Which Category would beaucoup Members? Prediction of categories that interests more members.

Abstract— This paper addresses the problem associated with predicting the categories that eventuate the members learning from their previous attendance. The data is gathered from Meetup, a social networking platform which consists of vast historical data using which we can build a recommender system using machine learning algorithms which learns the features of the members, which helps in understanding the interests of attendees of the events based on their previous attendance for various events or groups. This eventuates a recommender system by application of different models such as support vector machine, random forest, KNN and Naïve Bayes and yet found random forest as our champion model which boosts the high-level accuracy.

***Index Terms*— Machine Learning, Support Vector Machine, Random Forest, K Nearest Neighbor, Naïve Bayes Classifier, Meetup**

I. INTRODUCTION

THE Social Media platforms like Facebook, LinkedIn, Twitter helps people (sub)consciously represent themselves with their interests in a way which is appropriate to their intended viewers. Similarly[18], Meetup is a social networking platform where a user can meet the audience, who happens to possess the same interest as his/her by joining groups and attending events that interest them eventuating a common platform for its members. As of 2017, the online networking platform Meetup has gained 3 million registered members from over 182 countries [19].

Researchers are in continuous process to understand the interests and social behavior of the members. The potential online data is very vast which benefits the researchers for identifying the ground truth models in Data Mining [15].

Data Mining has moved way beyond from predicting models into recommenders. Text, language, image, artificial intelligence and deep learning projects which are non-linear, will require a vast data which must be gathered, explored and prepped to understand the business better [15].

In this paper, we intend to gain more knowledge and build a recommender model by gaining the insights of a social networking platform, Meetup which has generated vast data over the years that can be analyzed to build a recommender system using advanced data mining techniques. We demonstrate various data mining models and check the accuracy of all the models and select the champion model with high accuracy rate.

II. RELATED WORK

Prediction of the events is discussed by many authors. Earlier it was experimented on networks such as the telecommunications to find out the failures of any of the instruments used. The alarm messages that were produced in the network were used to find it [1]. These methods were like the time series prediction problems that can be found in the statistics. By recording the earlier observations, the next observation or event can be predicted [2]. The machine learning algorithm used was *'Time weaver',* It uses two steps for the learning process 1). Identifying prediction patterns takes set theory into account for searching the patterns (genetic algorithm is used for identifying patterns) 2). Generating prediction rules, an ordered list of the patterns contained in these rules (greedy algorithm was used for prediction of rules) [1].

Like in the previous study, this event prediction was on the computer networks, where it was monitored continuously for a period of one month. The events were categorized as 'harmless', 'warning', 'minor', 'critical' and 'fatal'. This study was to predict the severe events i.e. critical and fatal ones. Also, these events occur very rarely in the network. A vector having the two dimensions is used to denote the events e = (timestamp, type), the timestamp gives the data on when it occurred, and the type tells about the severity. The SVD (Singular Value Decomposition) method is used for the feature selection and SVM(Support Vector Machine) was used for the classification of the training data and later predict the events [3]

Unlike the previous studies on event predictions in the computer and telecommunication network. There is a special kind of the social networks based on the events and locations. We are in the Event-Based Social Networks(EBSN). For example, ‘meetup’ [4], it is an online website where people create new events, discuss them. They may be formal ones such as ‘R programming, ‘Tech meetup’ and the informal ones as ‘movies’, ’sports’ etc. In the online platform, they discuss regarding the events and the members or user comment on these events to show their interest in whether they are attending them. After this, they gather in a place where the event is organized, and it is called as an offline meeting or interaction. With this online and offline meetup data, they concluded that most of the users (70%) attending the events live within the 10-mile radius [4]. Clustering of the online and offline ESBN and comparing the results with the LSBN was done in the study.

A group recommendation system for the ESBN was proposed in [5] for the new user who would like to join a group. People when they first create an account and enter the networking site, they will have confusions about the groups because of their variety and large volumes. Some users just want someone to guide them or to be told that this is the best group to join based on the interests. For the group recommendation, they used a new method of ‘pairwise tag enhanced & matrix factorization’ which utilizes the features like location, social interests and the pattern of the implicit communication between the users. The interaction between the users and the groups is put into a model using the matrix factorization method [5]

Once the user is joined to a group, they will be looking for events to join. An event recommendation system is proposed in [6] and this is using the Bayesian probability and Matrix factorization methods. It takes inputs of the member social profile, comments and the plans made up for the events and gives out the event recommendation. Many other studies were conducted in event detection from the huge amounts of the text data available from the social networks like Twitter, Instagram, and Facebook. We can get to know the on-going events by the discussion of the user through the natural language processing and data mining [7]. But this is not so relevant to our work.

For any of the recommendation system to work, there should be some of the users that have already registered for it. If the events are not registered by the users, the recommendation system will not work as it is the problem of the cold start. This issue is solved using the collective Bayesian factorization [8]. A bit of the modification is done using the poission factorization with the Bayesian factorization to solve the same cold start problems and events recommendation on the relationships between the users that have already scheduled the events is discussed in [9].

All the above-discussed methods do only one type of recommendation at a time, a heterogeneous graph-based model was proposed in [10] for the group recommendation, tags for the group recommendation and the event recommendations in a single model. Node proximity of the graphs was calculated using the Random Walk with Restart[RWR]. Also, this was improved for the heterogeneous graphs with the multivariate Markov chains for the node proximities [10].

People who join the groups of the ESBN may have a connection or can be complete strangers. To know the mutual relationship or the influence of one another to schedule an event was studied using the dynamic mutual influence [11]. This could effectively predict the user decisions making process in the social events.

The events that are created are very short lived and there is no proper feedback from them i.e. the cold start. So, the recommendation is difficult, to overcome it a contextual and personalized recommender for the events was proposed in [12]. Where the preference and the influence were gathered from the previous events. The cost, social and location attributes were also gathered to find out the relationships between the hosts and the users.

After the group and the event recommendation, knowing the success of those groups or events is a curiosity. Predicting the success of groups was discussed in [13] using the metrics like average group size, average event attendance, the rate of the growth in event attendance and group size. Later the features like tag related (*‘average intra member tag vector similarities’*), count-based (*‘fraction of group members sending yes or no’*), time related (‘*which day of the week and time’*), location related (*‘average pairwise distance between the group members’*) [13]. Later this was fed into machine learning models such as the SVM, Naïve Bayes for classification and prediction.

III. Methodology

A. Random Forest

Random Forest Classification method is more like a bootstrapping algorithm with decision tree (Classification and Regression Tree/CART) model. Random Forest builds multiple CART models by selecting random variables and choosing random initial variables from the sample of observations which is provided as input. It will repeat the process several times and makes the final prediction for each observation. The prediction achieved for each observation will be the mean of each prediction [14].

The main purpose of us choosing Random Forest is it improves the accuracy of each prediction by encouraging the diversity of the tree.

B. K-NN

For the information retrieval, KNN is the most widely used algorithm by data scientist, as it is the simplest algorithm which would calculate the distance between the query of each observation and the data point of each observation in the train dataset which results in finding the K closest observation. In simple words, by using the KNN it would compare each and every data point mentioned in the query point and find nearest points [20].

C. Support Vector Machine (SVM)

Machine Learning algorithms are like a pack of armory which consists of blades, axes, swords, etc. SVM is like a sharp knife which is more powerful and stronger in building models.

SVM is a supervised machine learning algorithm that is helpful in both classification and regression models [16].

In SVM each observation acts a point in n-dimensional space where n is the number of features we have as input in our data. The value each coordinate will be the value of each feature, which performs the classification by finding which results in differentiating between the two classes.

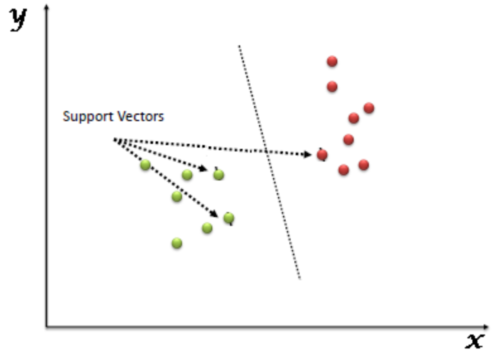
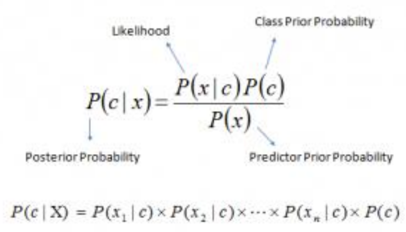


Fig. 2 SVM Classification

D. Naïve Bayes

Naïve Bayes Classification technique is based on Bayes Theorem, the Naïve Bayes classifier assumes that the presence of one particular feature in the observation is unrelated to any other feature present in the observation. For example. A fruit may be considered as orange which is orange in color and round shape with 3 inches in diameter. Even though these specific features are dependent on each other all these features independently contribute to that the fruit is orange hence it’s called naïve.

Naïve Bayes algorithms are easy to build and works like a charm on large datasets. It is also capable to outperform even highly sophisticated classification models [17].



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

IV. Implementation

A. Data extraction and Cleaning:

The data was extracted from meetup API using python code from meetup API documentation. The python code was modified to extract data in form of JSON then the JSON was converted to CSV format. Later, the data was cleaned to remove extra space, new lines, URL, photo URL. The python code was executed on jupyter notebook.

We extracted categories, topics, members_topics, events organized by groups. The implementation of the project was done in R using Rstudio IDE. In Data pre-processing we merged dataset by matching the group_id and memder_id who attend the events by categories of the events. There were 3 categories: "Arts & Culture", "Technology & Career " and "Book Clubs"

We checked for the missing values and omitted the missing values from the dataset. We converted dependent variable that is category_id as a factor and encoded with proper labels. Then we checked for the outliers for independent variables and we checked for class imbalance on a dependent variable, which was quite imbalanced.

K- Nearest Neighbor (K-NN):

We used K-NN classification algorithm, where new data points get classified in a class. We decided the K value based on the number of data points. It is a lazy learner because it doesn't learn much from the training data. The default method Euclidean distance was used. The k value was the square root of the train dataset points because as we experimented the best results were found based on this value and then the data was normalized because the independent variables were in different ranges using the standard normalization function. The library class was used to implement K-NN in R. Later we split the data dataset in 80:20 ratio as train data and test data and the K-NN algorithm is applied to the normalized train data. We have validated the model with test data. It did not predict data points with labels properly. We calculated the miss-classification rate through K-NN model, there was 23.33% miss-classification rate and the prediction rate were 76.7%.

B. Support Vector Machine (SVM):

Support vector machine classification algorithm was applied on Random sample data, because the extracted data set was huge, and we had memory issues while applying SVM. The library e1071 is used to implement SVM in R. We make use of the createDataPartition function to split the dataset as train and test data, this function was built under caret library. SVM was applied to the training dataset, SVM-type is c-classification, kernel type was radial, a total number of support vector machines were 4164 and number of classes were 3. The model was applied even applied to the test data. The misclassification rate was 30% in this model and prediction rate was 70%.

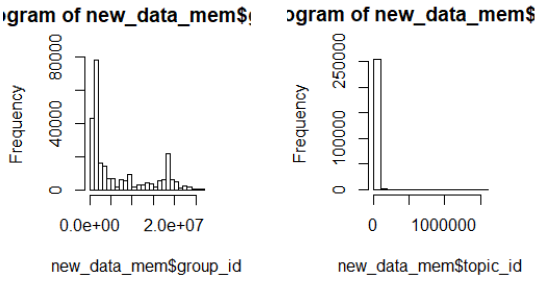
Then we tune the SVM model using tune method with different cost and epsilon ranges, the tune model also suggested that radial kernel and c-classification was the best model compared to other kernels.

C. Random forest:

Random forest classification algorithm is applied to the dataset using library randomForest in R. The dataset was split into 75:25 ratio as train and test data using the sample. 'split', which was built in function in catools package. The library, randomForest was applied on the train dataset where Type of random forest: classification, Number of trees: 500, No. of variables tried at each split: 1 and OOB estimate of error rate: 13.05 %. The confusion matrix will have explained in the evaluation. The accuracy of the model was 87% on the test data, later we applied tuneRF method to tune the model on the training data, then applied randomForest on the tuned model with ntree=300, mtry =2. The model outperformed compared to other algorithms with the accuracy of 98%

D. Naïve Bayes:

The Naïve Bayes uses only the categorical variables so that it can create a matrix having all the combination of the class and their features. As we have the numeric data here, we must convert it into categories. The easiest way to do it is to cut it into data points and the method is called discretization or binning. The columns group_id and topic_id are numeric that is cut into data points. They can be seen in the figure (1). The dataset was split into 80:20 ratio and the model were trained, applied on the test data.

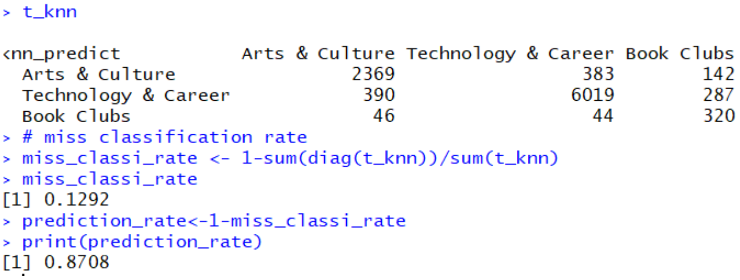


V. EVALUATION:

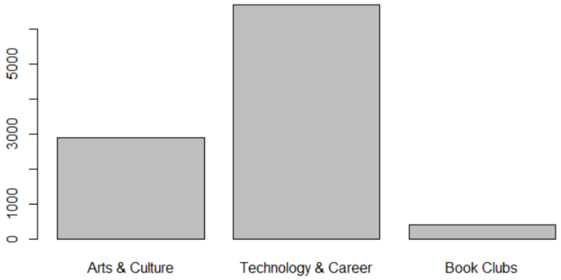
After implementation, the models were tested and evaluated the performance with proper metrics. In some cases, we tuned the models to achieve better results.

A. *K-Nearest Neighbor (K-NN)*:

The obtained results from the model are shown in figure 1 and 2. The first table represents correct and incorrect predictions. The diagonal elements represent the number of correct values and non-diagonal elements represents the incorrect prediction values. Using the below formula, we calculated misclassification rate (12%)and prediction rate (87%). Figure 2 represents the prediction on the test data for each category of the events may occur in future.



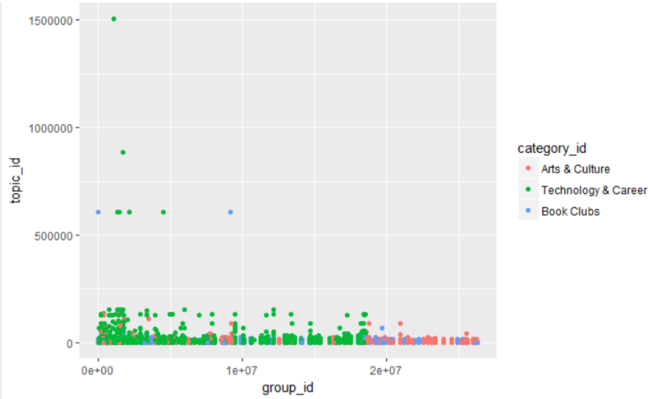
```
> t_knn

‹nn_predict          Arts & Culture Technology & Career Book Clubs
  Arts & Culture               2369                 383        142
  Technology & Career           390                6019        287
  Book Clubs                     46                  44        320
> # miss classification rate
> miss_classi_rate <- 1-sum(diag(t_knn))/sum(t_knn)
> miss_classi_rate
[1] 0.1292
> prediction_rate<-1-miss_classi_rate
> print(prediction_rate)
[1] 0.8708
```

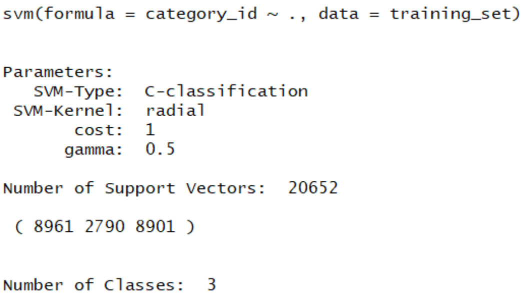


B. *Support Vector Machine (SVM)*:

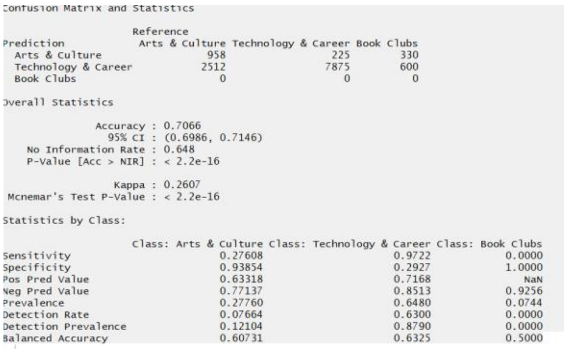
Figure 3 shows the quick plot on category_id distribution for members interest that is topic. The formulated SVM model produced following results shown in figure 4 and 5. There was 20652 number of support vectors. The confusion matrix shows the number of correct and incorrect prediction. The model accuracy is 70%. The sensitivity and specificity are also mentioned in the table. We tune the formulated model with tune function on test data with specific cost, but there were no changes in the results.

This process took considerable time to tune the model and it is also called as hyper-parameter optimization, it helps to use the best model, if the cost of the tuning model is high, the model may tend to over-fitting. Figure 6 shows the performance evaluation of SVM for the two parameters that is epsilon range and cost. The dark region represents better results in this area means low misclassification rate.

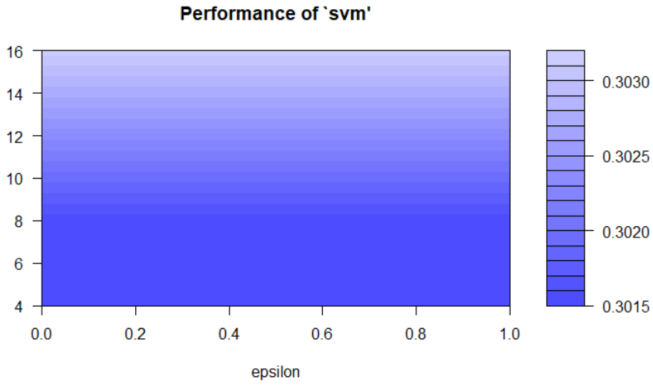




```
svm(formula = category_id ~ ., data = training_set)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.5

Number of Support Vectors:  20652

 ( 8961 2790 8901 )


Number of Classes:  3
```

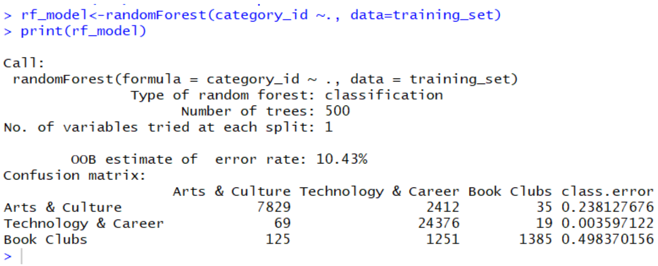


```
Confusion Matrix and Statistics

                     Reference
Prediction            Arts & Culture Technology & Career Book Clubs
  Arts & Culture                 958                 225        330
  Technology & Career           2512                7875        600
  Book Clubs                       0                   0          0

Overall Statistics

               Accuracy : 0.7066
                 95% CI : (0.6986, 0.7146)
    No Information Rate : 0.648
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.2607
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Arts & Culture Class: Technology & Career Class: Book Clubs
Sensitivity                        0.27608                     0.9722            0.0000
Specificity                        0.93854                     0.2927            1.0000
Pos Pred Value                     0.63318                     0.7168               NaN
Neg Pred Value                     0.77137                     0.8513            0.9256
Prevalence                         0.27760                     0.6480            0.0744
Detection Rate                     0.07664                     0.6300            0.0000
Detection Prevalence               0.12104                     0.8790            0.0000
Balanced Accuracy                  0.60731                     0.6325            0.5000
```

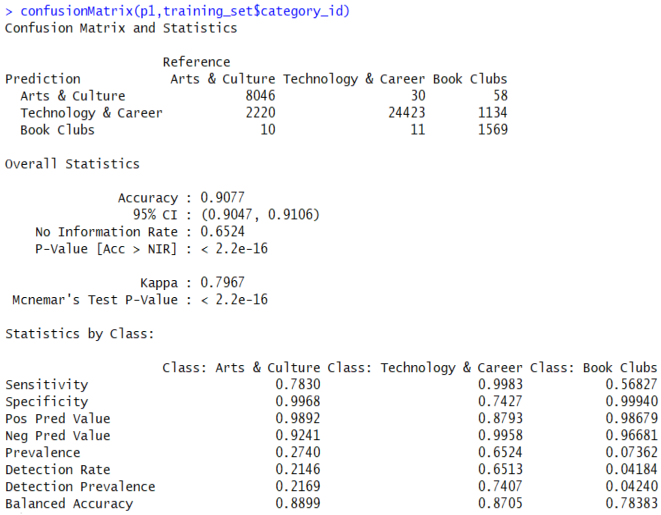


C. *Random Forest:*

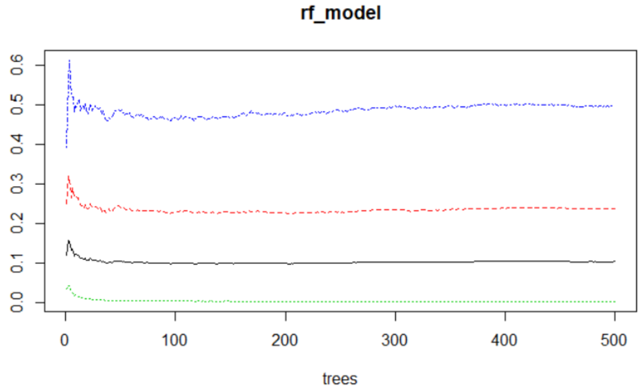
The formulated Random forest model results are shown in following figures 6 and 7. The number of trees for the classification random forest is 500 and one variable tried to split. The out of bag error rate is 10.43%. looking at the confusion matrix, we observe that predicting the Arts & culture error rate is quite less compared to Book clubs. When the model is used to predict on test data, we obtained accuracy of 90.77% and sensitivity and specificity values are shown in the table.



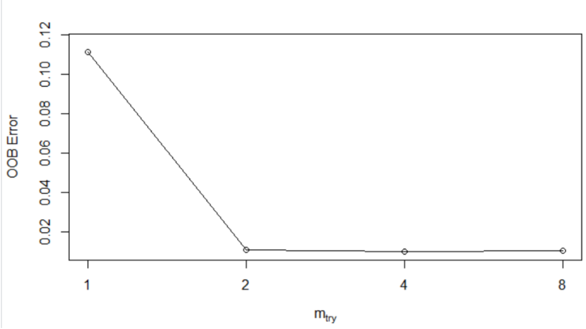
```
> rf_model<-randomForest(category_id ~., data=training_set)
> print(rf_model)

Call:
 randomForest(formula = category_id ~ ., data = training_set) 
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 1

        OOB estimate of  error rate: 10.43%
Confusion matrix:
                     Arts & Culture Technology & Career Book Clubs class.error
Arts & Culture                 7829                2412         35 0.238127676
Technology & Career              69               24376         19 0.003597122
Book Clubs                      125                1251       1385 0.498370156
> |
```

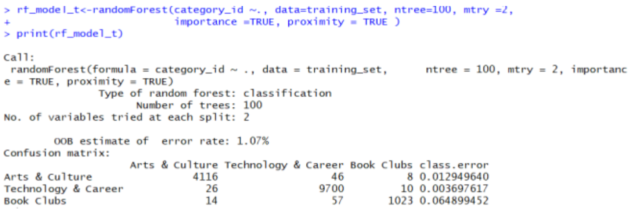


```
> confusionMatrix(p1,training_set$category_id)
Confusion Matrix and Statistics

                     Reference
Prediction            Arts & Culture Technology & Career Book Clubs
  Arts & Culture                8046                  30         58
  Technology & Career           2220               24423       1134
  Book Clubs                      10                  11       1569

Overall Statistics
                                          
               Accuracy : 0.9077          
                 95% CI : (0.9047, 0.9106)
    No Information Rate : 0.6524          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.7967          
 Mcnemar's Test P-Value : < 2.2e-16       

Statistics by Class:

                     Class: Arts & Culture Class: Technology & Career Class: Book Clubs
Sensitivity                         0.7830                     0.9983           0.56827
Specificity                         0.9968                     0.7427           0.99940
Pos Pred Value                      0.9892                     0.8793           0.98679
Neg Pred Value                      0.9241                     0.9958           0.96681
Prevalence                          0.2740                     0.6524           0.07362
Detection Rate                      0.2146                     0.6513           0.04184
Detection Prevalence                0.2169                     0.7407           0.04240
Balanced Accuracy                   0.8899                     0.8705           0.78383
```



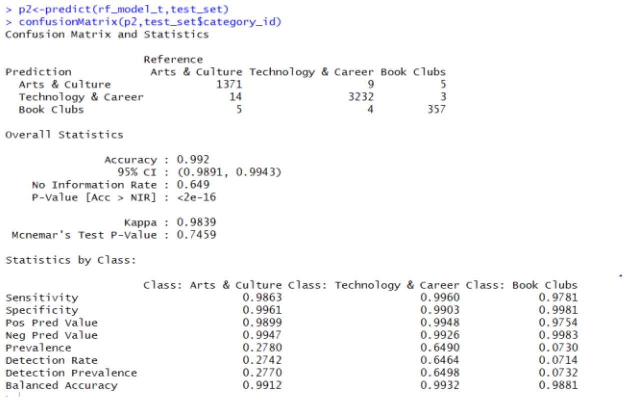
We make use of the tuneRF method to choose the mtry and ntree values in the randomForest model and we plotted OOB error rate at this stage. The figure 8 shows that after mtry 2 there are no changes in the error rate. So, we re-run the model with mtry=2 and ntree =100. The OOB error rate reduces to 1.07%, in the previous formulated model it was 10.43%.





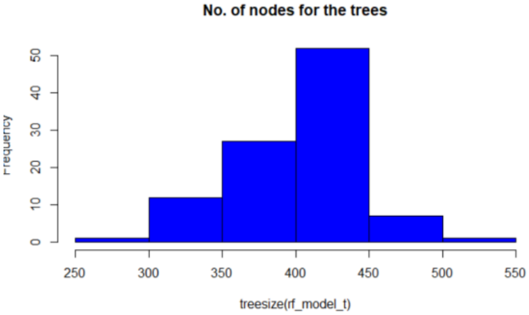
```
> rf_model_t<-randomForest(category_id ~., data=training_set, ntree=100, mtry =2,
+                          importance =TRUE, proximity = TRUE )
> print(rf_model_t)

Call:
 randomForest(formula = category_id ~ ., data = training_set,      ntree = 100, mtry = 2, importanc
e = TRUE, proximity = TRUE) 
               Type of random forest: classification
                     Number of trees: 100
No. of variables tried at each split: 2

        OOB estimate of  error rate: 1.07%
Confusion matrix:
                    Arts & Culture Technology & Career Book Clubs class.error
Arts & Culture                4116                  46          8 0.012949640
Technology & Career             26                9700         10 0.003697617
Book Clubs                      14                  57       1023 0.064899452
```

The below table represents the confusion matrix of the predicted model with correct and incorrect prediction with an accuracy of 99.2%. Compared to all other models the Random forest performed well.

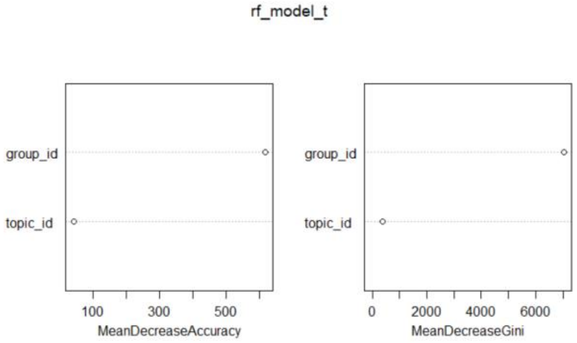


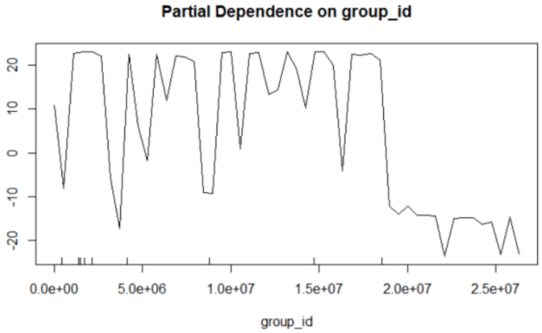
```
> p2<-predict(rf_model_t,test_set)
> confusionMatrix(p2,test_set$category_id)
Confusion Matrix and Statistics

                     Reference
Prediction            Arts & Culture Technology & Career Book Clubs
  Arts & Culture                1371                   9          5
  Technology & Career             14                3232          3
  Book Clubs                       5                   4        357

Overall Statistics
                                          
               Accuracy : 0.992           
                 95% CI : (0.9891, 0.9943)
    No Information Rate : 0.649           
    P-Value [Acc > NIR] : <2e-16          
                                          
                  Kappa : 0.9839          
 Mcnemar's Test P-Value : 0.7459          

Statistics by Class:

                     Class: Arts & Culture Class: Technology & Career Class: Book Clubs
Sensitivity                         0.9863                     0.9960            0.9781
Specificity                         0.9961                     0.9903            0.9981
Pos Pred Value                      0.9899                     0.9948            0.9754
Neg Pred Value                      0.9947                     0.9926            0.9983
Prevalence                          0.2780                     0.6490            0.0730
Detection Rate                      0.2742                     0.6464            0.0714
Detection Prevalence                0.2770                     0.6498            0.0732
Balanced Accuracy                   0.9912                     0.9932            0.9881
```

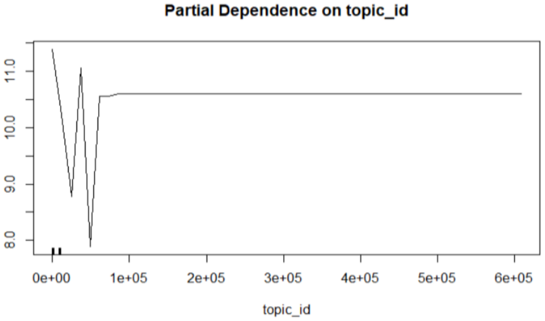
We plotted a histogram for a number of nodes for the tress to the tuned model. This is the distribution of nodes for the 100 trees, as we can see that biggest node is 50 for 400 trees.



These graphs show that how the model will outperform and will fail without each variable. It measures how pure the nodes are at the end of the tree without each variable and have plotted partial plot on each variable for every factor level.

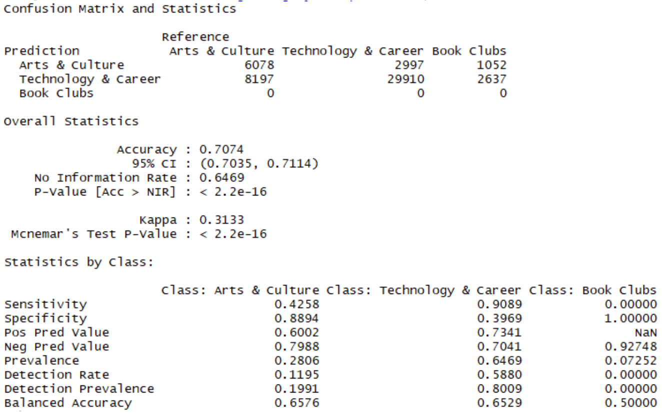






D. Naïve Bayes

The results for the Naïve Bayes is shown in the table (1) with the prediction rate of 70% and misclassification rate of 30%. The Kappa value is 0.31, according to Landin and Koch it's a fair classification as the value is in the range of 0.21-0.40. its poor classification according to Fleiss as the value is < 0.40. The sensitivity and specificity can also be seen in the table. 'Technology & Career' has 90% positive class correctly classified than the others. For specificity, the 'Arts & Culture has 89% negative class that are correctly classified. For 'Book Clubs', 0% of the positive class classified and 100% of the negative class correctly classified.



```
Confusion Matrix and Statistics

                      Reference
Prediction             Arts & Culture Technology & Career Book Clubs
  Arts & Culture                 6078                2997       1052
  Technology & Career            8197               29910       2637
  Book Clubs                        0                   0          0

Overall Statistics

               Accuracy : 0.7074
                 95% CI : (0.7035, 0.7114)
    No Information Rate : 0.6469
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3133
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Arts & Culture Class: Technology & Career Class: Book Clubs
Sensitivity                         0.4258                     0.9089           0.00000
Specificity                         0.8894                     0.3969           1.00000
Pos Pred Value                      0.6002                     0.7341               NaN
Neg Pred Value                      0.7988                     0.7041           0.92748
Prevalence                          0.2806                     0.6469           0.07252
Detection Rate                      0.1195                     0.5880           0.00000
Detection Prevalence                0.1991                     0.8009           0.00000
Balanced Accuracy                   0.6576                     0.6529           0.50000
```

VI. Conclusion

Using the Data Mining techniques, the extracted data were evaluated, and the performance was tested for all the models applied. The key findings of this paper are people have more interest and intend to attend 'Technology and Career' Meetups more often when compared to 'Arts and Culture' and 'Book Clubs'. This shows that members are more interested to build their networks to maximize their opportunities in their career.

VII. Limitations and Future Work

A limitation is data extraction, that it is prohibited for certain groups and sub-groups. The availability of data is uncertain and differs in volume based on locations.

The extension of this paper would be on including more categories for different locations.

VIII. References

[1] G.M. Weiss, and H. Hirsh, 1998, August. “Learning to Predict Rare Events in Event Sequences”. In *KDD* (pp. 359-363).

[2] P. J. Brockwell, and R. Davis, 1996. “Introduction to Time-Series and Forecasting. *Springer-Verlag.*

[3] C. Domeniconi, C. Perng, R. Vilalta, and S. Ma, “A Classification Approach for Prediction of Target Events in Temporal Sequences,” in *Principles of Data Mining and Knowledge Discovery*, 2002, pp. 125–137.

[4] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, “Event-based Social Networks: Linking the Online and Offline Social Worlds,” in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2012, pp. 1032–1040

[5] W. Zhang, J. Wang, and W. Feng, “Combining Latent Factor Model with Location Features for Event-based Group Recommendation,” in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2013, pp. 910–918.

[6] Qiao12, Z., Zhang, P., Zhou, C., Cao, Y., Guo, L. and Zhang, Y., “Event recommendation in event-based social networks”. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence. AAAI*. 2014

[7] D. Garcia Gasulla *et al.*, “Social network data analysis for event detection,” in *ECAI 2014: 21st European Conference on Artificial Intelligence: 18-22 august 2014, Prague, Czech Republic: proceedings*, 2014, pp. 1009–1010.

[8] W. Zhang and J. Wang, “A Collective Bayesian Poisson Factorization Model for Cold-start Local Event Recommendation,” in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2015, pp. 1455–1464.

[9] Y. Liao, W. Lam, L. Bing, and X. Shen, “Joint Modelling of Participant Influence and Latent Topics for Recommendation in Event-based Social Networks,” *ACM Trans. Inf. Syst.*, vol. 36, no. 3, pp. 29:1–29:31, Mar. 2018.

[10] T. A. N. Pham, X. Li, G. Cong, and Z. Zhang, “A general graph-based model for recommendation in event-based social networks,” in *2015 IEEE 31st International Conference on Data Engineering*, 2015, pp. 567–578.

[11] T. Xu, H. Zhong, H. Zhu, H. Xiong, E. Chen, and G. Liu, “Exploring the Impact of Dynamic Mutual Influence on Social Event Participation,” in *Proceedings of the 2015 SIAM International Conference on Data Mining*, 0 vols., Society for Industrial and Applied Mathematics, 2015, pp. 262–270.

[12] M. Wei and D. Wang, “CPERS: Contextual and Personalized Event Recommender System,” in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2016, pp. 421–426.

[13] S. Pramanik, M. Gundapuneni, S. Pathak, and B. Mitra, “Predicting Group Success in Meetup.,” in *ICWSM*, 2016, pp. 663–666.

[14] E. Debreuve, “An introduction to random forests,” p. 30.

[15] 2016 at 9:15am Posted by William Vorhies on July 26 and V. Blog, “CRISP-DM – a Standard Methodology to Ensure a Good Outcome.” [Online]. Available: https://www.datasciencecentral.com/profiles/blogs/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome.

[16] “Understanding Support Vector Machine algorithm from examples (along with code),” *Analytics Vidhya*, 13-Sep-2017.

[17] “6 Easy Steps to Learn Naive Bayes Algorithm (with code in Python),” *Analytics Vidhya*, 11-Sep-2017.

[18] M. Hall and S. Caton, “Am I who I say I am? Unobtrusive self-representation and personality recognition on Facebook,” *PLOS ONE*, vol. 12, no. 9, p. e0184417, Sep. 2017.

[19] “About - Meetup.” [Online]. Available: https://www.meetup.com/about/. [Accessed: 01-May-2018].

[20] “Introductory guide to Information Retrieval using kNN and KDTree.” [Online]. Available: https://www.analyticsvidhya.com/blog/2017/11/information-retrieval-using-kdtree/. [Accessed: 01-May-2018].