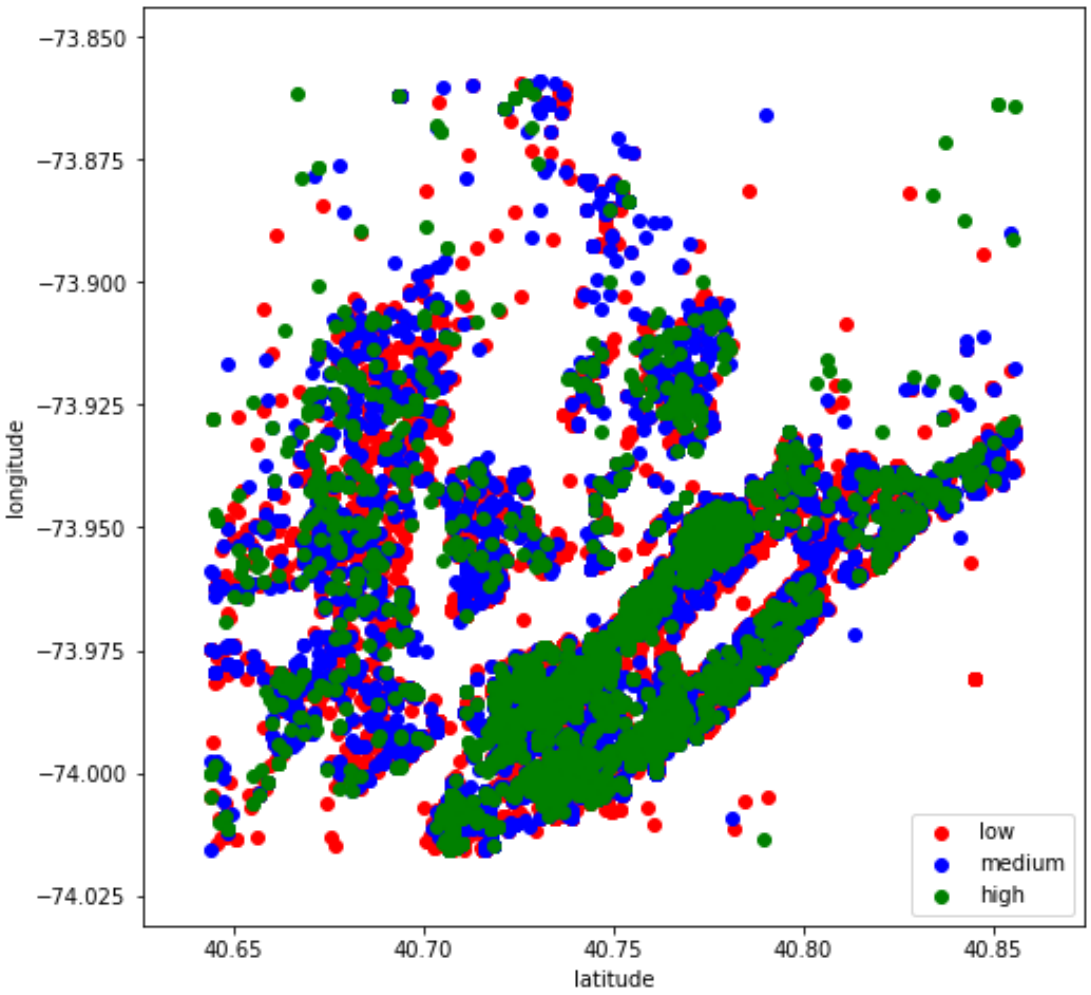
Github: https://github.com/DekaiLin/cmpt459project/tree/master/Milestone2
Decision tree: Dekai Lin, Logistic regression: Zherui Shao, SVM: Luowen Zhu

1. We first use ['bathrooms','bedrooms','latitude','longitude','price'] as our features because they are already there and we do not need to do calculation or transformation to get them. In Question 5, we add ['year','month','day','hour','minute'] and TF vector of the ['features'] as features for improvement. The ['year','month','day','hour','minute'] is derived from the ['created'] attribute.
2. We use the Python library Scikit-learn.
3. We use "scores = cross_val_score(treeModel,X, y, cv=10,scoring = 'neg_log_loss')" to perfom 10-fold cross-validation. After that, we get 10 scores and average the scores to get the estimated performance about the classifier.
4. First version (['bathrooms','bedrooms','latitude','longitude','price']) performance.
   Decision tree logloss: 0.69488 (validation on training dataset), 0.71082 (test dataset).
   Logistic regression logloss: 0.71036 (validation), 0.73383 (test).
   SVM logloss (only 100 iteration): 0.77193(validation), 0.79225 (test).
   Performance: Decision tree > Logistic regression > SVM

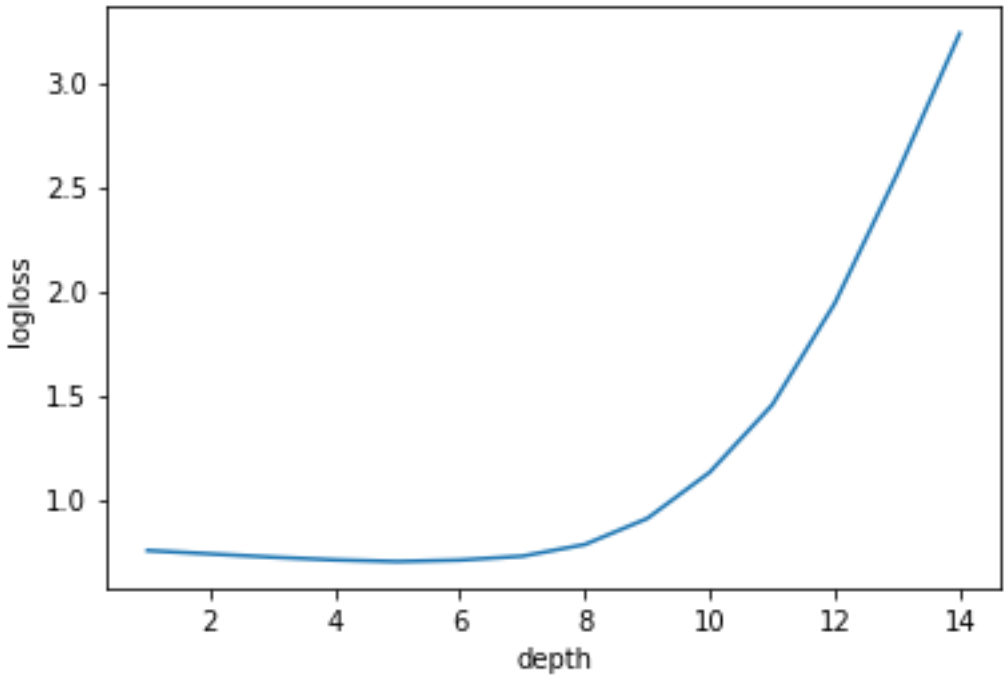
   The reason why Decision tree is outperform than Logistic regression is because Logistic regression assumes the data is linearly or curvy linearly separable in space. The plot below is only using ['latitude','longitude'] as features and this is clear to show that the data we use to train is not linearly separable.



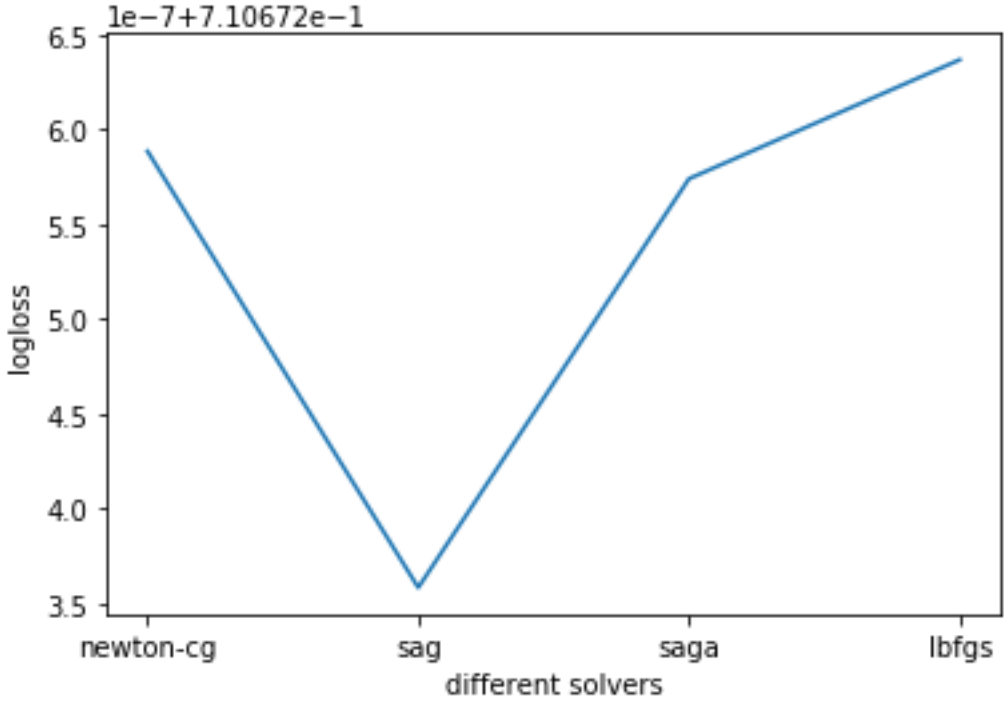
For the SVM, because it is so slow that we only set the max_iter = 100. Apparently, 100 iterations is not enough for the solver to get the converge classification. So the SVM performance is even worse than Logistic regression.

5. a. Tune Parameters:

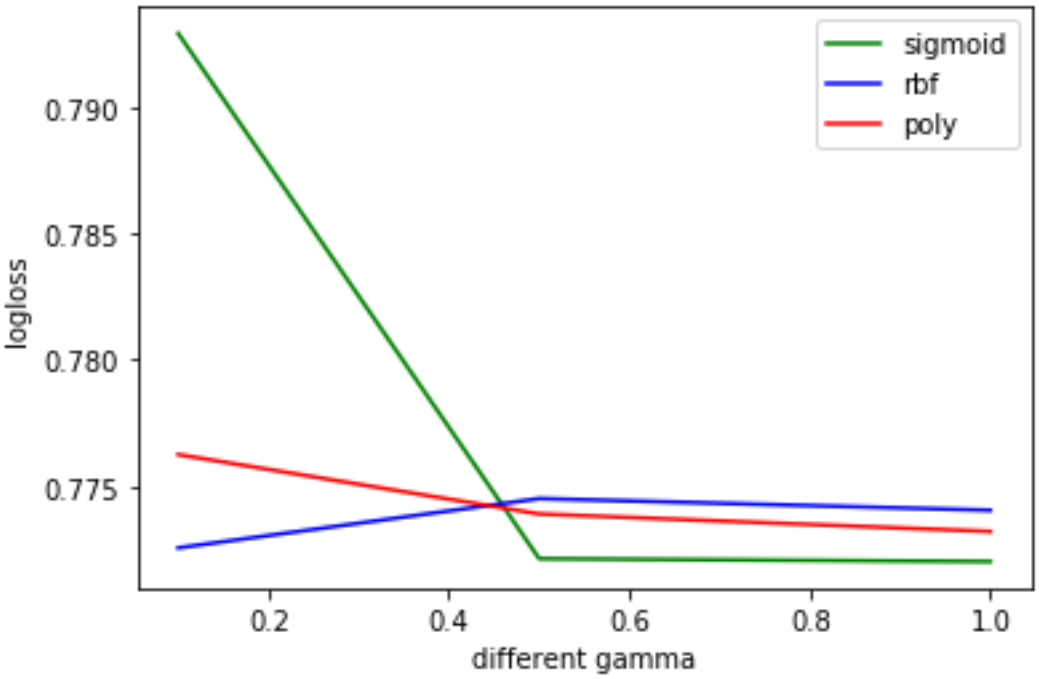
For the Decision tree classifier, we try with different depth of the tree and we find currently depth = 5 achieves the highest performance.



For the Logistic regression, we try with different solver and we find currently 'sag' solver achieves the highest performance.

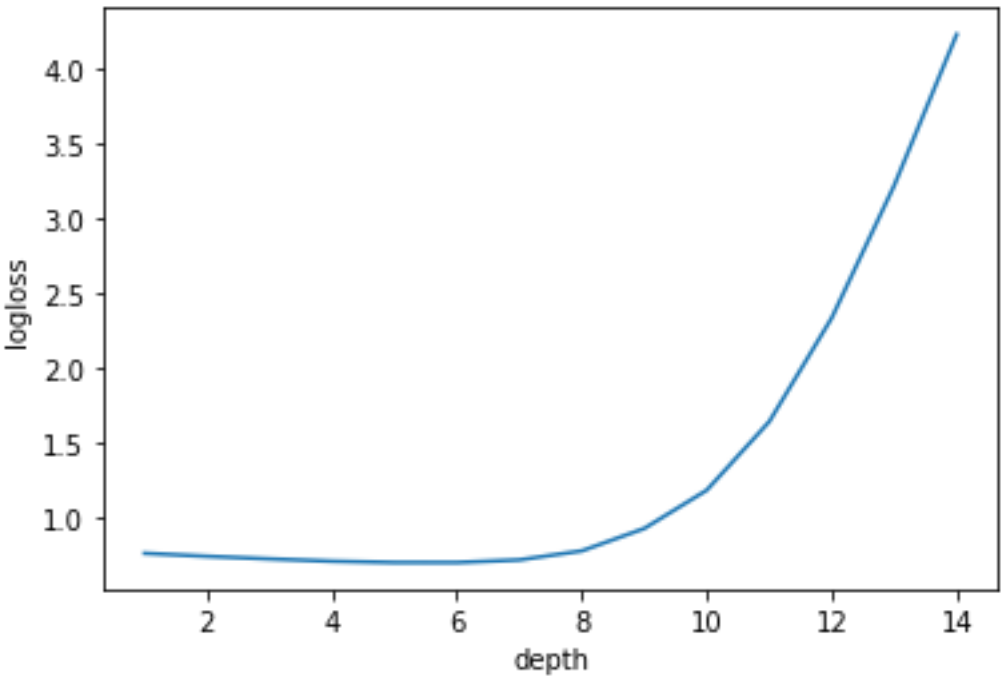


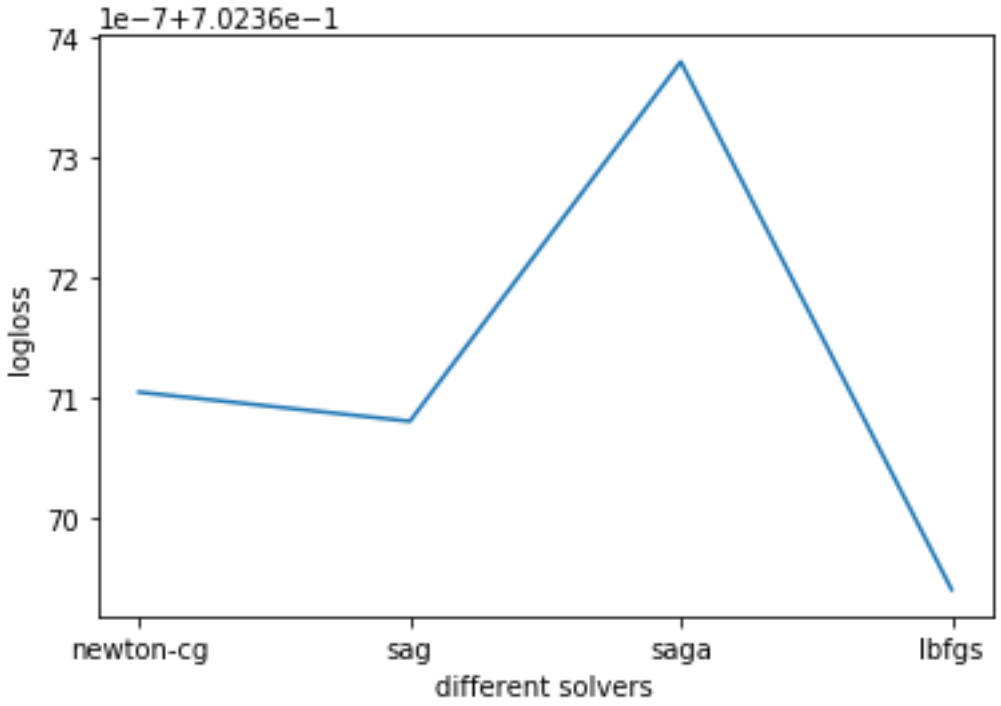
For the SVM, we try with different kernel and gamma and we find currently 'sigmoid' kernel and gamma = 1 achieves highest performance.

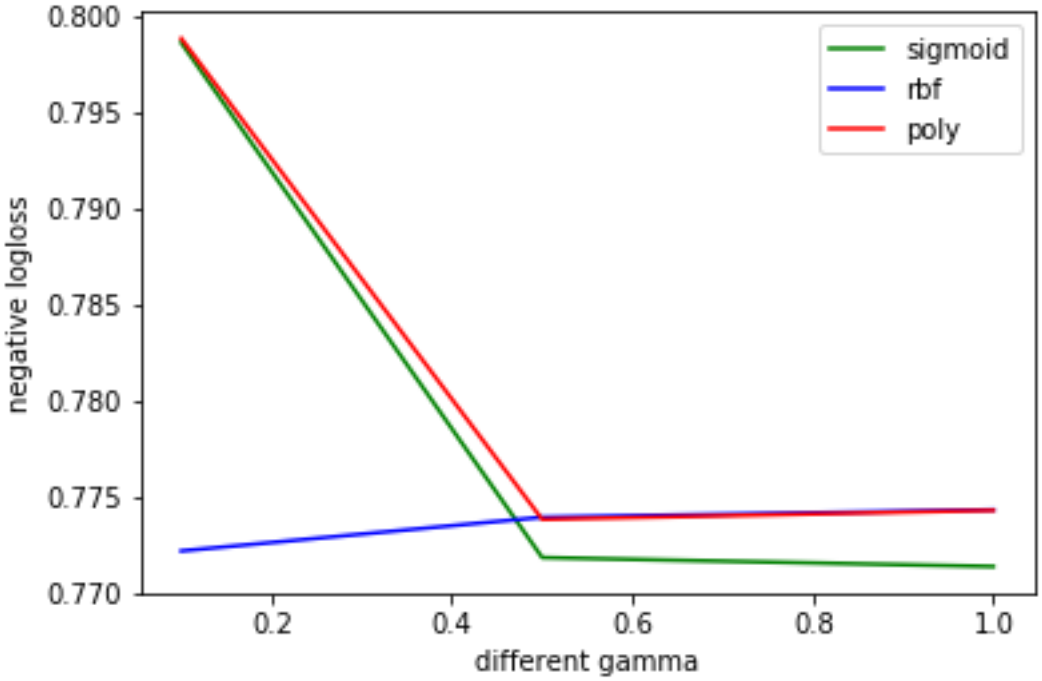


b. Use more features:

we add ['year','month','day','hour','minute'] into our training dataset. We derived these attributes from [‘created’] attribute.

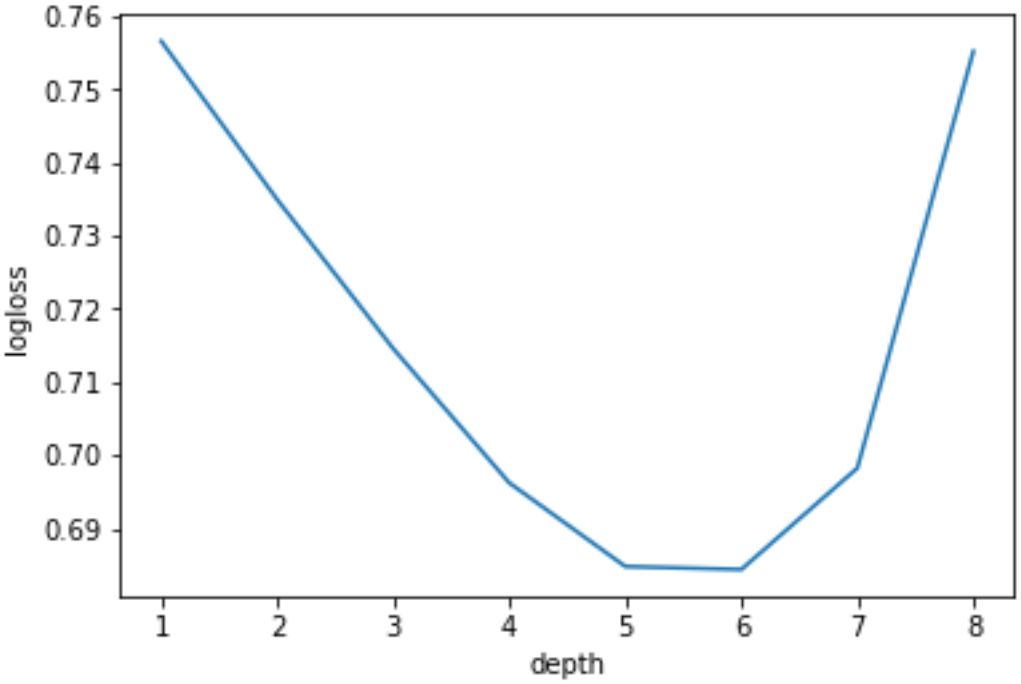


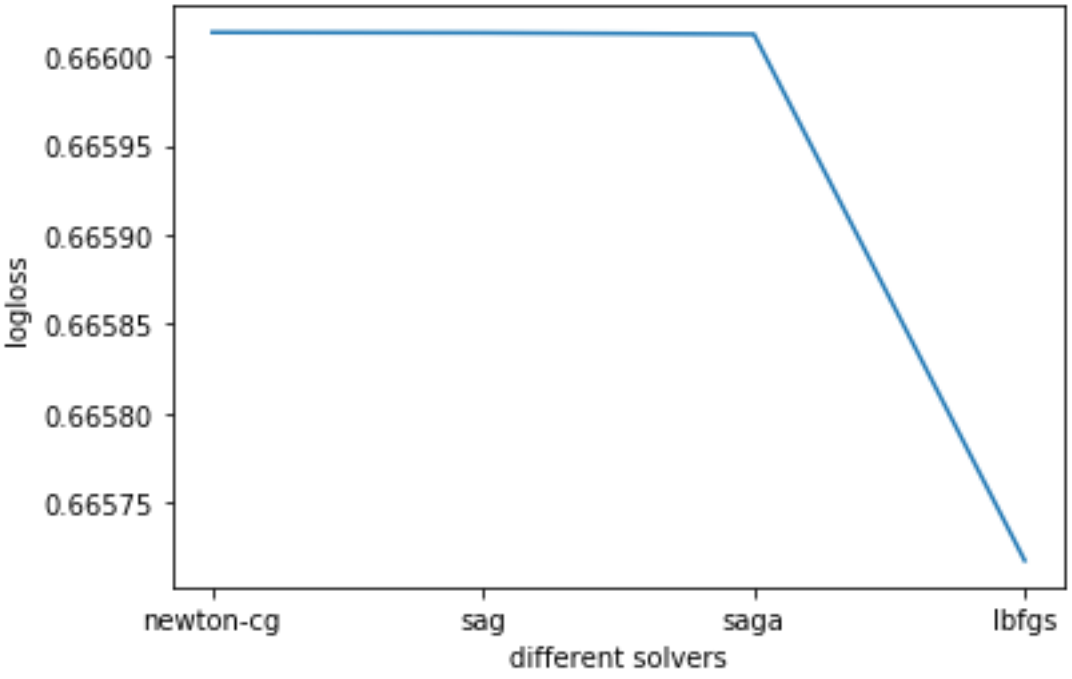


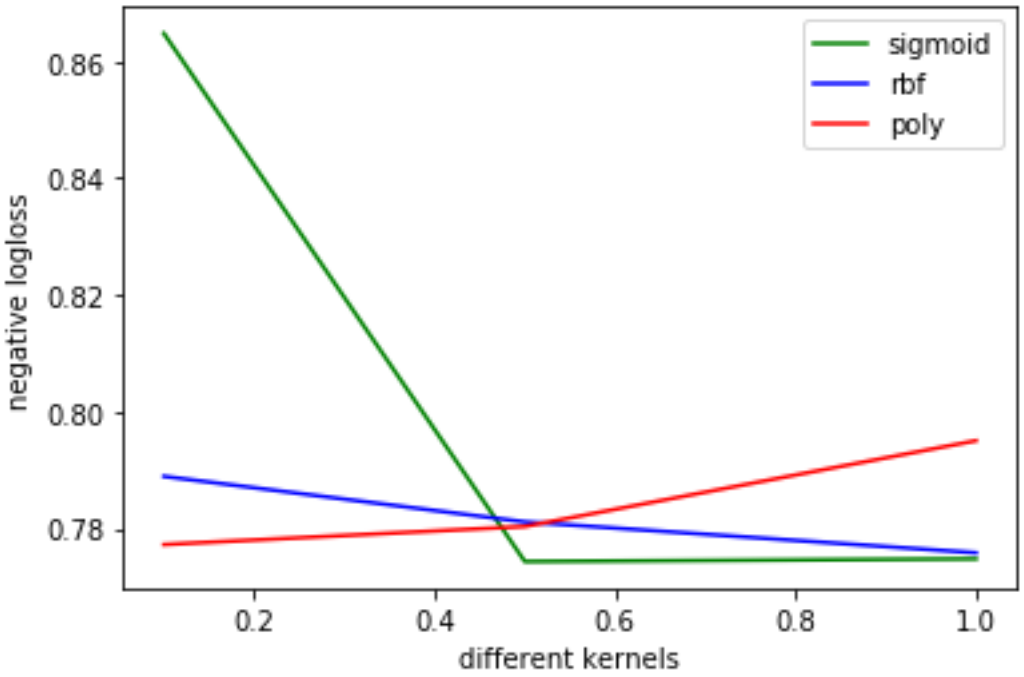


c. Add the unstructured attribute:

At this part, we try to use ['features'] as our new features. Firstly, we join the features in the ['features'] into a single string. So we can treat them as a short document, then we use CountVectorizer to create Term Frequency vector for each training object. Lastly, we hstack the new TF vector into the old attributes to create new training objects.


0.76
0.75
0.74
0.73
0.72
0.71
0.70
0.69
logloss
1
2
3
4
5
6
7
8
depth


0.66600
0.66595
0.66590
0.66585
0.66580
0.66575
logloss
newton-cg
sag
saga
lbfgs
different solvers



6. We compare the logloss on the training dataset and the test dataset, if the test dataset logloss is much bigger, it has overfitting. (For example, we have tried decision tree with depth = 5 and depth = 10 at the second improvement. The logloss of depth = 5 is (0.68489 vs 0.70843) and for depth = 10 is (0.59077 vs 1.33995). So the difference of depth = 10 is too big, it is overfitting.). So, before we actually train the model, we try the cross validation on different parameters first to make sure we do not overfitting the model.
7. Decision tree:

   Validation dataset: 0.69488 → 0.69488 (a. tune parameter) → 0.68489 (b. use more features) → 0.65620 (c. add the unstructured data)

   Test dataset: 0.71082 → 0.71082 → 0.70834 → 0.68476

   Logistic regression:

   Validation dataset: 0.71036 → 0.71036 → 0.71106 → 0.68127

   Test dataset: 0.73383 → 0.73383 → 0.72474 → 0.65759

   SVM:

   Validation dataset: 0.77193 → 0.77334 → 0.77129 → 0.73357

   Test dataset: 0.79225 → 0.79256 → 0.79062 → 0.77147

What we did in the modification step are “Algorithm Tuning” and “Feature Engineering”. Because the machine learning algorithms are driven by parameters, so these parameters majorly influence the outcome of learning process. In the tuning part, we tried to find out the optimal parameters by iterate some possible options and choose the ones have the best performance.

In the Feature engineering part, we extract more information from existing data as our new features. These features may have a higher ability to explain the variance in the training data thus giving improved model accuracy. For example, we extract [‘year’, ‘month’, ‘day’ ⋯] from the [‘created’] attribute. Here the [‘created’] date may not have direct correlation with the [‘interest_level’], but if we look at the year, month or day ⋯ it may have a higher correlation. So we unleash the hidden relationship of a data set in this step.

Because we set max_iter of SVM only 100 in the first two modification, it is too less for the classifier converge to get a best model. So the modifications did not improve the SVM performance a lot in the first two modification and there is a lot of improvement in the last modification.

8. We use classification accuracy as our evaluation metric on the decision tree. The accuracy on the Validation dataset in the last modification is 0.74339 with depth = 9. However, the logloss on test dataset increase to 1.02166. We also test the accuracy on depth = 6 which has the lowest logloss score and we got accuracy 0.71557 but we have logloss 0.68476 on the test dataset.

The difference between logloss and accuracy is logloss is used by the model to decide the probability of the class and accuracy is used to identify how well the model can correctly predict the labels.

So, logloss can indicate the certainty of the model in comparison to the correct labels of the classes in test samples.

In a word, accuracy and logloss are measuring 2 different things so we can not directly compare this two value which makes no sense.