifies the new cases, based on the similarity measure (e.g., distance functions). KNN is being used in statistical estimation and pattern recognition, as a non-parametric technique. A case in KNN is mainly classified by a majority vote of its neighbours. While the case is being assigned to the class, most common amongst its K nearest neighbours; measured by use of a distance function. If the value of K = 1, then the case is simply assigned to the class of its nearest neighbor.

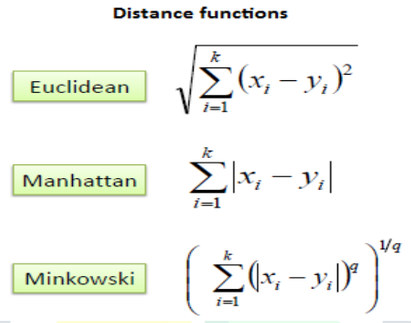


Figure 12: Distance Functions equations

Choosing optimal value for K is best done by first the inspection of the data. In general, a large K value is more precise, as it reduces the overall noise but there is no certainity. Cross-validation is another way to retrospectively determine the K value by using an independent dataset for validating the K value.

This KNN classifier concept is implemented using UCI dataset, providing the plot and confusion matrix as below:

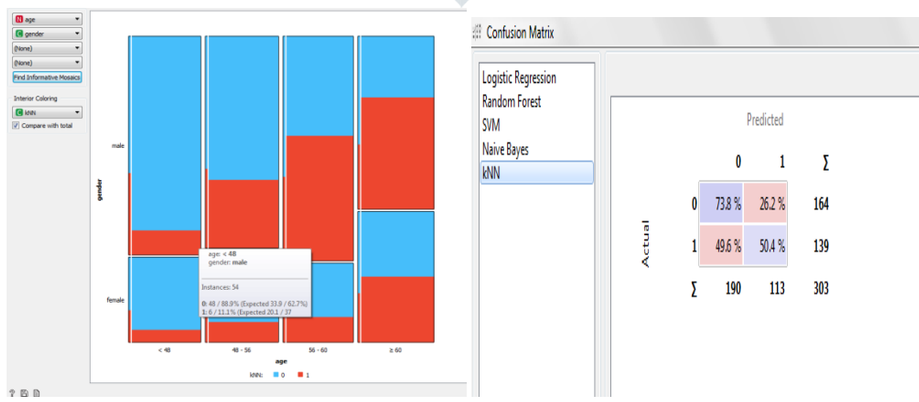
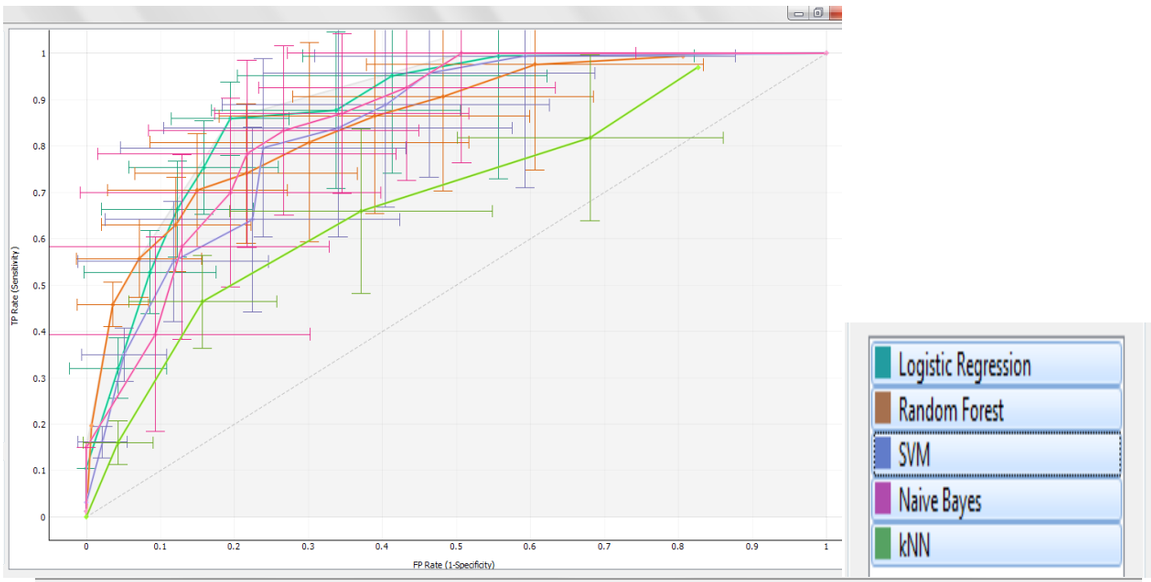


Figure 13: KNN Result Set

IV. COMPARISION OF CLASSIFICATION MODELS

In this study we have compared the performance of various classifiers used for feature selection. Data sets from benchmark Data set (UCI) are used for experimentation. Numbers of cross-folds in each case are 10. In general it is found that the performance of classification techniques varies with the same data sets. It shows the following results:



Sampling
- Cross validation
  - Number of folds: 10
  - Stratified
- Cross validation by feature
- Random sampling
  - Repeat train/test: 10
  - Training set size: 66 %
  - Stratified
- Leave one out
- Test on train data
- Test on test data

Target Class: (Average over classes)

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
kNN	0.699	0.630	0.625	0.629	0.630
SVM	0.893	0.845	0.845	0.845	0.845
Random Forest	0.891	0.802	0.801	0.803	0.802
Naive Bayes	0.908	0.835	0.835	0.835	0.835
Logistic Regression	0.909	0.848	0.847	0.849	0.848

Figure 14 & 15: Comparison Result of Classification Models

Confusion matrix plotted above in each classifier is a summary of the prediction results on a classification problem. The number of correct and incorrect predictions are then summarized with count values and then broken down by each class. It gives information about the errors being made by a classifier but also the types of errors that are being made. Most the performance measures are computed by using the confusion matrix as explained below.

AUC is used for area under the ROC curve. It is used in classification analysis to determine which of the used models best predicts the classes. The true positive rates are being plotted against the false positive rates. It can be interpreted that Logistic Regression (LR) provides high accuracy then the other model classifiers. Accuracy is merely a great measure but only when there exists a symmetric datasets, i.e. the values of false positives and the false negatives are almost the same. Therefore, some other parameters also help to evaluate the performance of model classifier. For LR model, we have 0.848 which means our model is approx. 85% accurate while SVM is 84% accurate.

Precision is defined as the ratio of correctly predicted positive observations to total predicted positive observations. High precision value relates to the low false positive rate. We have 0.849 precision which is pretty good for LR model. Recall is defined as the ratio of correctly predicted positive observations to all observations in actual class. We have recall value of 0.848 which is good for LR model.

F1 Score is termed as the weighted average of Precision and Recall. Therefore, this score takes both the false positives and the false negatives into account. Intuitively, it is not as easy to understand as the accuracy, but F1 is usually more useful than the accuracy, especially if there is uneven class distribution. Also, Accuracy works best if the false positives and the false egatives have similar cost. But, if the cost of false positives and false negatives are very different, both Precision and Recall are considered. In LR, F1 score is 0.847. SVM F1 score is 0.845.

V. CONCLUSION

In this paper, we explored the use of model-based feature selection method for detecting irrelevant updates to base relations of materialized view. This model-based feature selection method increases the probability of catching the irrelevant updates. It performs designing of feature subsets judiciously in order to capture the features in the base relations. This is done at the cost of using more space. Based on the model-based feature selection method, data and models are experimentally evaluated. Our experimental study indicates that the model based approach delivers better solutions than some heuristics. The method can catch most (or all) irrelevant updates to base relations that are missed by the traditional method. Thus, the fraction of irrelevant updates is non-negligible; therefore the load on the database due to materialized view maintenance can be significantly reduced. In addition, it can be interpreted that Logistic Regression (LR) provides high accuracy then the other model. For LR model, we have 0.848 which means our model is approx. 85% accurate while SVM is 84% accurate.

References

[1] E. Baralis, S. Paraboschi, and E. Teniente, "Materialized view selection in a multidimensional database", proceedzngs of the 23rd VLDB Conference, Athens, Greece, pages 156-165, 1997.

[2] K. Bennett, M. C. Ferris, and Y. Ioannidis, "A genetic algorithm for database query optimization", Technical Report TRlOO.4, University of Wisconsin, Madison(WI), 1991.

[3] D. E. Goldberg, "Genetic algorithms in search, optimizationand machine learning", *Addison Wesley,* Reading(MA), 1989.

[4] M. Gregory, "Genetic algorithm optimisation of distributed database queries", *Proc. of ICEC'98,* pages 271-276,1998.

[5] A.Gosain, S.Sabharwal, R. Gupta, "An Efficient Feature Selection Approach for Materialized Views", Proceedings of ICCCCM2013, Allahabad, India.

[6] H. Gupta, "Selection of views to materialize in a data warehouse", Proceedings of the International Conference on Data Engineering, Burmingham, *U.K.,* pages 98-112, April, 1997.

[7] H. Gupta, V. Harinarayan, and A. Rajaraman, "Index selection for olap", *Proceedings of* the International Conference on Data Engineering, pp. 208-219, 1997.

[8] H. Gupta and I. S. Mumick, "Selection of views to materialize under a maintenance cost constraint", Proceedings of the International Conference on Data Engineering, 1998.

[9] V. Harinarayan, A. Rajaraman, and J. D. Ullman, "Implementing data cubes efficiently", ACM SIGMOD International Conference on Management of *Data,* pages 205-227, 1996.

[10] W. J. Labio, D. Quass, and B. Adelberg, "Physical database design for data warehouses", Proceedings of the International Conference on Data Engineering, pages 277-288, 1997.

[11] T. K. Sel1is, "Multiple-query optimization", *ACM* Transactions on Database Systems, 13(1):23-52, March 1988.

[12] Msachusetts Institute of Technology. Galib: A c++ genetic algorithms library. http://lancet.mit.edu/galib-2.4/GAlib.html, v2.4.

[13] K. A. Ross, D. Srivastava, and S. Sudarshan, "Materialized view maintenance and integrity constraint checking: Trading space for time", *Proceedings of the ACM SIGMOD,* pages 447-458, 1996.

[14] S. Timos, S. Kyuseok and D. Nau, "Improvements on a heuristic algorithm for multiplequery optimization", Data and Knowledge Engineering, 12:197-222, 1994.

[15] R. E Smith, D. E. Goldberg, and J. A. Earickson, "SGA-C: A C-language implementation of simple genetic algorithm", *TCGA Report* ***No.*** *91002,* MarchJ994.

[16] M. Steinbrunn, G. Moerkotte, and A. Kemper, "Heuristic and randomized optimization for the join ordering problem", *VLDB,* 6(3):191-208, 1997.

[17] D. Theodoratos and T. Sellis, "Data warehouse configuration", Proceedings of the 23rd VLDB Conference Athens, Greece, 1997, pages 126-135, 1997.

[18] J. Widom, "Research problems in data warehouse", proceedinas of 4th International Conference on Information and Knowledge Management, pages 25-30, 1995.

[19] J. Yang, K. Karlapalem, and Q. Li, "Algorithm for materialized view design in data warehousing environment", *VLDB '97,* pages 20-40, 1997.

[20] H. Witten, E. Frank. 2005. Data Mining: Practical machine learning tools and techniques. 2nd Edition, Morgan Kaufmann, San Francisco.

[21] J. Han And M. Kamber. 2001. Data Mining: Concepts and Techniques. San Francisco, Morgan Kauffmann Publishers.

[22] Jennifer G. Dy. 2004. Feature Selection for Unsupervised Learning, Journal of Machine Learning, pp845-889.

[23] M.A.Jayaram, Asha Gowda Karegowda.2007. Integrating Decision Tree and ANN for Categorization of Diabetics Data. International Conference on Computer Aided Engineering, December 13-15, 2007, IIT Madras, Chennai, India.

[24] M. Dash, K. Choi, P. Scheuermann, H. Liu, "Feature Selection for Clustering – a Filter Solution", In Proceedings of the Second International Conference on Data Mining, 2002.

[25] S. Doraisamy ,S. Golzari, N. M. Norowi, Md. Nasir, B Sulaiman, N. I. Udzir, "A Study on Feature Selection and ClassificationTechniques for Automatic Genre Classification of Traditional Malay Music" ismir2008.ismir.net/papers/ISMIR2008 256.pdf(2008).

[26] V. Rotz, and T. Lange, "Feature Selection in Clustering Problems", In Advances in Neural Information Processing Systems 16, 2003.

[27] Y.Saeys, I.Inza, and P. Larr, A. N. Naga,, "A review of feature selection techniquesin bioinformatics, Bioinformatics, 23(19),, pp.2507-2517., 2007.