Customer Churn Prediction and Classification

| Priya Jani  Ahmedabad University  Ahmedabad, India  [priya.j@ahduni.edu.in](mailto:priya.j@ahduni.edu.in) | Yansi Memdani  Ahmedabad University  Ahmedabad, India  [yansi.m@ahduni.edu.in](mailto:yansi.m@ahduni.edu.in) | Priyanshu Pathak  Ahmedabad University  Ahmedabad, India  [priyanshu.p@ahduni.edu.in](mailto:priyanshu.p@ahduni.edu.in) | Mohnish Mirchandani  Ahmedabad University  Ahmedabad, India  [mohnish.m@ahduni.edu.in](mailto:mohnish.m@ahduni.edu.in) |
| --- | --- | --- | --- |

***Abstract – The following report provides an overview and implementation of the common algorithms like Logistic Regression, SVM and Naive Bayes which are used for prediction for the problem of customer churn classification. It compares the same across multiple parameters and shows the differences and similarities of the algorithms in their computations.***

***Keywords - Customer Churn, Classification, Prediction, Logistic Regression, Support Vector Machine, Naive Bayes Classification***

1. INTRODUCTION

Customer churn, also referred to as subscriber churn or logo churn, refers to the proportion of subscribers who terminate their subscriptions and is commonly expressed as a percentage. Customer churn prediction and analysis is one of the foremost and widespread applications of classical machine learning. Customer churn is a critical metric that can display customer satisfaction at the macro scale. Additionally, the telecom sector generally sees more significant churn rates than other sectors. This creates a large-scale requirement for better prediction models.

1. LITERATURE SURVEY

Customer Churn prediction has been a problem of the data transfer age with prominent research across the problem statement. Network analysis for customer churn is one-way networks connected by the similarity of churning members used to predict the possibility of churn. Classical Machine Learning algorithms provide a proven approach to predicting churn. Deep Learning models tend to go one step further to predict churn rates.

1. IMPLEMENTATION

*A. Data cleaning*

On checking for duplicate and missing values, we found the data accurate and consistent. There were no missing values.

*B. Exploratory Data Analysis*

Few fields were in a different format. We transformed the feature into the numerical format by filling empty spaces(NaN) with a median. Every feature was plotted against the churn rate to get an idea of what the feature is and how it impacts the overall churn rate of an individual.

A function named “exploring\_addtional\_services” was made that takes features as input, converts data into a percentage, and plots it.

*C. Data Preprocessing*

Machine learning models work only with numerical values. Thus some relevant features were transformed into numerical ones. Binary encoding is used after transformation.

*D. Correlation*

A correlation matrix was used as a measure to find a linear relationship between 2 variables. Some of the features were highly correlated, and thus they can be dropped from the dataset, for example, DeviceProtection, TechSupport, StreamingTV, and StreamingMovies.

*E. Generalized Linear Model*

It is a regression model that describes relations between predictor variables and a response variable. GLMs allow for more flexible, non-linear relationships by employing a diverse underlying statistical distribution.

GLM outputs the following measures: estimated coefficients, standard errors, and p values.

Estimated coefficients represent the estimated effects of the predictor variables on the response variable.

Standard Errors represent the standard deviation of the sampling distribution of the estimated coefficients.

P values are used to determine the statistical significance. A small p-value (e.g., less than 0.05) suggests that the estimated coefficient is unlikely to have arisen by chance.

*F. Feature Scaling*

Feature Scaling is a technique to standardize the independent features present in the data in a fixed range.

The columns tenure, monthly charges, and total charges are scaled individually and then pushed back to the original database.

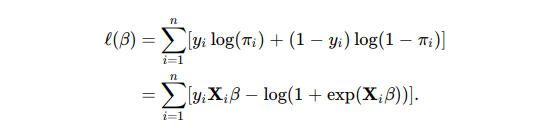
*G. Classification Models*

For the three models we have used, the approaches are as follows:

1. *Binary Logistic Regression:* It is a classification algorithm for classifying whether the customer will churn. Here we have made some assumptions as follows:

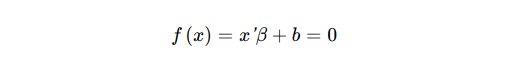
* The dependent variable is binary that is, it has only two options, 0 or 1.
* It should have no or very little multicollinearity between the independent variables. The independent variables should be linearly related to log odds (ratio of success to failure).
* Logistic regression requires a fairly large sample size to make it more reliable.

The log-likelihood for the logistic regression is as follows:

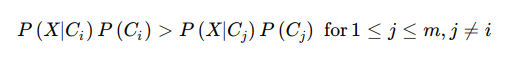
****

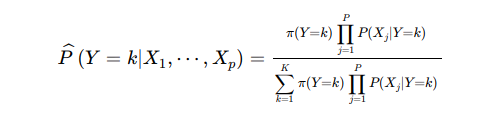
[](https://www.codecogs.com/eqnedit.php?latex=%20%5Cpi%20%3D%20#0) success probability

1. *Support Vector Machine (SVM):* The support vector classifier finds the largest gap possible between the nearest points of two clusters.

****

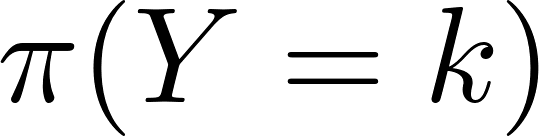
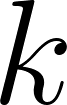
1. *Naive Bayes Classifier:* It classifies using the conditional probability approach to determine the probability of the hypothesis given some prior knowledge. We have used Gaussian Distribution for code implementation.

****

****

Y is the random variable corresponding to the churn class index of an observation.

X1, X2, and X3 are the predictors of an observation.

[](https://www.codecogs.com/eqnedit.php?latex=%5Cpi(%20Y%3Dk%20)#0) is the prior probability that a class index is [](https://www.codecogs.com/eqnedit.php?latex=%20k%20#0)

For the same, code implementation involved the following libraries:

* Sklearn library that supports mathematical, statistical, and general machine learning algorithms (imported algorithms from this library)
* Numpy for linear algebra
* Pandas for data processing
* Matplotlib, Plotly, and seaborn for visualization

We created one baseline model function and then used the same for three models that prints accuracy, precision, recall, and f1score along with the confusion matrix.

We try to fit training data to train the model and predict it using the testing data for each model.

model.fit(X\_train,y\_train)

y\_pred = model.predict(X\_test)

The implementation can be found here: <https://colab.research.google.com/drive/1yQ5No-dFUdl9hPT6nHA-ajK4If53K1id?usp=sharing#scrollTo=AxIXOx1BWUrH>

1. RESULTS

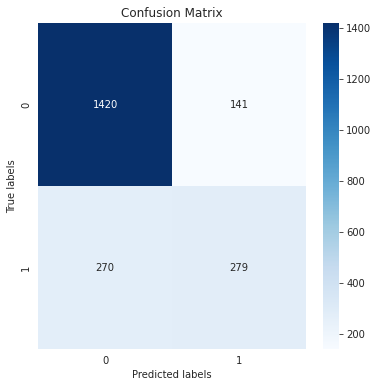
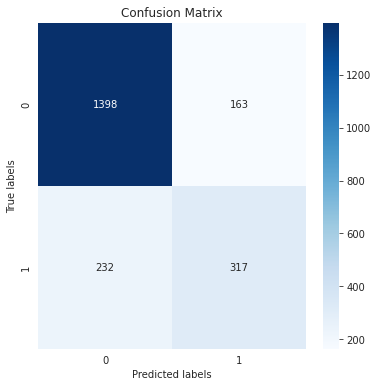
The models are evaluated based on the following factors:

* Accuracy: measures the proportion of correct predictions made by the model
* Precision: measures the accuracy of positive predictions made by the model
* Recall: refers to the model's capacity to detect positive instances accurately
* F1 Score: harmonic mean of precision and recall

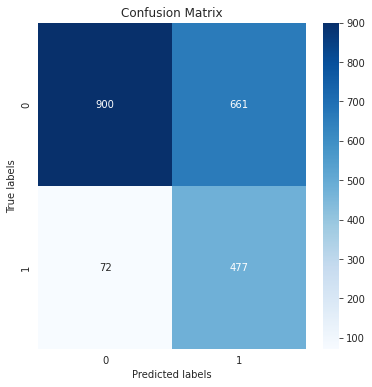
|  | Logistic Regression  (%) | Support Vector Classifier (%) | Naive Bayes Classifier (%) |
| --- | --- | --- | --- |
| Accuracy | 81.27 | 80.52 | 65.26 |
| Precision | 66.04 | 66.42 | 41.91 |
| Recall | 57.74 | 50.81 | 86.88 |
| F1 Score | 80.85 | 79.61 | 67.28 |

For the three models, confusion matrices were created.

The parameters for the same are output tests and output predicted(i.e., Model name, y\_test, and y\_pred)



Logistic regression SVC



Naive Bayes

From confusion matrix output, and we get correct and incorrect predictions.

1. CONCLUSION

We see varied results across the algorithms. Each has its own merits and demerits.

Looking at the confusion matrices, the values of incorrect predictions were higher for Naive Bayes and less for SVM. The correct prediction was higher for logistic regression and less for Naive Bayes

Logistic Regression proved to be the most accurate, while Support Vector Classification proved to be the most precise. Support Vector Classification and Logistic Regression are very similar in their performance metrics. Naive Bayes Classification renders generally lower performance metrics and thus becomes the least ideal choice.

1. REFERENCES

[1]“12.1 - Logistic Regression | STAT 462,” *12.1 - Logistic Regression | STAT 462*. [Online]. Available: <https://online.stat.psu.edu/stat462/node/207/>

[1]S. R. Publishing, “Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms,” *Churn Prediction Using Machine Learning and Recommendations Plans for Telecoms*, Nov. 05, 2019. [Online]. Available: <https://www.scirp.org/html/3-1731142_96177.htm>

[1]“Naive Bayes Classifier in Machine Learning - Javatpoint,” *www.javatpoint.com*. [Online]. Available: <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>

Ahmad, A. K., Jafar, A., & Aljoumaa, K. (2019, March 20). Customer churn prediction in telecom using machine learning in big data platform. Journal of Big Data, 6(1). <https://doi.org/10.1186/s40537-019-0191-6>