Machine Learning Engineer Nanodegree

Capstone Project

Young Kyung Kim
September 21st, 2019

I. Definition

Project Overview

I am going to make a model to predict behavior of customers whether they would be retained or not. This data had been provided by IBM for teaching and data contains information about Telco customers. This kind of data and analysis is important for service provider in deciding marketing strategy to retain their customers. If they know which customers are not likely to be retained, then they could prevent this from happening by giving better offer to those customers. This kind of actions will prevent profit loss. I was personal motivated to investigate this problem because this dataset and problem seemed to be something I will be doing in company as a data scientist. In *Model of Customer Churn Prediction on Support Vector Machine,* Xia and Jin constructed a model to predict which customer will churn or not. It had build a model with support vector machine and talks about how different parameters in support vector machine that affect different metrics.

Citation

Zhao, Jing, and Xing-Hua Dang. “Bank Customer Churn Prediction Based on Support Vector Machine: Taking a Commercial Bank's VIP Customer Churn as the Example.” 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing, vol. 28, no. 1, Jan. 2008, pp. 71–77., doi:10.1109/wicom.2008.2509.

Problem Statement

One of the biggest concerns for many telephone service providers is retaining customers. To retain customers, the providers could provide retention program to current customers but it usually cost a lot of money. If the provider could know which customers will likely to leave, then it could save its cost on retention program by providing the program to them. In another word, identifying the possible leaving customers will increase efficiency for retention program. My hypothesis is that there will be a model that could predict whether a customer will likely to leave or not.
Y = f(x)
Y = Whether customer leaves or not
f(x) = prediction model

Since this is a classification problem, I will use support vector machine and random forest method to build a model to predict customer's behavior.

Metrics

In this section, propose at least one evaluation metric that can be used to quantify the performance of both the benchmark model and the solution model. The evaluation metric(s) you propose should be appropriate given the context of the data, the problem statement, and the intended solution. Describe how the evaluation metric(s) are derived and provide an

example of their mathematical representations (if applicable). Complex evaluation metrics should be clearly defined and quantifiable (can be expressed in mathematical or logical terms).
I will use accuracy score, which is explained in above, as an evaluation metrics.

The equation for accuracy score is

Acc = (True Positive + True Negative)/ (True Positive + False Positive + False Negative + True Negative)

True Positive = Whether the model had correctly predicted positive value
False Positive = When the model had predicted negatively when actual value is positive
True Negative = Whether the model had correctly predicted negative value
False Negative = Whether the model had predicted positively when actual value is negative

There are other metrics I should consider as recall and precision. Recall represents how well the model have predicted customer leaving among who actually left. Precision represents how well the model have predited customer leaving among who they predicted will leave. Below is the equations for each metrics

The equation for recall is
Recall = True Positive / (True Positive + False Negative)

The equation for precision is
Recall = True Positive / (True Positive + True Negative)

We should consider strengths of recall and precision in given problem before choosing one of the metrics. Let's assume that Telecom are going to send people, who are predicted to leave, an e-mail advertisement, which doesn't cost much. In this case, if we do not focus on recall, then we might miss some people actually leaving. This mistake will cost a lot to the company. However, if we do not focus on precision, then we might mistakenly send e-mail advertisement to people, who are not actually leaving. This mistake will not cost much. Therefore, I will consider recall as another metrics over precision.

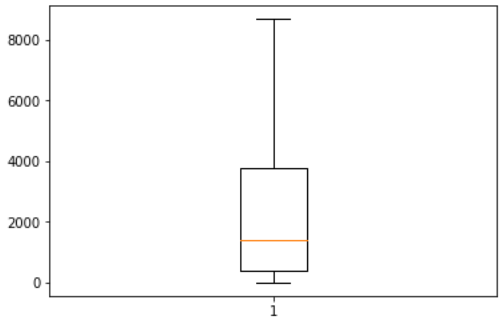
II. Analysis

Data Exploration¶

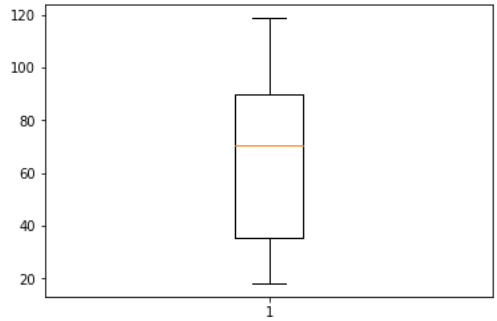
For this project, I will use dataset provided by IBM and Telco, which could accessed in the Kaggle(https://www.kaggle.com/blastchar/telco-customer-churn/home). The dataset contains information on whether customer leaving the service, what kind of services that customer has sign up for, customer's account information and customer's demographic information. For what kind of services that customer has sign up for, the data contains information whether they have signed up for phone or internet services. These kinds of information could be important. For example, customer who signed up for phone could have higher chance of leaving compared to customer who signed up for internet. Such information will be helpful for predicting whether customers will leave or not.

Similar to what kind of services that customer has sign up for, customer's account information could be important. Whether contract is monthly or yearly based could influence whether customers will leave or not. Also, customer's demographic whether they are married or not

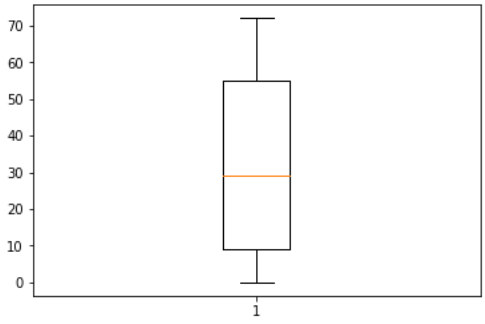
could influence a lot in predicting whether customers will leave or not. Even though it is not clear which kind of data would relevant in this project, it is our job to find out through exploration and testing.



Tenure



Monthly Charge



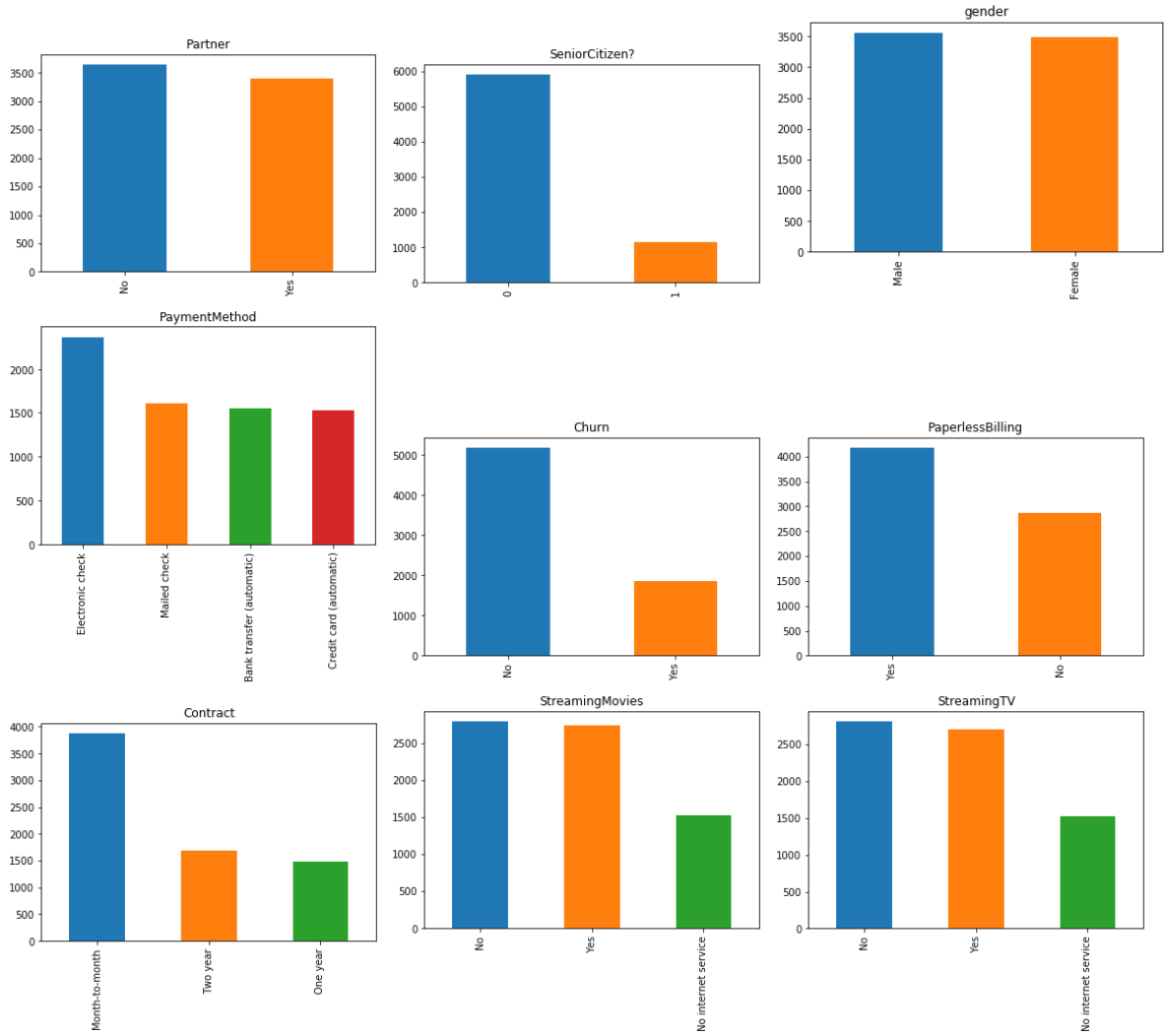
Total Charges

Given dataset has 7043 inputs with 21 variables. The dataset do not have any missing values but it had some empty string as the values. I have converted those empty string into 0. Dataset has various categorical variables. I am planning to change all those categorical variables into dummy variables. There are 3 numerical variables but from looking at boxplot none of them seems to have outliers. Yet, from looking at their boxplot it seems they might have distribution problem. It seems like the distribution is skewed right for all three features. Three features may need a normalization and I will explore the distribution futher in next section.
Below are list of variables in the dataset:

- "customerIDCustomer": ID
- "genderCustomer gender": (female, male)
- "SeniorCitizenWhether": the customer is a senior citizen or not (1, 0)
- "PartnerWhether": the customer has a partner or not (Yes, No)
- "DependentsWhether": the customer has dependents or not (Yes, No)
- "tenureNumber": of months the customer has stayed with the company
- "PhoneServiceWhether" the customer has a phone service or not (Yes, No)
- "MultipleLinesWhether" the customer has multiple lines or not (Yes, No, No phone service)
- "InternetServiceCustomer's": internet service provider (DSL, Fiber optic, No)
- "OnlineSecurityWhether": the customer has online security or not (Yes, No, No internet service)
- "OnlineBackupWhether": the customer has online backup or not (Yes, No, No internet service)

- "DeviceProtectionWhether": the customer has device protection or not (Yes, No, No internet service)
- "TechSupportWhether": the customer has tech support or not (Yes, No, No internet service)
- "StreamingTVWhether": the customer has streaming TV or not (Yes, No, No internet service)
- "StreamingMoviesWhether": the customer has streaming movies or not (Yes, No, No internet service)
- "ContractThe": contract term of the customer (Month-to-month, One year, Two year)
- "PaperlessBillingWhether": the customer has paperless billing or not (Yes, No)
- "PaymentMethodThe customer's": payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
- "MonthlyChargesThe": amount charged to the customer monthly
- "TotalChargesThe": total amount charged to the customer
- "ChurnWhether": the customer churned or not (Yes or No)

Exploratory Visualization



Most of categorical variables has at least 1,000 data points for each level, except for Phone Service and Multipe Line Variable. We should take account that maybe we might have too little data points for no level of Phone Service to represent the population for people, who don't have phone service. All the continuous variable do not have normal distribution. For Tenure variable distrubtion, it seems to be bimodal with two peak on 0 and 70. For Monthy Charges and Total Charges Histogram, it seems to be skewed right. However, both SVM and Random Forest technique do not require normal distribution but I will still normalize the data as model might give some features more importance over other features.

Algorithms and Techniques

Since dataset has output variable, which is Churn, I am going to use supervised learning instead of unsupervised learning. Also as output variable is categorical variable, I will use support vector machine and random forest to build a model.

Support Vector Machine

With given training data, support vector machine will find the optimized equation for a given function to classify training data. It will find the optimized equation through calculating and comparing score for different equations for the given function. For example, support vector machine will give negative points to misclassification of data points and give positive points to larger margin. Support Vector Machine will choose a equation that has highest total points. It is possible to change how support vector machine calculate points. For example, we could change how much support vector machine will give negative points to misclassification. We could set support vector machine to give large negative points to misclassification to find best equation to classify training data. Yet, this will cause overfitting, so it is important to try different negative points to misclassification to find the best one. Furthermore, we could try different function for classification. We could use linear or radial basis function to seperate two classes of data.

The following parameters could be tuned for optimization in support vector machine:

- Penalty: How much to give penalty for making mistake.
- Kernel: Method or equation to seperate two classification. (linear, polynomial,radial basis function, etc.)
- Degree: How many degree for polynomial kernel function.

Random Forest

Random forest is a combination of different decision trees that had been derived from the training data. Random forst will randomly choose data points and features to build decision trees. These decision trees will be combined to make one model. One of the advantage that random forest have over decision tree, especially that has large depth, is that it less prone to overfitting as it will consider different decision trees. Just like support vector machine, we could change how random forest work. For example, we could choose how many tress to construct before combining them. We could also set how many features to use in building individual tree.

The following parameters could be tuned for optimization in random forest:

- Number of estimators: How many trees to construct.
- Criterion: Which technique to use to split data.
- Max_features: How many features to use build a model.
- Max_depth: Maxium depth of nodes
- min_samples_leaf: Minimum number of samples required to be at a leaf node.

For both model, I am first going to preprocess data for training. Then I will set variables for Grid Search, which will help me to find the best parameter values for each model.

I will compare the result and choose the best one.

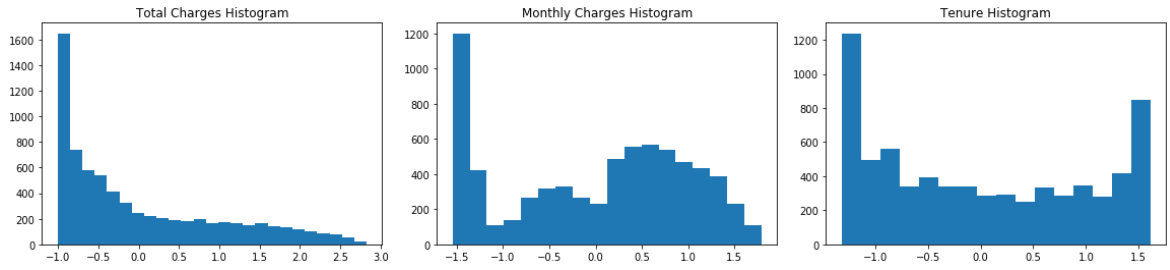
Benchmark

Since this data has been posted on Kaggle, there are a lot of model posted online. One of the posted model used various methods to solve the problem. For example, it used regression and random forest. It achieved about 70% of accuracy score. The equation for accuracy score is Acc = (True Positive + True Negative)/ (True Positive + False Positive + False Negative + True Negative) True Positive = Whether the model had correctly predicted positive value False Positive = When the model had predicted negatively when actual value is positive True Negative = Whether the model had correctly predicted negative value False Negative = Whether the model had predicted positively when actual value is negative

III. Methodology

Data Preprocessing

- I will normalize the continuous variable as model might give some features more importance over other features. Looking at below diagram, we could notice that value has normalized.
- I have replaced binary variable(yes or no) into 1 for yes and 0 for no.
- For categorical variables that can take more than two values, I have made dummies variables for them. Looking at below diagram, we could notice the dummy variables being created.
- I have also divided the dataset into training and test set with the ratio of 8:2. 5634 samples have been assigned to training set and 1409 samples have been assigned to test set.
- Correlated features do not affect accuracy of classification accuracy, so I didn't conduct feature selections.




```
Training set has 5634 samples with 40 variables
Test set has 1409 samples with 40 variables
```

Implementation

Support Vector Machine

- Load data.
- Preprocess the data as above.
- Create Support Vector Machine with default value.
    - Penalty: 1
    - Kernel: 'rbf(radial basis function)'
    - Gamma: 'auto'

- Insert preprocessed training dataset into Support Vector Machine to train.
- Predict classification of test dataset through inseting test dataset into trained model.
- With the prediction, calculate acuracy and recall score through comparing to real outcome.
    - Got accuracy score of 81.41%.
    - Got recall score of 51.74%Random Forest

Random Forest¶

- Load data.
- Preprocess the data as above.
- Create Random Forest with default value.

    - Number of estimators: 10
    - Criterion: 'gini'
    - Max_features: 'auto'
    - Max_depth: 'None'
    - min_samples_leaf: 'None'
- Insert preprocessed training dataset into Support Vector Machine to train.
- Predict classification of test dataset through inseting test dataset into trained model.
- With the prediction, calculate acuracy and recall score through comparing to real outcome.

Refinement

In order to increase accuracy score and recall score, I have used Grid Search technique. First I have set list of parameters to test for each algorithm, which are listed below.

Parameters for Support Vector Machine:

```
- Penalty: [0.1,0.5,1,1.5,2]

- Kernel: ['linear', 'poly', 'rbf', 'sigmoid']

- Degree: [3,4,5,6,7,8,9]

- Gamma: ['auto',0.03,0.05,0.75,0.1]
```

Parameters for Random Forest:

```
- Number of estimators: [5,10,15,20,25,30]

- Criterion: ['gini', 'entropy']

- Max_features: [5,10,15,20,30,35]

- Max_depth: [5,10,15,20]

- min_samples_leaf: [2,5,10,15]
```

I have set accuracy score as metric for the Grid Search.
Grid Search technique is going through all given parameters into the model to find the parameter with best accuracy score and recall. After finding the best parameters it will save those parameters. These parameters and accuracy score and recall score are listed below.

Support Vector Machine with accuracy score as scoring method.

- Penalty: 2
- Kernel: 'rbf(radial basis function)'
- Gamma: 0.05

Support Vector Machine with recall score as scoring method.

- Penalty: 1.5
- Kernel: 'sigmoid'
- Gamma: 0.025

Random Forest with accuracy score as scoring method.

- Number of estimators: 15
- Criterion: 'entrophy'
- Max_features: 5
- Max_depth: 10
- min_samples_leaf: 1

Random Forest with recall score as scoring method.

- Number of estimators: 15
- Criterion: 'entrophy'
- Max_features: 15
- Max_depth: 10
- min_samples_leaf: 1

IV. Results

Model Evaluation and Validation

Result from Grid Search (scoring method: accuracy score)

Support Vector Machine

Accuracy score: 81.41%
Recall score: 52.82%

Parameter for optimized Support Vector Machine

- Penalty: 2

- Kernel: 'rbf(radial basis function)'
- Gamma: 0.05

Random Forest

Accuracy score: 80.62%
Recall score: 53.35

Parameter for optimized Random Forest

- Number of estimators: 15
- Criterion: 'entropy'
- Max_features: 5
- Max_depth: 10
- min_samples_leaf: 1

Result from Grid Search (scoring method: recall score)

Support Vector Machine

Accuracy score: 80.77%
Recall score: 61.93%

Parameter for optimized Support Vector Machine

- Penalty: 1.5
- Kernel: 'sigmoid'
- Gamma: 0.025

Random Forest

Accuracy score: 80.62%
Recall score: 55.5%

Parameter for optimized Random Forest

- Number of estimators: 15
- Criterion: 'entropy'

- Max_features: 15

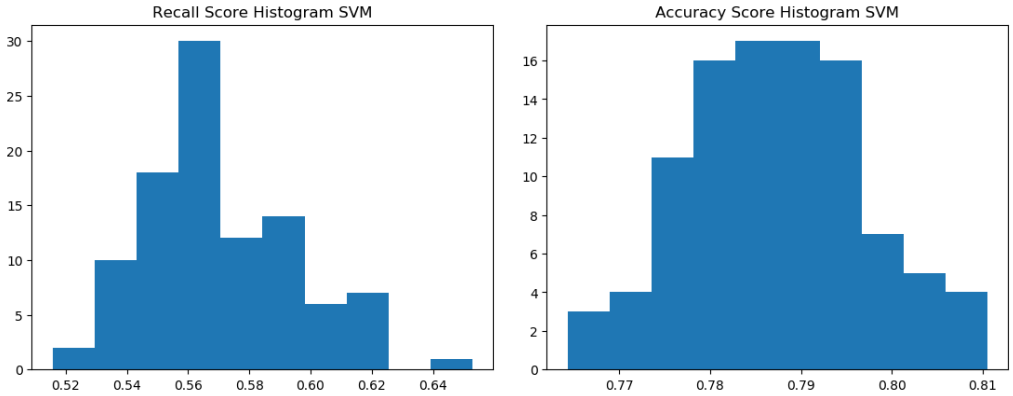
- Max_depth: 10

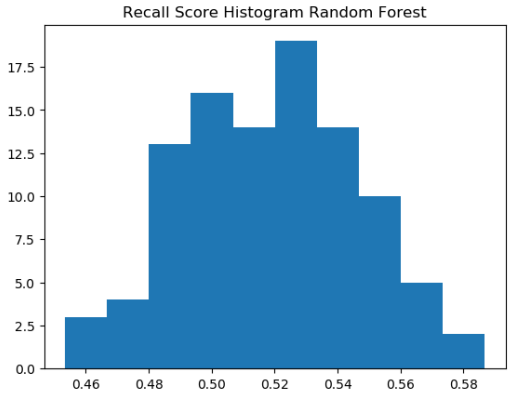
- min_samples_leaf: 1

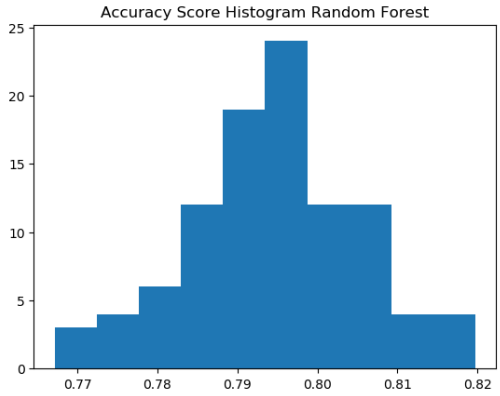
From comparing accuracy score and recall score SVM is better method in prediction compared to Random Forest. Among two SVM model, I am going to choose SVM that has been optimized with recall score. It has 9% higher score in recall score while less than !% lower score in accuracy score compared to another SVM model. As mention in above, if advertisement doesn't cost much, then identifying person who will leave will be more important than identifying person who will not leave. In that sense, recall score will be important to company profit. Thus I will choose SVM that has been optimized with recall score as the final model.

Robustness

In order to check robustness, I have randomly seperated data into training and test dataset differently for 100 times. I fitted training dataset into the Support Vector Machine and Random Forest with the parameters optimized by using recall score as the scoring method. From the trained model, I got 100 recall and accuracy score for Support Vector Machine and Random Forest. Looking at the histograms below the distribution of those 100 score is normal. Thus, model generalize well to unseen data. It also doesn't seemed to be sensitive toward outliers as the score doesn't change much from looking at the distribution. With this data, we can trust our model will have score in between certain range. Also I have compared whether there is a significant difference between two model in score. With p-value of 7.401479555314805e-30, I can confidently state that there is significant difference between two model in recall score. Support Vector Machine had higher recall score in average. With p-value of 6.01881657323168e-07, I can confidently state that there is significant difference between two model in accuracy score. Random Forest had higher accuracy score in average.







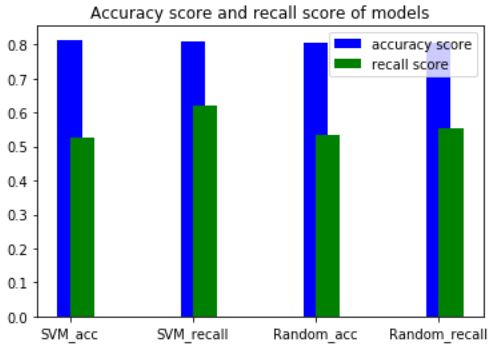
Justification

Benchmark result had 70% accuracy score and final model had about 10% higher accuracy score compared to benchmark model. In accuracy score, final model had better performance than benchmark model. It was impossible to compare recall score between benckmark model and final model as benchmark model didn't have recall score. However, if we are assuming if advertisement didn't cost much, then recall score would be important. For example, if we had 2,000 people who are planning to churn out, then 10 percent lower in recall score would mean failing to identifying 200 people. Considering that each customer pay about 2279.7 dollar in average, having 10 percent lower in recall score could cost about 455940 dollar. I have chosen final model considering those factors.

V. Conclusion

Free-Form Visualization

Looking at the below plot, we can notice that there isn't much difference in accuracy score among 4 models. However, there is noticeable difference in recall score. For example, support vector machine optimized with recall score has about 10% higher recall score compared to other models.



Reflection

Process:

1. Import data and identify problem
2. Identify metric and search benchmark model to solve the problem
3. Explore data
4. Preprocess data for training

5. Train data using gridsearch method
6. Compare models with different metric
7. Choose the best model.

One of the difficult aspects of the project was preprocess data part as it was hard for me to find just empty value. I had to go through each cell to find it out. One of the interesting fact I found was that changing scoring method for grid search could dramtically change the outcome of the optimized models.

Improvement

One of the improvement I could make is to consider different possible situation. I have only considered a situation that advertisement wouldn't cost much. However, if advertisement cost a lot than I should consider other metrics as precision and F1 score, which consider recall and precision score. Thus I will use F1 score as a scoring method for grid search instead of recall score.