ISYE 6740 – Fall 2023
Project Final Report

Team Member Names: Wenyu Sui (last 4 digits of GTID: 4094)

Project Title: Telecom Customer Churn Prediction

I. Problem Statement

The industry of telecommunications is competitive. In this industry, customers are able to make their selections between different service providers. They may actively switch from one phone carrier to another because of price differences, service qualities and other factors. Typically, phone carriers experience an average annual customer churn rate of 15-25%. [1] In 2020, this number was 21%. [2] Since acquiring a new customer normally costs 5-10 times more than retaining an existing customer [1], it is extremely important for phone carriers to improve their client retention rate.

One important issue involved in retaining the customers is to detect those customers who are likely to change service providers in the near future. The phone carriers could reach out to these customers and retain them by offering special discounts or improving services based on their feedback, if the phone carriers could predict that these customers would churn.

The task of this project is to develop a machine learning model which helps phone carriers to predict their customer churns. The model will take each customer's temporal behavioral data as input and predict if the customer will churn in the future. We will develop a few models using different algorithms, use cross validations to measure their performance, and select one model that performs best.

II. Data Source

The dataset used in this project is obtained from the following Kaggle competition:
https://www.kaggle.com/competitions/telecom-churn-case-study-hackathon-c52/overview
We are going to use the train.csv dataset to perform our analysis. [1]

This dataset contains 172 variables collected from 69,999 telecom customers from June to August in 2014. The target variable $churn\_probability$ contains a value of 0 or 1, where 0 represents "not churn" and 1 represents "churn" – this value is what our machine learning model needs to predict. The other 171 variables contain temporal behavioral data of the telecom customers. These variables can be used as predicting variables in our machine learning model.

The table in Appendix I – Abbreviations of Column Definitions contains the various abbreviations that help us understand the meaning of each variable. For example, the column $onnet\_mou\_8$

contains "*onnet*", "*mou*" and "*8*" in the column name. Therefore, it represents the minutes of usage of voice calls within the same operator network in August.

III. Research Methodology

This section will discuss our methodologies to develop and select the model that predicts telecom customer churns. It contains three subsections: III.1 Data Preprocessing, III.2 Classification Algorithms and III.3 Key Performance Metrics.

III.1 Data Preprocessing

Here are the key steps we take to preprocess the data:

1. **Split training and testing data.** Since the test dataset provided by the original data source [1] doesn't include the target variable $churn\_probability$, we need to create our own test dataset. We are going to use 80% of the train data (55,999 samples) provided by the original data source as our training dataset, and the rest 20% (14,000 samples) as our testing dataset. The data is shuffled before being split.

2. **Exclude useless columns.** Some columns are meaningless for analysis and need to be removed from the training data. These columns are typically calendar dates of the last day in each month, columns with one single value that doesn't have any variations, or columns with too many missing values. (number of missing values greater than 15% of the sample size in training dataset)

   For example, the columns $date\_of\_last\_rech\_6$, $date\_of\_last\_rech\_7$ and $date\_of\_last\_rech\_8$ simply show the date of the last day in June, July and August of 2014. These columns need to be removed since they don't provide any useful information for our analysis.

   In this step, 47 columns are removed from the training datasets. Please refer to Appendix II – Column Removed from Original Dataset to see the names of the removed columns.

3. **Impute missing values.** For each column in the training dataset, if the number of missing values is smaller than 15% of the sample size, then we will impute the missing values by setting them equal to the mean value of the rest of the data in the same column.

4. **Standardize the data.** Since the variables are measured in different scales, we need to standardize each variable in the training dataset by subtracting its sample mean and dividing by its sample standard deviation.

5. **Principal component analysis. (PCA)** The processed training dataset contains 124 predicting variables in total, and many of these variables are correlated to each other as displayed in the heatmap below. Therefore, we use PCA to reduce dimensions of the data and eliminate collinearity.

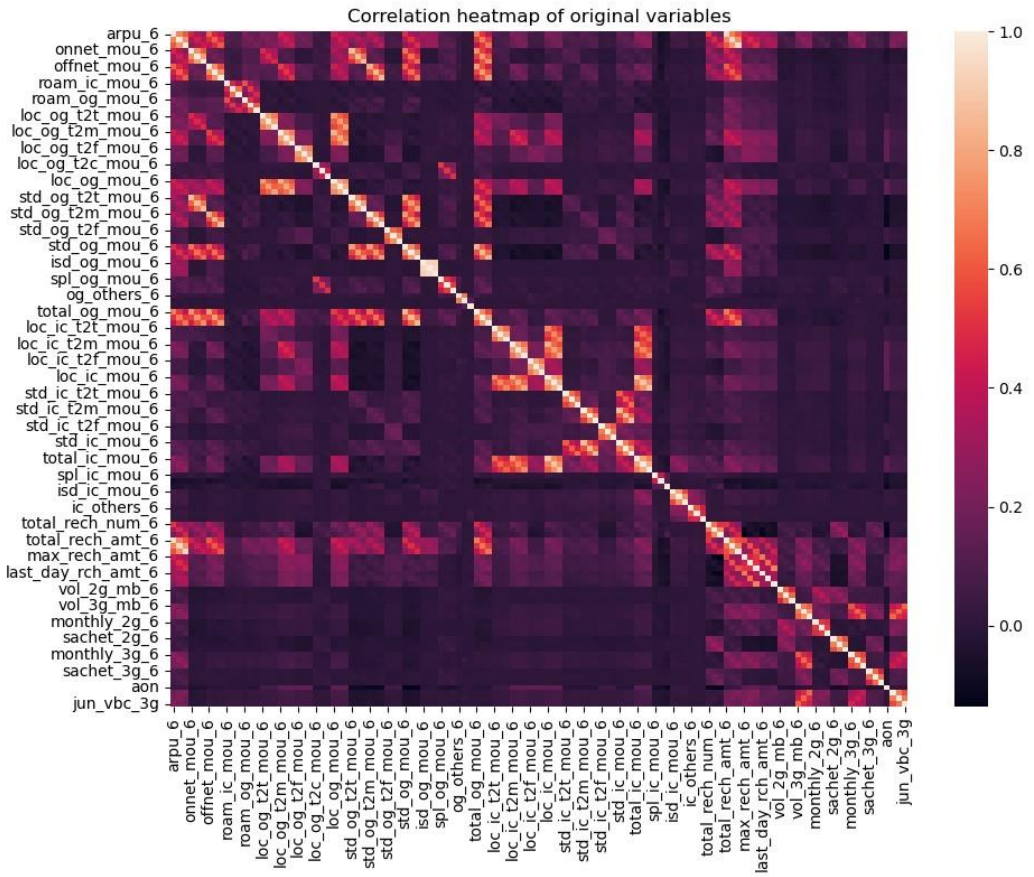

Figure 1 Correlation Heatmap of Original Variables

Here, we choose to use the first 70 PCA variables returned by $sklearn.decomposition.PCA$ [3], which explain 95.67% of variance in the original dataset. The variables processed by PCA may not be used in every classification model in this project. They will only be used in the models that are easily influenced by multicollinearity such as logistic regression and linear SVM.

6. **Apply steps 2-5 to the test dataset.** To make sure that the test dataset can be used as input to our models in later steps, we need to apply steps 2-5 to the test dataset in exactly the same order and scale as what we did to the training dataset.

7. **Divide the training dataset for 5-fold cross validation.** We shuffle and divide the training dataset into five subsets with equal sample size. These five subsets of data will later be used for the 5-fold cross validation.

III.2 Classification Algorithms

This project tries to predict whether the telecom customers will churn or not, which is a typical classification problem in machine learning. Therefore, we are going to develop the following classification models and compare their performance using Python package *scikit-learn* [4]:

> *Logistic Regression, K-nearest Neighbors (KNN), Neural Network, Random Forest, AdaBoost and Support Vector Machine (SVM)*

The model that achieves the best performance in the 5-fold cross validation will be selected as our final model and used for further analysis.

One challenge is that our original dataset is very imbalanced. Only 7,132 out of 69,999 samples have 1 in column $churn\_probability$ while all the other samples have 0. To avoid any influence caused by the imbalanced weights of 0 and 1, we need to tweak some hyperparameters of certain models. Here are the details of each model:

1. Logistic Regression

In statistics, the logistic model (or logit model) is a statistical model that models the probability of an event taking place by having the log-odds for the event be a linear combination of one or more independent variables. [5]

Here, we fit a logistic regression model using the 70 PCA variables created previously as the independent variables and $churn\_probability$ as the dependent variable. To avoid any influence caused by the imbalanced weights of 0 and 1 in $churn\_probability$, we adjust the weights of each class (0 and 1) inversely proportional to their class frequencies in the input data. Since there are about 10.19% samples having 0 and about 89.81% samples having 1, we assign a weight of 9.81[①] to each sample having 0 and a weight of 1.11[②] to each sample having 1 when we fit the logistic regression model.

2. K-nearest Neighbor (KNN)

The KNN algorithm is a non-parametric supervised learning method used for classification. For each data point in the test dataset, its predicted class is determined by the majority class of other k "nearest" data points from the training data.

In this project, we fit a KNN model using Euclidian distance (l2 norm) to measure the distance between data points in a 70-dimensioanl space. The dimensions are measured by the 70 PCA variables created previously. The weights of the training data points are assigned inversely of their distance. In other words, closer neighbors of a test data point will have a greater influence on the prediction made by the model.

We also try each value of $k = 5, 6, 7, 8, 9, 10$ to see if the model performs differently as the k value changes.

---

① $9.81 \approx 1 \div 10.19\%$

② $1.11 \approx 1 \div 89.81\%$

3. Neural Network

Artificial neural networks (or neural network) are a branch of machine learning models that are built using principles of neuronal organization discovered by connectionism in the biological neural networks constituting animal brains. [6]

In this project, we train a fully connected neural network model with 6 hidden layers. There are (200, 100, 50, 30, 30, 30) nodes in each layer. The model is trained with the predicting variables in their original scales. It uses the adaptive learning rate and takes the rectified linear unit function as the activation function. It will also terminate training early to avoid overfitting when the validation score is not improving.

4. Random Forest

Random forest is an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. [7]

In this project, we construct multiple random forest models with different number of decision trees and compare their performance. We construct random forests with 50, 100, 150 and 200 trees respectively. The random forest models are fitted with the predicting variables in their original scale.

5. AdaBoost

AdaBoost is a statistical classification meta-algorithm for binary classification. It is adaptive in the sense that subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. The output of the weak learners is combined into a weighted sum that represents the final output of the boosted classifier. [8]

In this project, we use decision stump as the "weaker learner" of AdaBoost algorithm. We construct AdaBoost models with 50, 100, 150 and 200 decision stumps respectively. All the AdaBoost models are trained with the predicting variables in their original scale.

6. Support Vector Machine (SVM)

Support Vector Machines (SVMs) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. [9] The linear SVM model searches for hyperplanes in high-dimensional data, which divide the training dataset into multiple classes.

In this project, we fit two SVM models. The first SVM model uses the linear kernel. The regularization parameter (C value) is set equal to 1.

The second SVM model uses the Radial Basis Function (RBF) kernel, which converts the model's decision boundary from linear to non-linear. The regularization parameter (C value) is set equal to 0.7. The kernel coefficient (gamma value) is set to be 0.5.

In both the SVM models, we adjust weights of each class (0 and 1) inversely proportional to their class frequencies in the input data. This step is to avoid any issues caused by the imbalanced class weights in the column $churn\_probability$. Both the SVM models are fitted with the 70 PCA variables created previously.

III.3 Key Performance Metrics

To avoid any issue caused by overfitting, we use 5-fold cross validation to compare the performance of the models. We will use Classification Accuracy as the most important performance metric, which is defined as below:

$$Classification\ Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

In the formula above, "positive" means that the customer has churned ($churn\_probability$ = 1), and "negative" mean that the customer has not churned ($churn\_probability$ = 0).

Another metric that we may consider is Sensitivity, which is defined below:

$$Sensitivity = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

There are two reasons why we are interested in the Sensitivity of the model. First, in this problem, we are more concerned about identifying the customers who will churn in the future than identifying the customers who will not churn. Therefore, we would like the model to correctly identify as many churned customers as possible. Second, our dataset is imbalanced. It contains too many unchurned customers and too few churned customers. In this situation, a model can easily achieve a very high Classification Accuracy (but very low Sensitivity) in cross validation by predicting all the input datapoints to be unchurned.

In the end, the model that achieves the highest classification accuracy and a reasonably high sensitivity in the 5-fold cross validation will be selected as our final model.

IV. Model Selection

The following table summarizes and compares the performance of each model in the 5-fold cross validation.[3] The values of True Positive, True Negative, False Positive and False Negative are the numbers of predictions aggregated from all the five folds of test data.

[3] In the first column of Table 1, the k value represents the number of neighbors used in the KNN model. For the Random Forest models, the n value represents the number of decision trees. For the AdaBoost models, the n value represents the number of decision stumps.

Classifier	True Negative	False Positive	False Negative	True Positive	Total	Classification Accuracy	Sensitivity
Logistic Regression - PCA variables	38,497	11,803	987	4,712	55,999	77.16%	82.68%
KNN - PCA variables, k = 5	48,856	1,444	3,289	2,410	55,999	91.55%	42.29%
KNN - PCA variables, k = 6	48,890	1,410	3,221	2,478	55,999	91.73%	43.48%
KNN - PCA variables, k = 7	49,056	1,244	3,357	2,342	55,999	91.78%	41.09%
KNN - PCA variables, k = 8	49,077	1,223	3,331	2,368	55,999	91.87%	41.55%
KNN - PCA variables, k = 9	49,194	1,106	3,412	2,287	55,999	91.93%	40.13%
KNN - PCA variables, k = 10	49,208	1,092	3,392	2,307	55,999	91.99%	40.48%
Neural Network - original variables	49,096	1,204	2,535	3,164	55,999	93.32%	55.52%
Random Forest - original variables, n = 50	49,305	995	2,291	3,408	55,999	94.13%	59.80%
Random Forest - original variables, n = 100	49,320	980	2,302	3,397	55,999	94.14%	59.61%
Random Forest - original variables, n = 150	49,324	976	2,283	3,416	55,999	94.18%	59.94%
Random Forest - original variables, n = 200	49,323	977	2,299	3,400	55,999	94.15%	59.66%
AdaBoost - decision stump with original variables, n = 50	49,137	1,163	2,217	3,482	55,999	93.96%	61.10%
AdaBoost - decision stump with original variables, n = 100	49,124	1,176	2,221	3,478	55,999	93.93%	61.03%
AdaBoost - decision stump with original variables, n = 150	49,105	1,195	2,236	3,463	55,999	93.87%	60.77%
AdaBoost - decision stump with original variables, n = 200	49,066	1,234	2,220	3,479	55,999	93.83%	61.05%
Linear SVM - PCA variables	39,490	10,810	969	4,730	55,999	78.97%	83.00%
Kernel SVM - PCA varaibles, RBF kernel	49,267	1,033	3,942	1,757	55,999	91.12%	30.83%

Table 1 Model Performance Comparison using 5-Fold Cross Validation

It can be observed that the Random Forest model with 150 decision trees performs best among all the models in the cross validation. It achieves an overall classification accuracy of 94.18%, and it also achieves 59.94% sensitivity. Therefore, we would like to choose this model as our final model and further measure its performance on the test dataset.

V. Evaluate Model Performance on the Test Dataset

In this section, we use the test dataset to measure the performance of the model selected in the previous section. The test dataset contains 124 predicting variables, along with the target variable $churn\_probability$, of 14,000 samples.

First, we use all the samples in the training dataset to fit a random forest model with 150 decision trees. Then, we feed the model with predicting variables from the test dataset to predict the values of $churn\_probability$. Finally, we compare the predictions made by the model and the true values of $churn\_probability$ in the test dataset. The confusion matrix, classification accuracy and sensitivity of the model are displayed as below:

		PREDICTED 0	PREDICTED 1
TRUE	0	12,347	220
	1	603	830

Classification Accuracy =	94.12%
Sensitivity =	57.92%

Table 2 Confusion Matrix, Accuracy and Sensitivity of Random Forest with 150 Decision Trees, Measured Using Test Dataset

It can be observed that the random forest model achieves an overall accuracy of 94.12% and sensitivity of 57.92%. Since the samples in the test dataset were not used to train the model,

we except that the model will achieve similar performance when it makes predictions on the new unseen samples in the real world.

VI. Variables by Importance

One property of the random forest model is that it can identify the impurity-based importance of each variable in the process of making a classification prediction. Here are the 20 most important variables and their impurity-based feature importance for the random forest model fitted in the previous section. The variables are ranked by their importance from the highest to the lowest.

Rank	Variable	Impurity-Based Feature Importance
1	total_ic_mou_8	0.082
2	total_og_mou_8	0.069
3	arpu_8	0.034
4	roam_og_mou_8	0.030
5	total_rech_amt_8	0.029
6	isd_og_mou_8	0.028
7	og_others_8	0.028
8	roam_ic_mou_8	0.027
9	max_rech_amt_8	0.025
10	loc_ic_mou_8	0.023
11	loc_ic_t2m_mou_8	0.018
12	total_ic_mou_7	0.016
13	last_day_rch_amt_8	0.016
14	aon	0.013
15	offnet_mou_8	0.012
16	isd_ic_mou_8	0.011
17	total_rech_num_8	0.011
18	loc_og_mou_8	0.010
19	arpu_7	0.010
20	roam_ic_mou_7	0.010

Table 3 The 20 Most Important Variables in Making Classification Decisions

It is not surprising that the *"total minutes of incoming voice calls in Aug"* and the *"total minutes of outgoing voice calls in Aug"* are the most important two variables in predicting whether a customer will churn or not. This finding is intuitive, because a customer is not likely to suddenly change their service carrier if they are heavily using their current service plan.

It can also be observed that the dollar amounts spent by the customers play an important role in predicting customer churns. Variables such as *"total recharge amount in Aug"*, *"maximum recharge amount in Aug"* and *"average revenue per user in Aug"* are all ranked highly in Table 3. However, we cannot determine whether a customer spending more money is more likely to churn or less likely to churn, compared to customers spending less money. One possible

scenario is that customers spending more money are more likely to churn since their service plans are more expensive. Another possible scenario is that these customers are not likely to churn since they like their current service plan and thus pay a lot for it. In short, it may take some additional investigations to measure the true impact of these variables related to customer spending.

Another finding is that the data of the most recent period is more important than the data from earlier periods. The original dataset contains variables of June, July and August. In Table 3, 16 out of 20 most important variables are data of August. Three variables are data of July. None of the most important 20 variables are data of June.

This section only provides a very brief discussion about the findings from the variable importance of the random forest model. Our model achieves good performance and identifies the important input variables, but it does not clearly explain why the customers decide to churn. In the real world, it is worthwhile to further research and measure the true impact of the important variables identified by the machine learning model. Business decision makers would like to know whether the change in a variable will urge the customers to churn or hold the customers back. This will help them better understand why the customers choose to leave, and how they can improve their business process.

VII. Conclusions and Future Work

In this project, we developed machine learning models using classification algorithms to help telecom service providers to predict future customer churns. The models take the temporal behavioral data of the telecom customers as input and predict whether the customers will churn in the future.

We developed models including logistic regression, k-nearest neighbors, neural network, random forest, AdaBoost and support vector machines. We compared the performance (measured by classification accuracy and sensitivity) of these models using 5-fold cross validation and found that the random forest model with 150 decision trees performed best among all the models. Therefore, we selected this model as our final model and measured its performance using a different test dataset. The model achieved an overall classification accuracy of 94.12% and sensitivity of 57.92% on the test dataset.

In the last section, we also briefly discussed the implications of the impurity-based importance of the input variables in the random forest model.

To further improve the performance of the machine learning model and create more values for its users, we believe that the following steps will be beneficial to perform:

1. Keep tuning the hyperparameters of the machine learning models mentioned in this project. Certain models could perform better than before or even outperform the selected final model after tuning.

2. Collect more types of data from the customers to train the model. For example, the telecom service providers can collect demographic data from their customers and include these data as predicting variables to train the model.

3. Further investigate the impact of the important variables identified by the classification model. This helps the business decision makers to better understand why the customers choose to churn and how to improve the business process.

4. Operationalize the machine learning model by letting it run on a schedule and always make predictions with the latest data.

Appendix I – Abbreviations of Column Definitions

Acronyms	Description
CIRCLE_ID	Telecom circle area to which the customer belongs to
LOC	Local calls - within same telecom circle
STD	STD calls - outside the calling circle
IC	Incoming calls
OG	Outgoing calls
T2T	Operator T to T, i.e., within same operator mobile to mobile
T2M	Operator T to other operator mobiles
T2O	Operator T to other operator fixed line
T2F	Operator T to fixed lines of T
T2C	Operator T to its own call center
ARPU	Average revenue per user
MOU	Minutes of usage - voice calls
AON	Age on network - number of days the customer is using the operator T network
ONNET	All kind of calls within the same operator network
OFFNET	All kind of calls outside the operator T network
ROAM	Indicates that customer is in roaming zone during the call
SPL	Special calls
ISD	ISD calls
RECH	Recharge
NUM	Number
AMT	Amount in local currency
MAX	Maximum
DATA	Mobile internet
3G	3G network
AV	Average
VOL	Mobile internet usage volume in MB
2G	G network
PCK	Prepaid service schemes called - PACKS
NIGHT	Scheme to use during specific night hours only
MONTHLY	Service schemes with validity equivalent to a month
SACHET	Service schemes with validity smaller than a month
*.6	KPI for the month of June
*.7	KPI for the month of July
*.8	KPI for the month of August
FB_USER	Service scheme to avail services of Facebook and similar social networking sites
VBC	Volume based cost - when no specific scheme is not purchased and paid as per usage

Appendix II – Column Removed from Original Dataset

id
last_date_of_month_6
last_date_of_month_7
last_date_of_month_8
date_of_last_rech_6
date_of_last_rech_7
date_of_last_rech_8
date_of_last_rech_data_6
date_of_last_rech_data_7
date_of_last_rech_data_8
circle_id
loc_og_t2o_mou
std_og_t2o_mou
loc_ic_t2o_mou
std_og_t2c_mou_6
std_og_t2c_mou_7
std_og_t2c_mou_8
std_ic_t2o_mou_6
std_ic_t2o_mou_7
std_ic_t2o_mou_8
total_rech_data_6
total_rech_data_7
total_rech_data_8
max_rech_data_6
max_rech_data_7
max_rech_data_8
count_rech_2g_6
count_rech_2g_7
count_rech_2g_8
count_rech_3g_6
count_rech_3g_7
count_rech_3g_8
av_rech_amt_data_6
av_rech_amt_data_7
av_rech_amt_data_8
arpu_3g_6
arpu_3g_7
arpu_3g_8
arpu_2g_6
arpu_2g_7
arpu_2g_8
night_pck_user_6
night_pck_user_7
night_pck_user_8
fb_user_6
fb_user_7
fb_user_8

References

[1] T. Jain, "Telecom Churn Case Study Hackathon - Predict churning customers for a Telecom company based on temporal behaviour," Upgrad, 12 Sep 2023. [Online]. Available: https://www.kaggle.com/competitions/telecom-churn-case-study-hackathon-c52/overview. [Accessed 17 Sep 2023].

[2] Statista Research Department, "Customer churn rate in the United States in 2020, by industry," statista, 6 Jul 2022. [Online]. Available: https://www.statista.com/statistics/816735/customer-churn-rate-by-industry-us/. [Accessed 17 Sep 2023].

[3] scikit-learn, "sklearn.decomposition.PCA," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html. [Accessed 19 Nov 2023].

[4] scikit-learn, "scikit-learn: Machine Learning in Python," [Online]. Available: https://scikit-learn.org/stable/about.html#people. [Accessed 1 Nov 2023].

[5] Wikipedia, "Logistic regression - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Logistic_regression. [Accessed 19 Nov 2023].

[6] Wikipedia, "Artificial neural network - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Artificial_neural_network. [Accessed 19 Nov 2023].

[7] Wikipedia, "Random forest - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Random_forest. [Accessed 19 Nov 2023].

[8] Wikipedia, "AdaBoost - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/AdaBoost. [Accessed 19 Nov 2023].

[9] Wikipedia, "Support vector machine - Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine. [Accessed 19 Nov 2023].