Newsletter Churn Prediction

Rohan Chandra

Addressing the Dataset

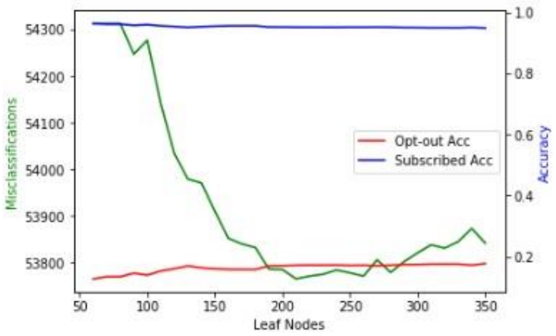
- Feature extraction and data wrangling to determine relevant features(many features not usable)
- Data is unbalanced, the majority of users were remaining subscribed
- Different data transformations proved to be useful for the different models, as well as different features

Profile.Id	Domain	Engagemer	Signup	Opens	Clicks	Last.Open	Last.Click	Optout.Time	Geolocatio	Geolocation.State	Geolocation.Country	Geolc	Lifetime.Message	Top.Device	referral_so	first_name	last_
5572fb559	sbcglobal.n	hardbounc	6/6/2015 9:53	51	11	1/20/2016 0:35	8/12/2015 19:50		El Dorado I	CA US	US	9576	169	Other	DMi	Betty	Goo
5441111b6	hotmail.co	optout	10/17/2014 8:52	2	0	11/21/2014 7:37		11/21/2014 7:45					203		Untagged		
56cde9d9d	yahoo.com	optout	2/24/2016 12:35	0	0			3/2/2016 19:45					10		DMi	Ana	Alva
57e68298c	gmail.com	dormant	9/24/2016 9:41	0	0								257		Facebook		
5498bd77f	sbcglobal.n	optout	12/22/2014 19:55	30	1	1/6/2015 21:10	1/1/2015 21:21	1/6/2015 21:11			US		39	Android			
57f84fbad	hotmail.co	passive	10/7/2016 21:45	54	29	4/17/2017 15:29	4/6/2017 10:34		Birminghar	MI US	US	4800	235	Chrome	In Book Ad		
573f226e7	yahoo.com	optout	5/20/2016 10:42	16	0	5/25/2016 20:24		5/25/2016 20:26					9		LiveIntent		
55f6fb952	gmail.com	disengaged	9/14/2015 12:53	16	0	2/5/2017 20:51							632		DMi	Rose	Dos
542c8af71	hotmsil.cor	hardbounc	10/1/2014 19:15	0	0								1		Untagged		
547240a36	yahoo.com	hardbounc	11/23/2014 15:16	0	0								1		Facebook		
54917c5a1	ymail.com	optout	12/17/2014 7:51	31	5	4/30/2016 9:31	1/1/2016 12:53	4/30/2016 9:32	Evanston, \	WY US	US	8293	438	iPad	Facebook		
5844b929C	aol.com	passive	12/4/2016 19:47	164	20	4/19/2017 12:46	3/15/2017 20:20		New York, I	NY US	US	1002	164	Android Tablet			
5631559ef	yahoo.com	optout	10/28/2015 19:09	20	2	11/24/2015 10:30	10/30/2015 11:08	11/24/2015 10:30	Stockton, C	CA US	US	9520	20	iPhone	In Book Ad		
567665cd9	hotmail.co	disengaged	12/20/2015 3:24	11	0	1/10/2017 11:58							540		DMi	Sandra	Heb
53ebd8281	yahoo.com	dormant	8/13/2014 17:27	0	0								1335		Untagged		
56a9647a1	yahoo.com	optout	1/27/2016 19:44	0	0			2/11/2016 15:18					20		DMi	Lynn	Mh
58b5697bc	gmail.com	active	2/28/2017 7:13	45	23	4/19/2017 13:36	4/17/2017 13:33		Killington, '	VT US	US	0575	66	Chrome	LiveIntent		
568ce7e4a	gmail.com	dormant	1/6/2016 5:09	0	0								258		DMi	Monique	Ives
56a82b95c	gmail.com	disengaged	1/26/2016 21:29	74	0	2/3/2017 9:44							525		DMi	danielle	cha
53f9c552f	gmail.com	optout	8/24/2014 6:58	56	1	5/31/2015 17:46	6/1/2015 8:10	6/1/2015 8:11			US		859	Android	Untagged		
56edcf121	gmail.com	dormant	3/19/2016 18:13	0	0								219		LiveIntent		
56a2e3835	verizon.net	optout	1/22/2016 21:20	29	51	3/19/2016 10:19	6/24/2016 9:55	6/26/2016 15:47	Thousand (	CA US	US	9136	173	iPad	In Book Ad		
5526b41e1	yahoo.com	disengaged	4/9/2015 13:17	1	0	11/17/2015 19:46							622		Permission	Teresa	Saff
58e7ec722	yahoo.com	passive	4/7/2017 15:45	2	0	4/15/2017 22:04							17		torsweeps		
578154dbc	verizon.net	optout	7/9/2016 15:47	51	0	8/13/2016 21:08		8/6/2016 12:54					33		LiveIntent		
585570092	comcast.ne	optout	12/17/2016 12:04	7	0	1/21/2017 11:33		1/21/2017 11:33					42		LiveIntent		
549b4276f	yahoo.com	optout	12/24/2014 17:47	7	0	1/5/2015 9:55		1/5/2015 9:55					29		Facebook		
570c0c124	sbcglobal.n	optout	4/11/2016 16:41	2	0	4/13/2016 11:02		4/13/2016 11:02					2		BookRiotSweeps		
544af5e0f	aol.com	hardbounc	10/24/2014 20:59	0	0								1		Untagged		
585162272	gmail.com	passive	12/14/2016 10:15	55	4	4/17/2017 22:36	3/29/2017 15:09		Decatur, G	GA US	US	3003	155	Android	Google		
58e7ec6f2	verizon.net	dormant	4/7/2017 15:45	0	0								11		torsweeps		
56a021c71	gmail.com	dormant	1/20/2016 19:09	0	0								241		DMi	cedric	coe
56e6d8eb6	gmail.com	disengaged	3/14/2016 11:29	1	0	6/5/2016 9:51							437		DMi	Ryan	Gar
54208f961	comcast.ne	optout	9/23/2014 16:29	622	77	6/12/2016 23:20	11/11/2015 13:30	4/14/2016 10:21	Mechanics	PA US	US	1705	1015	Android Ta	Facebook		
54a6c4b91	yahoo.com	disengaged	1/2/2015 11:18	2	5	1/27/2015 12:18	1/27/2015 12:19		Barberton,	OH US	US	4420	24	iPhone			
55ca11f29	gmail.com	optout	8/11/2015 11:17	4	0	8/17/2015 11:04		8/17/2015 11:05					5		DMi	Bee	Pars
569d18dcb	gmail.com	optout	1/18/2016 11:54	2	0	3/2/2016 19:15		3/2/2016 19:16					51		DMi	Megan	Rus
5862f5a20	aol.com	disengaged	12/27/2016 18:13	2	0	2/23/2017 23:37							142				

Decision Tree

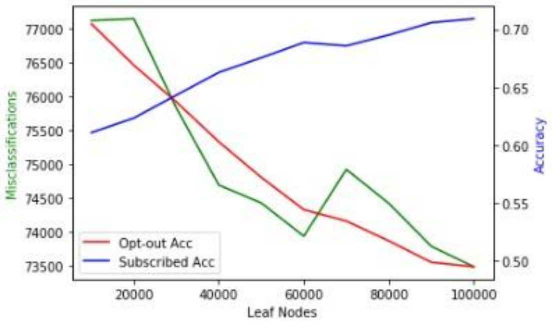
- Used relevant features and got unbalanced results
    - Minimal misclassifications at 210 leaf nodes
    - Great subscriber accuracy with poor opt-out accuracy

```
Test Opt-out Percent Correct:  0.17116292427261143
Test Subscribed Percent Correct:  0.951306222766972
Test Misclassifications:  53765
```



- Class weights of tree were balanced
    - More misclassifications than unbalanced tree
    - Less gap in the accuracy of predicted Opt-out and Subscribed users
    - Misclassifications continue to decrease as max nodes increases

```
Test Opt-out Percent Correct:  0.5440920431497137
Test Subscribed Percent Correct:  0.6885279565683337
Test Misclassifications:  73937
```



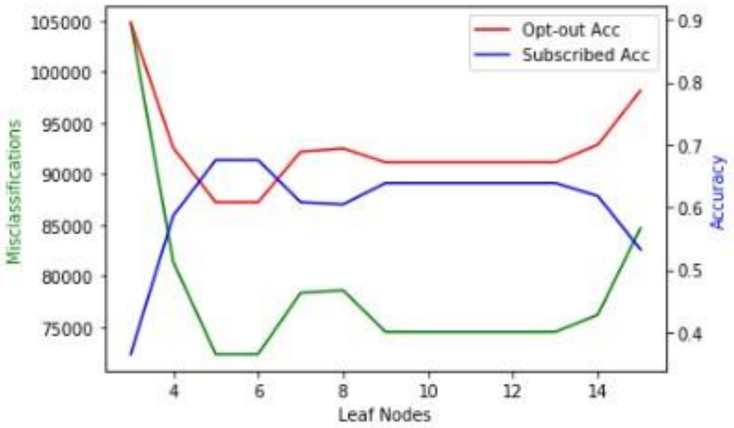
Decision Tree Improvement

Problems:

- Model training time is very high
- Final tree is very complex with no optimal max leaf nodes
- Accuracy of model still suffered

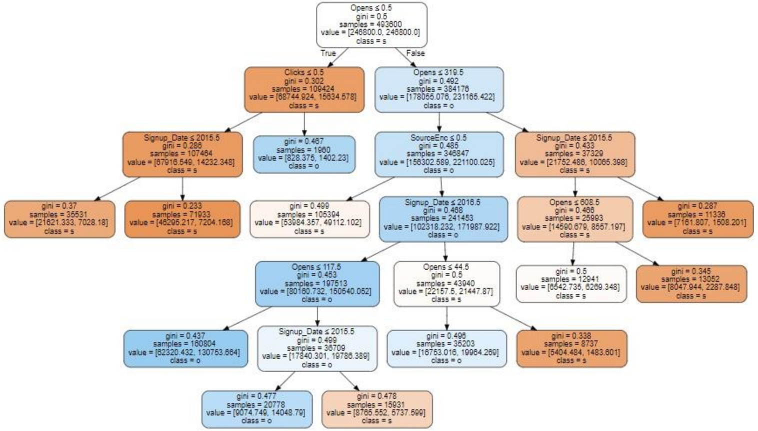
Discovered Solution:

- Initial inclusion of the large amount of one-hot encoded user genre preference features helped lower misclassifications before class weights were balanced
- After balancing the weights these features harmed the accuracy of the model while making the model overly complex.
- Removal of preference features greatly simplified model and improved accuracy
- Max leaf nodes was optimized at 5
- A defined max depth only hindered the accuracy
- Other variables such as minimum samples for a split and minimum samples per leaf had no effect on outcome




```
Training Opt-out Percent Correct:  0.609967796350253
Training Subscribed Percent Correct:  0.6751996255300402
Test Opt-out Percent Correct:  0.6083140380162619
Test Subscribed Percent Correct:  0.6755331095866676
Test Misclassifications:  72384
```

Decision Tree Conclusion



If the goal was to minimize misclassifications:

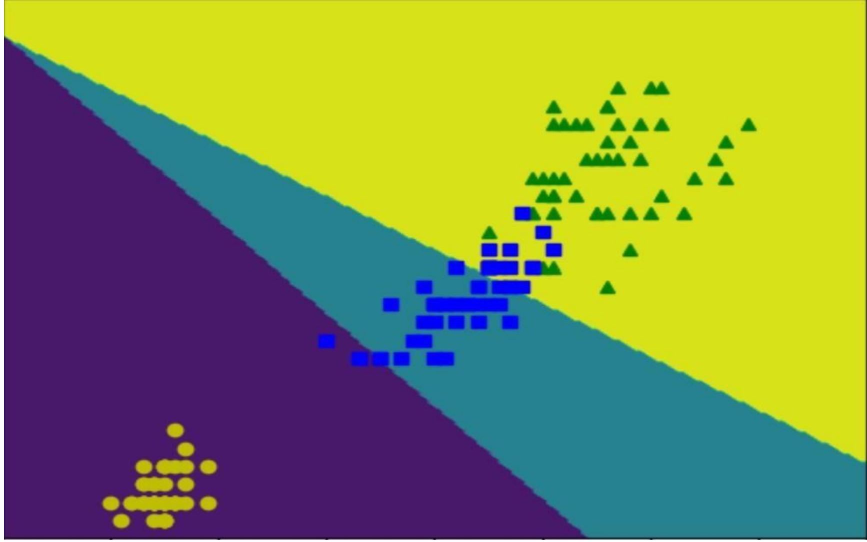
- Balancing the class weights would hurt the results
- Subscriber prediction accuracy excels at the cost of the opt-out accuracy

With a goal of predicting users who will potentially opt-out:

- Balancing class weights puts an emphasis on equal accuracy for each classification
- More useful outcome since the model attempts to accurately predict opt-out users

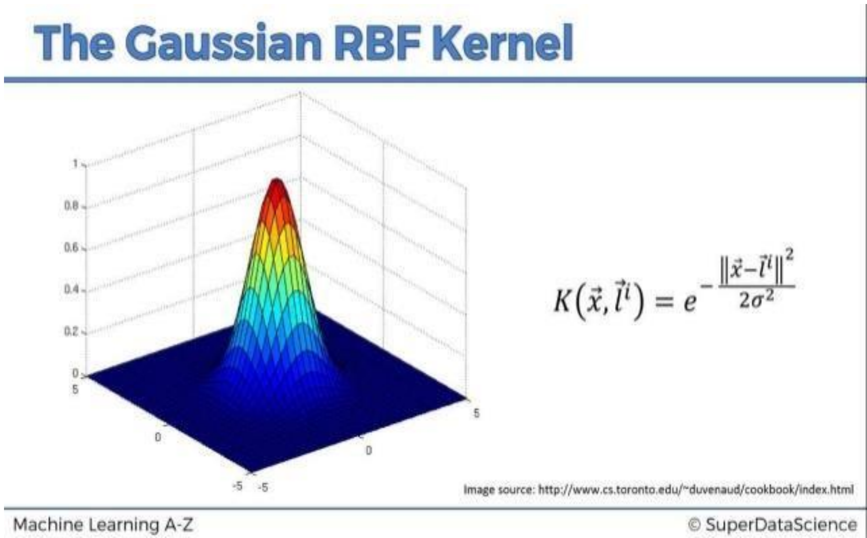
Takeaways:

- An overly complex model will lead to little information gain with each split
- Even after all attempted optimization, a decision tree is most likely the ideal model for the task



SVM

- Standard SVC from scikit-learn too slow, hard to test with multiple parameters due to long training times
- Used LinearSVC, got imbalanced results(very low accuracy for predicting optouts, and very high accuracy for predicting subscribers)
- Used Nystroem kernel approximation paired with a LinearSVC, and RBFSampler kernel approximation paired with SGDClassifier; results were still not good
- The above 2 models behave similar to an SVM with an RBF kernel, but with much faster training times
- Attempted an ensemble of all the above models using a



VotingClassifier ;no improvement

- Attempted to loop through all possible parameters for most of the above models; no improvement

Solution

- Dropped most of the above models; kept only 3 which seemed to show slightly better results more consistently
- The format of the data, it’s imbalanced examples and the features used were the problem
- Applied One-Hot-Encoding and changed other features of the dataset to suit the selected models
- Dropped a few features, included some which were not used earlier
- Tuned the parameters and narrowed down the useful models to 2 with the below parameters:
    - LinearSVC(fit_intercept = False,dual = False, class_weight = 'balanced')
    - RBFSampler(gamma= 0.1, random_state=1); SGDClassifier(max_iter=1000, class_weight = 'balanced')

LinearSVC

```python
# Without SourceEnc and no Parameters Set: Test Opt-Out Percent Correct: 0.607 Test Subscribed Percent Correct: 0.87
#Parameters:  fit_intercept = False: Optout Accuracy : 0.98, subscribed accuracy: 0.038
#             fit_intercept = False,dual =False: Optout Accuracy: 0.773, subscribed accuracy: 0.851
#             dual =False: Optout Accuracy: 0.7724, subsribed accuracy: 0.851
#             C = 3.0, max_iter = 2000: Optout : 0.54, subscribed = 0.92

from sklearn import svm
from sklearn.svm import SVC
clf = svm.LinearSVC(fit_intercept = False,dual = False, class_weight = 'balanced')
clf.fit(X_train,Y_train)
yPredTrain = clf.predict(X_train)
yPredTest = clf.predict(X_test)
trainOptAcc, trainSubAcc, trainMisclass = accuracy(Y_train, yPredTrain)
testOptAcc, testSubAcc, testMisclass = accuracy(Y_test, yPredTest)

print("Training Opt-out Percent Correct: ",trainOptAcc, "\nTraining Subscribed Percent Correct: ", trainSubAcc)
print("Test Opt-out Percent Correct: ",testOptAcc, "\nTest Subscribed Percent Correct: ", testSubAcc, "\nTest Misclassificati
```

```
Training Opt-out Percent Correct:  0.7744594387363901
Training Subscribed Percent Correct:  0.850011013822347
Test Opt-out Percent Correct:  0.7728537325220326
Test Subscribed Percent Correct:  0.850934666401422
Test Misclassifications:  35884
```

RBF-SGD

```python
# Without SourceEnc and no Parameters Set: Test Opt-Out Percent Correct: 0.51Test Subscribed Percent Correct: 0.55
#Parameters:  RBFsampler gamma = 0.1: Optout Accuracy : 0.56, subscribed accuracy: 0.59
#             RBFsampler gamma = 0.3: Optout Accuracy: 0.49, subscribed accuracy: 0.61
#default      RBFsampler gamma = 1.0: Optout Accuracy: 0.51, subscribed accuracy: 0.55
#             RBFsampler gamma = 3.0: Optout Accuracy: 0.48, subscribed accuracy: 0.58
#             SGDClassifier alpha =0.001,max_iter = 2000: optout accuracy: 0.46,subscriber accuracy: 0.60
#             Best: gamma = 0.1(RBF), max_iter =1000 and class_weight = balanced(SGD) (Without SourceEnc column)
from sklearn import svm
from sklearn.svm import SVC
from sklearn.kernel_approximation import RBFSampler
from sklearn.linear_model import SGDClassifier
rbf_feature = RBFSampler(gamma= 0.1, random_state=1)
X_features = rbf_feature.fit_transform(X_train)
X_fit = rbf_feature.fit_transform(X_test)
sgd = SGDClassifier(max_iter=1000, class_weight = 'balanced')
    #clf = svm.LinearSVC()
    #sgd.fit(X_train, Y_train) #16% optout
sgd.fit(X_features, Y_train)
print(sgd.score(X_fit, Y_test))
print(sgd.score(X_features, Y_train))
yPredTrain = sgd.predict(X_features)
yPredTest = sgd.predict(X_fit)
trainOptAcc, trainSubAcc, trainMisclass = accuracy(Y_train, yPredTrain)
testOptAcc, testSubAcc, testMisclass = accuracy(Y_test, yPredTest)
print("Training Opt-out Percent Correct: ",trainOptAcc, "\nTraining Subscribed Percent Correct: ", trainSubAcc,"\t")
print("Test Opt-out Percent Correct: ",testOptAcc, "\nTest Subscribed Percent Correct: ", testSubAcc, "\nTest Misclassificat
```

```
0.5978850735544379
0.5992706645056726
Training Opt-out Percent Correct:  0.5311225272197516
Training Subscribed Percent Correct:  0.6237430475246434
Test Opt-out Percent Correct:  0.527794231148924
Test Subscribed Percent Correct:  0.6229440868633327
Test Misclassifications:  85065
```

- LinearSVC provided the highest accuracy, RBFSampler with SGDClassifier was good enough to be included as well

LinearSVC Test Accuracies

No Parameters Set: Optout Accuracy: 60.7%, Subscribed Accuracy: 87%

Parameters: fit_intercept = False: Optout Accuracy : 42.36%, Subscribed Accuracy: 95.68%

fit_intercept = False,dual =False: Optout Accuracy: 77.3%, Subscribed Accuracy: 85.1%

dual =False: Optout Accuracy: 77.24%, Subscribed Accuracy: 85.1%

C = 3.0, max_iter = 2000: Optout Accuracy: 54%, Subscribed Accuracy: 92%

Best: fit_intercept = False,dual =False

Best Accuracy: Optout: 77.3% Subscribed: 85.1%

RBF+SGD Test Accuracies

No Parameters Set: Optout Accuracy: 51%, Subscribed Accuracy: 55%

Parameters: RBFsampler gamma = 0.1: Optout Accuracy : 53%, Subscribed Accuracy: 62%

RBFsampler gamma = 0.3: Optout Accuracy: 49%, Subscribed Accuracy: 61%

default RBFsampler gamma = 1.0: Optout Accuracy: 51%, Subscribed Accuracy: 55%

RBFsampler gamma = 3.0: Optout Accuracy: 48%, Subscribed Accuracy: 58%

SGDClassifier alpha =0.001,max_iter = 2000: Optout Accuracy: 46%, Subscribed Accuracy: 60%

Best: gamma = 0.1(RBF), max_iter =1000 and class_weight = balanced(SGD)

Best Accuracy: Optout: 53% Subscribed: 62%

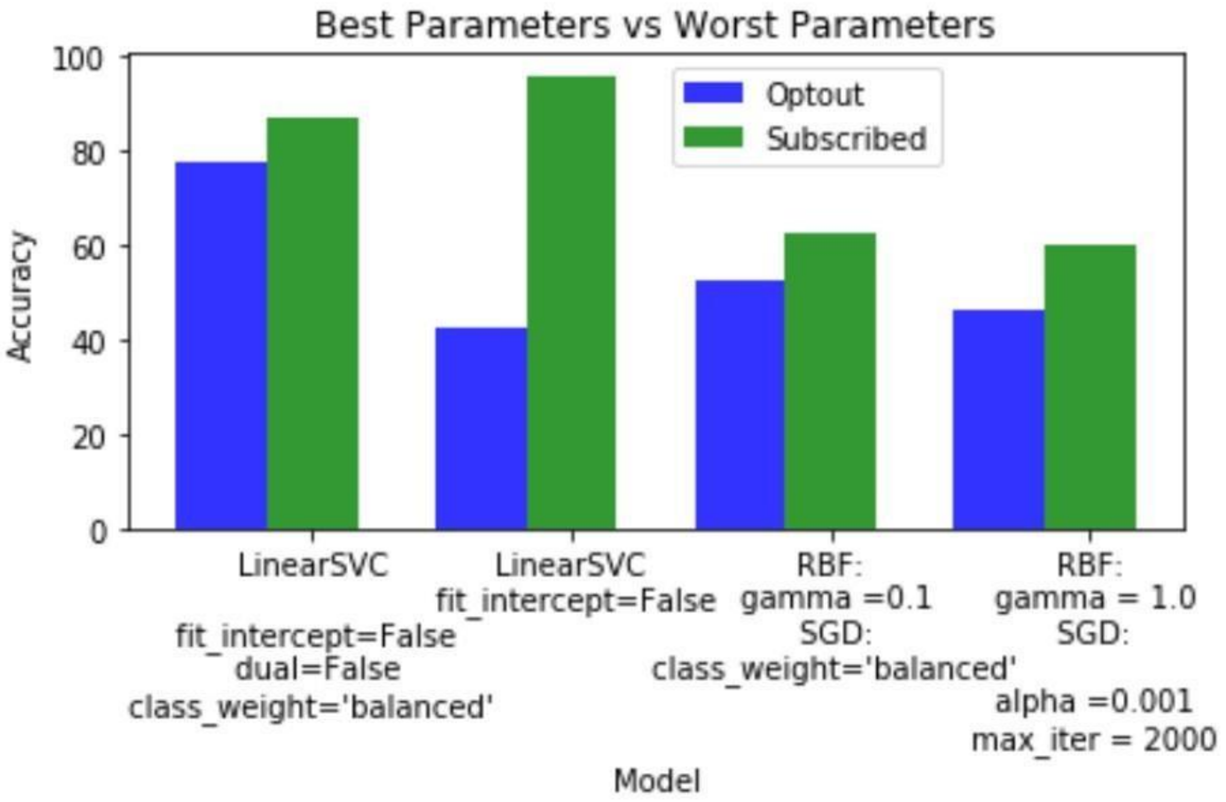
SVM-Final Results

- Above are some of the better parameters encountered while testing and their respective accuracies, for both chosen model types

- LinearSVC produced a good accuracy of 77.3% for Optout and 85.1% for Subscribed with the best parameters

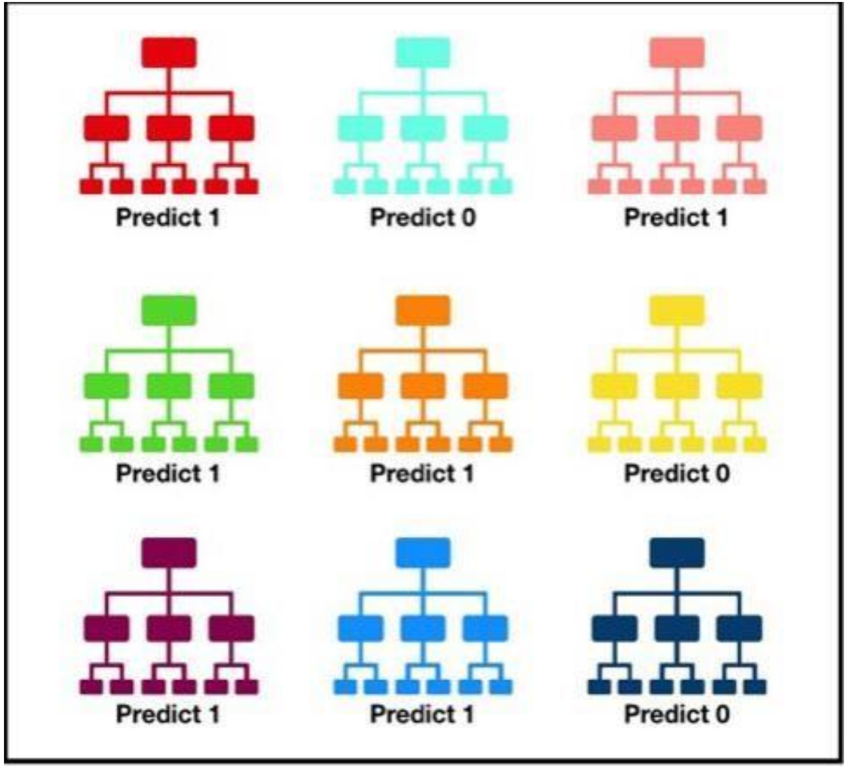
- RBF+SGD produced a mediocre accuracy of 53% for Optout and 62% for Subscribed with the best parameters

SVM-Final Results



Random Forest

- Random Forest consists of a large number of individual decision trees that operate as an ensemble.
- Each tree gives us a class prediction and the class with the most votes becomes our prediction
- Random Forest uses bagging and random features when building the individual trees to create a forest of uncorrelated trees.



Tally: Six 1s and Three 0s

Feature Extraction

- Recursive Feature Extraction (RFE):
    - Retrieving the top 3 features using the logistic regression algorithm.

- The features taken into account are:
  - Profile.Id,Signup,Last.Open,Last.Click,Opt out.Time,Opens,Clicks,Lifetime.Message,r eferral_source,Top.Device, Geolocation.City,Domain
- The target variable is the ‘Engagement’
- Retrieving the 3 most favorable features with the following ranks:

```
In [16]: model = LogisticRegression(solver='lbfgs')
         rfe = RFE(model, 3)
         fit = rfe.fit(X, Y)
         print("Num Features: %d" % fit.n_features_)
         print("Selected Features: %s" % fit.support_)
         print("Feature Ranking: %s" % fit.ranking_)
```

```
Num Features: 3
Selected Features: [False  True  True False  True False False False False False False False]
Feature Ranking: [10  1  1  2  1  5  6  7  8  9  4  3]
```

- Random Forest Algorithm produced an accuracy of 96.5% for subscribed and 80% for optout, which is the highest of all 3 types of models.

Conclusion

Decision Tree

- Opt-out Accuracy: 60.8%
- Subscribed Accuracy: 67.6%

SVM

- Opt-out Accuracy: 77.3%
- Subscribed Accuracy: 85.1%

Random Forest

- Opt-out Accuracy: 96.5%
- Subscribed Accuracy: 80%

References

1. https://towardsdatascience.com/optimizing-hyperparameters-in-randomforest-classification-ec7741f9d3f6
2. https://scikit-learn.org/stable/modules/kernel_approximation.html
3. https://www.geeksforgeeks.org/ml-one-hot-encoding-of-datasets-in-python/
4. http://chrisstrelioff.ws/sandbox/2015/06/08/decision_trees_in_python_with_scikit_learn_and_pandas.html
5. https://towardsdatascience.com/understanding-random-forest-58381e0602d2