**Facebook comments**

The objective is to predict the number of comments received by a post in Facebook. I have converted the no of comments column to a binary column(‘bin’) with <5 posts as 0s and >5 posts as 1s. The thresholding is done under the assumption that posts that receives <5 comments are less popular. Hence, we are trying to predict popular and less popular posts. Independent variables are selected based on the results from the assignment 1.

**Independent variables**: page\_likes, page\_checking, page\_talking, cc1, cc2, cc3, cc4, cc5, c1\_avg, c4\_avg, post\_sharecount, h\_local.

**Dependent variable**: bin (binary variable based on thresholding on no of comments variable) Feature variant 1 dataset is taken, and min-max normalization is done on all the selected

features. The dataset is divided into 70/30 split using train\_test\_split from sklearn. The 70% of the data

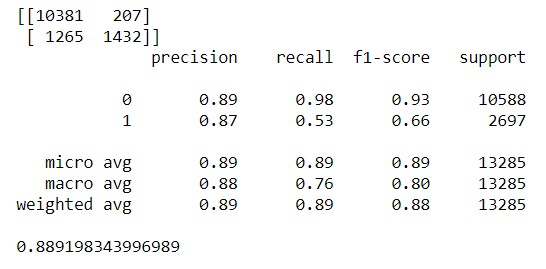
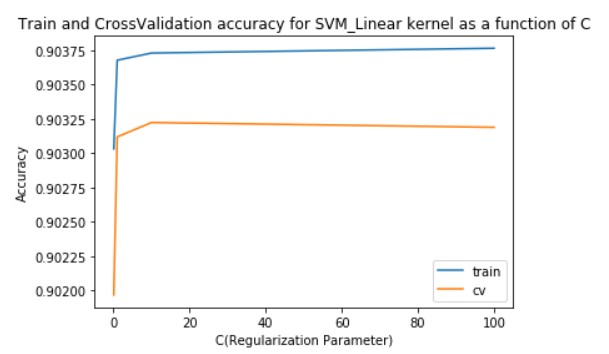
is used for training various models and doing cross validation to select the best hyperparameters for all the models. The remaining 30% will be used as Test set to evaluate the accuracy of the best model taken through cross validation. For all the models, cross validation is dSone for kfold=3 as it will help us generalize the model and will not allow the model to overfit the training data.

**Learning Algorithm 1: Support Vector Machine (SVM):**

Sklearn package in python is used to implement the SVM with various kernels for the dataset.

***Linear Kernel Function:***

The train set is trained on SVM-Linear Kernel function with various values of C and cross- validation is done to calculate the cross-validation and train score to select the optimum C value. C is the regularization parameter for error in the margin that can be tolerated. Lower the value of C, higher the margin will be, it will help the model to tolerate more error and generalize better. Similarly, for higher value of C, margin will be very small, and model tends to overfit the data. Below are the graphs(left) for Train and CV scores for all values of C from 0.1, 1, 10 and 100 respectively and confusion matrix(right) when C=10 on Test set.



From the graph, we are getting maximum test accuracy when C= 10. The test accuracy continuously increases and reaches optimum value at C= 10 and starts to stabilize after C = 10 as the

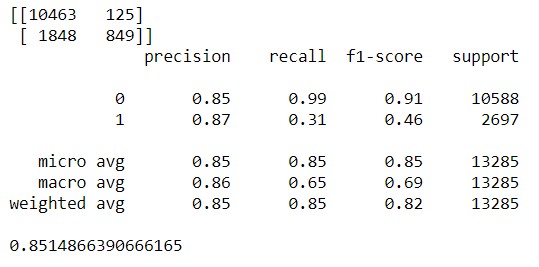
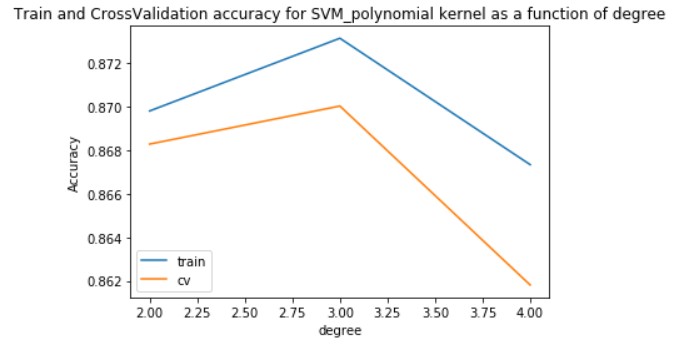
model can not able to generalize any further without increasing bias. When we run the model on the

Test set with C=10, the accuracy = **0.88919**.

***Polynomial Kernel Function:***

The train set is trained on SVM\_polynomial kernel with various degree values from 2 to 4. As the degree of polynomial increases, the model parameters get more flexible and starts to fit the noise in the data. Hence for higher degree of polynomial, model tends to overfit the data which leads to very less train error but very high-test error. Hence, we will plot the learning curve as a function of degree after cross-validation to find the best polynomial degree suited for the model. The below graph shows Train and cross-validation scores(left) for SVM\_polynomial kernel and confusion matrix(right) when degree is

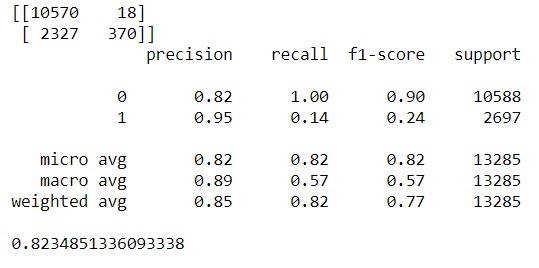
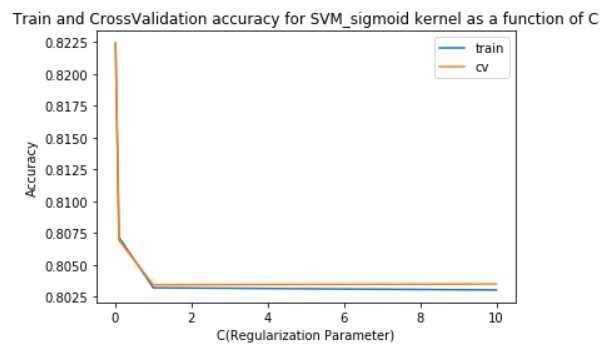
3 on Test set.



From the graph, we can conclude that at degree =3, the model performs the best. As degree goes above 3, model overfits the data and can not able to generalize well on the unseen data. It results in very less test accuracy. The accuracy on the Test set when degree = 3 is **0.8514**.

***Sigmoid Kernel Function:***

Like linear kernel, we will vary the C value for the sigmoid kernel to find the optimum C value at which model does not underfit or overfit the data. Below are the graphs(left) for Train and CV scores for all values of C from 0.01,0.1, 1 and 10 respectively and confusion matrix(right) when C=0.01 on Test set.



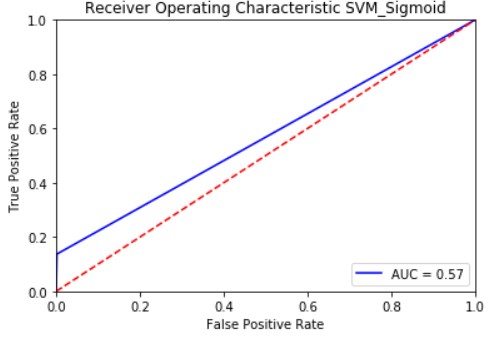
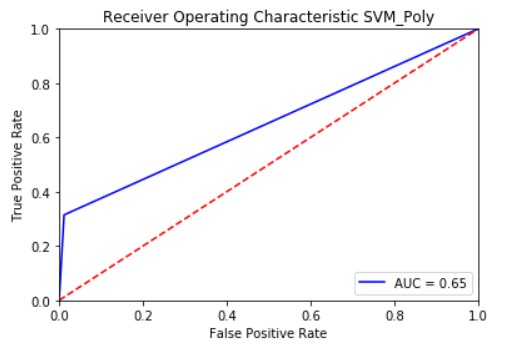
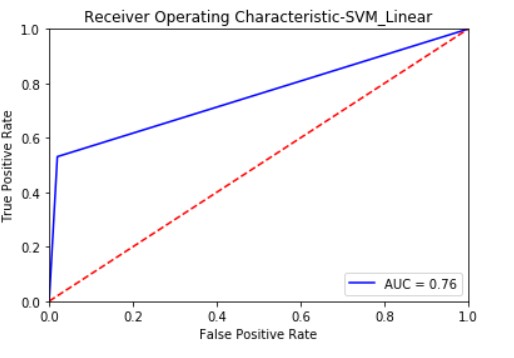
From the graph, we can confirm that the model performs the best when C =0.01. As the value of the value of C increases, Margin gets reduced and model overfits the training data. Hence, we see a drastic reduction in the Cross-validation accuracy. The accuracy on the Test set when C =0.01 is **0.82348**.

***Comparison between all the kernels in SVM:***

From the above three graphs, the best cross validation accuracy for Linear, polynomial and

Sigmoid are **0.903, 0.875 and 0.8225** respectively. from the above three confusion matrix on test set,

the test accuracy for Linear, polynomial and Sigmoid are **0.89, 0.85 and 0.82** respectively. Hence, we can clearly state that for the Facebook dataset, SVM\_linear kernel function performs the best. When we plot the ROC (Receiver Operating Characte) curve and calculate the AUC (Area Under Curve) for all the three kernel functions, we get the below graphs.

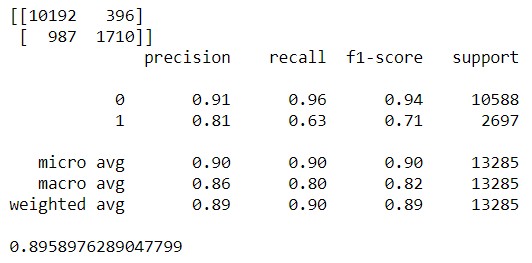
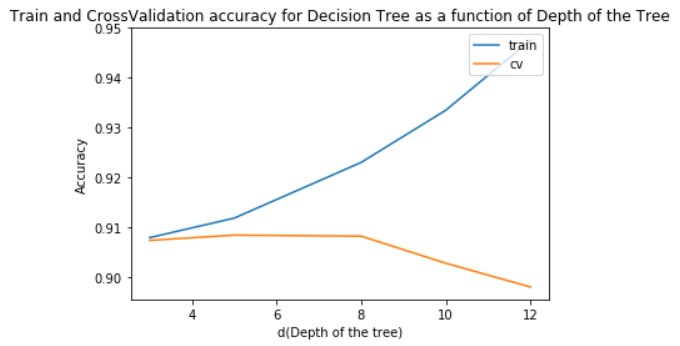


From the above ROC graphs, we can clearly see that AUC for Linear is very big at 0.76 compare to other kernel functions which have AUC=0.65 and AUC= 0.57 respectively. With AUC at 0.57, sigmoid kernel function performs poorly on the provided dataset. Hence, we can state that Linear SVM function is better at distinguishing between popular and less popular posts. With Linear model, we have 76% chance of correctly classifying popular and less popular posts.

In SVM kernel functions, we can conclude that, The Facebook dataset can be better classified by a linear hyperplane than a curved (polynomial) hyperplane or a hyperplane based on sigmoid. This is mainly because of the spread of data on the feature variables.

**Learning Algorithm 2: Decision Tree:**

Sklearn package in python is used to implement Decision Tree with various depths (Pruning) for the dataset. We are using Entropy to calculate the best attribute required for the root nodes and successive nodes. Both Gini Index and Entropy performs like each other with Entropy being computationally expensive. We are using Entropy mainly because the values of lower probability will get scaled up. This will help us to deal with datasets which has high imbalances. When we run the decision tree without any depth criteria, the cv and train score are 0.88 and 0.99. From train score, we can clearly say that the model has fully overfit the data. The value of Train score is close to 1 because it has numerous continuous attributes. Hence, we do pruning with various depths to find the optimum depth at which the model generalizes the data well.

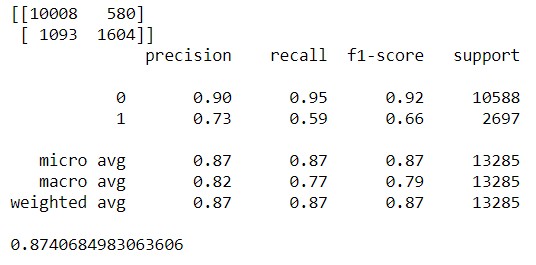
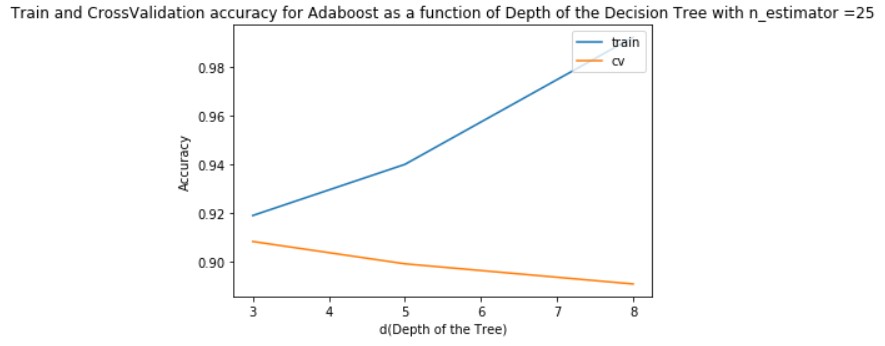


The above graph(left) gives the Cross-validation (Cv) and train scores as a function of depth of the tree and confusion matrix(right) for test set when depth =5. From the graph, we can conclude that when depth=5, we get the maximum Cv score. When we increase the depth more than 5, the model overfits the data. That can be clearly seen as Cv score goes down and train score goes up. The accuracy on the test set for decision tree at depth 5 = **0.895**.

**Learning Algorithm 3: Adaboost with Decision Tree:**

We perform the same decision tree algorithm in Adaboost method with n\_estimators as 25 and

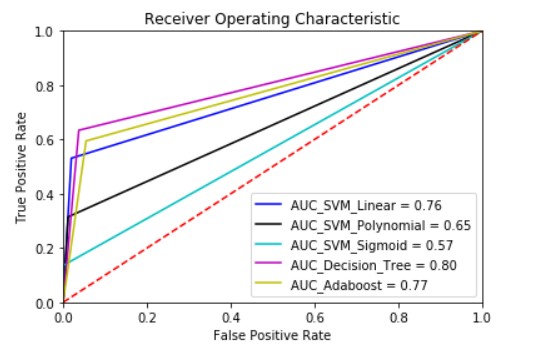
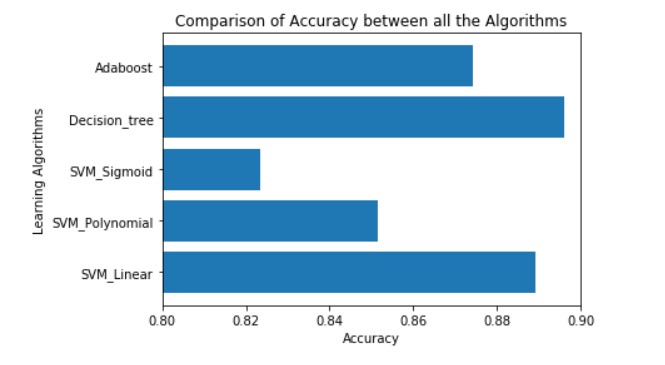
50 for various depths of the tree. For the both the n\_estimators, model is performing similarly. We will try to find the optimum depth at which the model does not overfit or underfit the data. The below graph(left) gives the Cross-validation (Cv) and train scores as a function of depth of the tree and confusion matrix(right) for test set when depth =3.



From the graph, the model performs the best when depth =3. As depth increases, model overfits the data and Cv score goes down. The accuracy for the Test set is **0.87406**.

***Comparison between all the Learning Algorithms*:**

By using the best hyperparameters from all functions, we have predicted the test set and computed the accuracy on all the functions. The below bar chart (Left) shows the Accuracy of all the functions and the graph (Right) ROC curve with AUC for all the functions based on the predicted Test set values.



In the bar chart, we can clearly see that the decision tree algorithm performs the best on Facebook dataset and it is closely followed by SVM\_linear and Adaboost algorithms respectively in terms of Accuracy. The ROC curve also provides us with the same picture. AUC for Decision tree is **0.8**

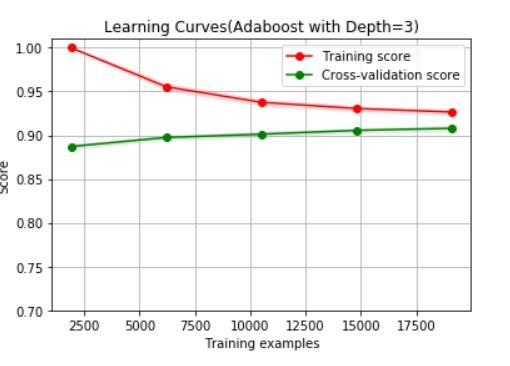
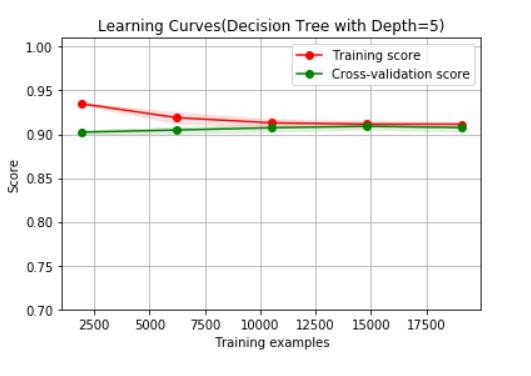
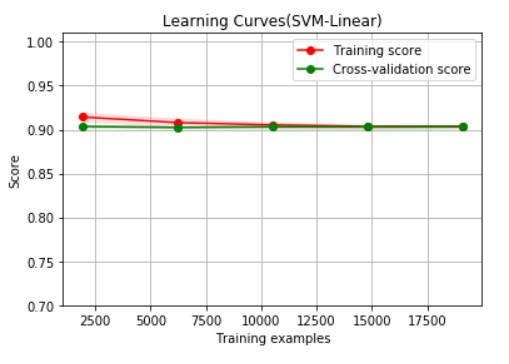
which is higher than all the other algorithm’s AUC. SVM\_Sigmoid with AUC of 0.57 is the poor classifier for this dataset as the value is so close to 0.5.

The main reason decision tree could be performing best on the dataset is due to high number of Continuous variables in the dataset. The reason Adaboost not performing well on the dataset is due to overfitting the data and can not able to generalize well on the data. SVM\_linear is performing well on data but could not match the performance of the Decision Tree because there are features that are

good classifiers of the data enabling decision tree to perform better.

**Learning Curve as function of increasing size of Train and Test Set:**

For each learning algorithm, learning curve is plotted based on the increasing size of the Train and test set sizes. The below graph shows the results for all the algorithms.



From the above graph, we can clearly see that, as the train and test size increases, the Cross- validation score increases, and training score decreases at the same time and after a point it turns to stabilize. We can conclude from above graphs that dataset with size of minimum 15000 data points would suffice for any algorithm.

**Conclusion:**

The final verdict is that Decision tree with depth as 5 is the better model of all the three learning algorithms and Sigmoid kernel of SVM performs poor of all the models.