Customer Retention Analysis for Music Streaming Services

Manasi Khandekar
UNI: mk4679
mk4679@columbia.edu

Aishwarya Sen
UNI: as6718
as6718@columbia.edu

Ajinkeya Chitrey
UNI: ac5166
ac5166@columbia.edu

Abstract

Customer retention analysis or churn analysis is the analysis of a company or service's consumer loss rate with the aim to minimize it. This is of utmost importance in today's fiercely competitive market and our project aims to address the factors that influence churn for music streaming services with the objective of helping such services take precautionary measures to retain their consumers. With the aid of big data tools such as PySpark and MLlib we have created an end-to-end pipeline to analyse factors and carry out classification, the output of which is an interactive application which predicts if a consumer is likely to churn from a music service or not. We have used a variety of popular machine learning models, among which Random Forest achieves the best results with a validation score of 89% and a high train set score of 94.6%.

1. Introduction

The rise in popularity of online streaming services has been accompanied by market saturation by the sheer large number of such platforms. There are already signs of saturation in signing of new customers. The consumers have a finite amount of time and budget to spend on these streaming services, particularly as they spend less time at home in the post pandemic environment. Customers face dilemma while choosing between different services and this is what makes churn or customer retention analysis important in today's scenario. Businesses now need to track their customer metrics, analyse them and convince their customers to continue on their platforms. Churn analytics provides estimates of the rates at which customers discontinue the usage of a service and also answer important questions like who, why, when and where, i.e. who discontinued the service, why they did so, when they did it and where they did it from. These services can then use tactics like offering higher personalization and incentives to retain customers.

Our project aims to study and understand the factors that affects this trend for music streaming services. With the help of several big data technologies such as Pyspark and MLLib we aim at conducting a thorough analysis and providing substantial results that can provide companies with the knowledge of factors affecting churn for the services they provide. A comprehensive analysis was carried out in which exploratory data analysis techniques were used to understand the underlying pattern and feature relations, data prepocessing and feature engineering techniques were used to process the data followed by model building. Several classifiers including Logistic Regression, Naive Bayes, Linear SVC and Random Forest were trained and tested on the data, with Random Forest Classifier achieving the highest weighted F1 Score of 89% and a train score of 94.6% post hyper parameter tuning. Thorough experimentation was carried out in order to use different combinations of features and processes to order the best generalized and most accurate models. Finally, we used streamlit to create an interactive application where the user can input permutations of different features to observe their impact on the consumer i.e. if the consumer churns or stays with the service.

2. Related Works

Customer churn prediction is an area of significant interest for various industries. Even minor improvements can give positive results in terms of business revenues due to which studying churn predictions has been a relevant topic. It has been used in banking and telecommunications sectors the most, but in recent years it has been applied to the OTT over the top streaming services as well. This includes the music streaming services too. A variety of techniques and classifiers have been implemented in the previous research papers. Novel models and techniques are emerging all the time and optimization of algorithms is furthering the data science work in customer churn prediction (Santharam and Krishnan, 2018)[3].

A study comparing different machine learning algorithms was done by Vafeiadis et al. (2015)[8] for customer churn prediction on telecommunication data. It measured the model performances of Support Vector

Machines, Naïve Bayes classifiers, Artificial Neural Networks, Decision Trees, and Multi-Layer Artificial Neural Networks based on precision, recall, accuracy, and the F-measure scores. They found that the Decision Tree Classifier reached an accuracy of almost 0.94 and that model boosting helped improve the model performance slightly. Along similar lines Khodabandehlou and Zivari Rahman (2017)[2] compared Artificial Neural Networks, Support Vector Machines, and Decision Trees but on retail industry data. The came to the conclusion that a multi-layer perceptron performs the best with an accuracy of 0.91.

In 2017, Nimmagadda and Subramaniam compared logistic regression, artificial neural networks and gradient boosting methods using log loss as metric. They found that gradient boosting performed the best [7]. Another study in 2019 by Zhou et al.[9] proposed a novel model based on Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) called densely-connected LSTMCNN (DLCNN). Comparison of these models based on the metric, area under the curve (AUC) gave the following results- DLCNN had highest AUC of 0.8703, LSTM was next with an AUC of 0.8652 then CNN with an AUC of 0.8514 and XGBoost with AUC of 0.8514.

There are two crucial requirements for using churn prediction insights well: high classification performance and a high level of model interpretability. In the business sense of things, explainability of models is quite important for efficient decision making. Based on these 2 factors, different models can be compared. Logistic Regression coefficients are less intuitive than, for example, linear regression (Fernandes et al., 2020)[1]. A drawback of logistic regression is the assumption of linear relationship in the churn dataset which might not always be true (Molnar, 2021)[5]. Being a non linear model, decision tree addresses the shortcomings of logistic regression. Decision trees are easy to interpret visually, by merely looking at the root node and the edges of the tree forms further subsets. Random forest and gradient boosting algorithms are not quite interpretable since they are almost as if black box models. It is still possible to estimate the importance of different features using powerful libraries such as scikit-learn from Python (Revert, 2019)[6].

In 2021, Mantas Matusevičius [4] compared the 4 models of logistic regression, decision trees, random forest and XGBoost. In terms of accuracy, XGBoost and Ransom Forest model were superior, but results of logistic regression and decision trees models were lucid and easily explainable. Also the data used was of just one month so a conclusion was that using data from a longer time duration will improve the classifier performance.

The previous research that we came across has been over different kinds of datasets like telecom sector, retail sector, OTT entertainment services. The methods therefore used previously can be tried out on the dataset that we have.Taking into consideration comparison of various machine learning models based on metrics of accuracy and interpretability, we decided to try out different models on our dataset and select the one with the best accuracy. The models we decided to use were, Logistic Regression, Random Forest, Gradient Boosting, Linear SVC and Naive Bayes.

3. Data Description

One of the defining criterias of any dataset used in big data is that it follows the three V's - volume, velocity and variety. In this project we have obtained a dataset via Udacity which was collected in such a way that it satisifes these three major criteria. The data used was consolidated from multiple streaming service APIs over a period of 2 months - October 1, 2018, to November 30, 2018. It contains a total of 26 million logs or rows and consists of information for 22278 users. The streamed data contained event-level data to track the usage patterns of the services by customers as well as customer-level demographic information to study how usage patterns differ for different segments of customers. The total data accumulated was over 12 GB and had 18 columns common to all service APIs. Customer-level data included features such as customer gender, country, sex, etc. Activity log data included features such as the last song played, current subscription level, etc.

The dataset columns are as follows:

1. artist: string - artist name
2. auth: string - authentification method
3. firstName: string - user first name
4. gender: string - user gender
5. itemInSession: long - length of user session
6. lastName: string - user last name
7. length: double - length of the song listened
8. level: string - sparkifiy user service level (paid or free)
9. location: string - location of the user
10. method: string - http service method
11. page: - user service interaction event
12. registration: long - timestamp of user service registration
13. sessionId: service session id

14. song: song name played by the user

15. status: - http status

16. ts: long - timestamp of user service event

17. userAgent: - web browser used

18. userId: string - unique userid

The various preprocessing and feature engineering techniques used by us in order to obtain a more clean and reliable version of the dataset has been discussed in future sections.

4. Methodology

A systematic methodology has been followed to carry out an extensive analysis of the influence of various relevant factors on customer churn as well as to create several efficient classification models to categorize the users into two categories - churn or not churn.
Below, the sectioning of the methodology is briefly described:

1. **System Overview**:
   The software and primary techstack used by us has been described along with their applications in our project. Additionally, the bottlenecks with respect to the software has been described along with possible counter-measures.

2. **Exploratory Data Analysis**:
   A comprehensive analysis using several data visualization techniques on Tableau to understand the impact and correlation of the various features on customer churn, has been carried out. The analysis has been presented in the form of an interactive and interpretable dashboard.

3. **Data Processing**:
   This section details the various data preprocessing and feature engineering steps carried out in order to obtain a more reliable and clean dataset. These include data preprocessing steps such as dealing with null values, missing value imputation, correction of date formatting and dropping of redundant rows and columns. Feature engineering steps include creation of 'churn' target variable column along with various other significant features modeling user engagement and activity that are essential for classification.

4. **Modeling**:
   This section provides an elaborate discussion on the various models that have been used for classification as well as the various experiments carried out to increase the accuracy and efficiency of the models. Various experiments include one hot encoding of categorical features, normalization of features and hyperparameter tuning.

5. **Model Deployment**:
   The best model is finally deployed on an interactive, user-friendly frontend application using Streamlit, where the user can input various combinations of feature values to predict if the customer will churn or not.

4.1. System Overview

First, an overview of the softwares and techstacks used by us for the various stages in our pipeline is provided, followed by the system constraints or bottlenecks faced by us and suggested measures that can overcome them, and finally the steps taken by us to use these technologies to build our project.

4.1.1 Software and Techstack

1. **Google Cloud Storage**:
   Google Cloud Storage is a service to store objects to remote servers. These objects can be data in any file format like csv, json, parquet etc. This online file storage web service hosted by Google Cloud Platform (GCP) has been utilized by us for storage as well as interaction with our large dataset, using other GCP services. Its high performance, scalability and security as well as sharing capabilities were the reasons for us to opt for GCS.

2. **Tableau**:
   Tableau is a data visualization and business intelligence tool which helps the user gain insights from the data he/she has and make it actionable. Tableau's user-friendly interface for exploring, managing and visualizing data for obtaining useful insights rendered it the top choice for us to carry out data analysis and visualization for our project.

3. **Google Cloud Dataproc**:
   Google Cloud Dataproc is a service to run Hadoop and Spark jobs for batch processing, querying, streaming, and machine learning. Google Cloud Dataproc allows us to integrate the usage of cloud-scale databases, open-source tools, programming languages and other softwares for us to carry out the various stages of data processing and modeling in our project.

4. **Apache Spark**:
   Processing tasks on extremeley large datasets can be

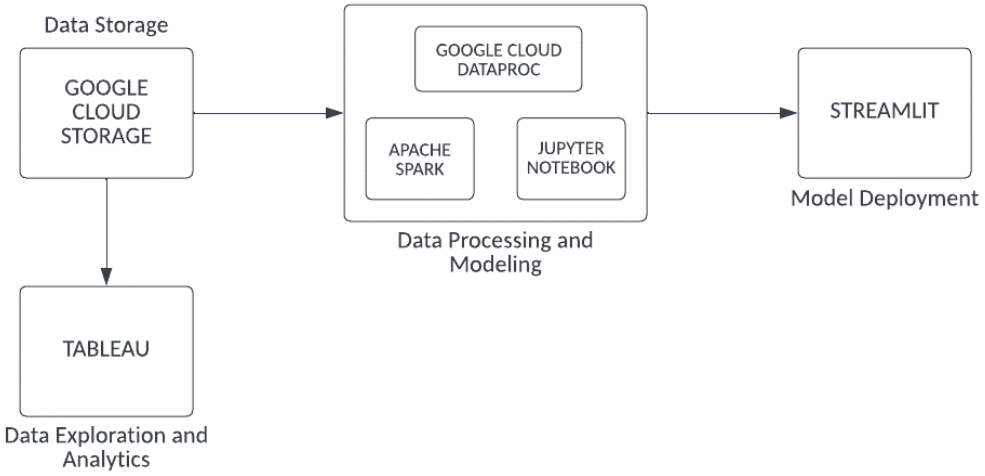

Figure 1: System Design

performed quickly using Apache Spark data processing framework. Apache Spark provides a large-scale data processing and analytic's engine which allows data parallelism and assists us in the processing of our vast dataset and associated models. Spark has an easy to use API which simplifies the work to a great extent.

5. **Jupyter Notebook**:
   Jupyter Notebook is a web based interactive platform for computation which gives a simple, streamlined experience. Jupyter Notebook, hosted on the dataproc is used by us as the primary platform for the creation and running of our models and codes.

6. **Streamlit**:
   This open-source python framework is used for quick deployment of machine learning based web applications and is utilized by us to create an interactive frontend application for our project.

4.1.2 System Constraints

1. The current system architecture is not equipped to handle live streaming data. So an open future scope item where we aim to continuously improve the deployed models by feeding it live streaming api sessions data will require including a component such as Cloud Dataflow.

2. Due to cost restraints, our current dataproc cluster configuration is not very computationally optimized. So to handle larger datasets or tons of streaming data in real time will require a computationally more powerful infrastructure.

4.1.3 System Design

Figure 1 illustrates the system design or the pipeline that the customer retention analysis model follows. Below, our system design is briefly described.

1. **Data Storage:**
   First, the consolidated dataset was obtained and uploaded to bucket on Google Cloud Storage (GCS) which provides a platform for easy storage and access of large scale databases as well as for rapid querying by other GCS Services. The data is stored as a json file.

2. **Data Exploration and Analysis:**
   Next, this data was utilized to create dashboards on Tableau, which enables the observation of various features and patterns in the vast dataset.

3. **Data Processing and Modeling:**
   Tools such as Google Cloud Dataproc, Apache Spark and Jupyter Notebook were utilized. Comprehensive data cleaning, feature engineering and model training, tuning of parameters and testing was carried out. The model with the best generalized performance was saved to the google cloud bucket.

4. **Model Deployment:**
   Finally, the best model - the model which obtains the most robust scores would be deployed on streamlit to create the frontend.

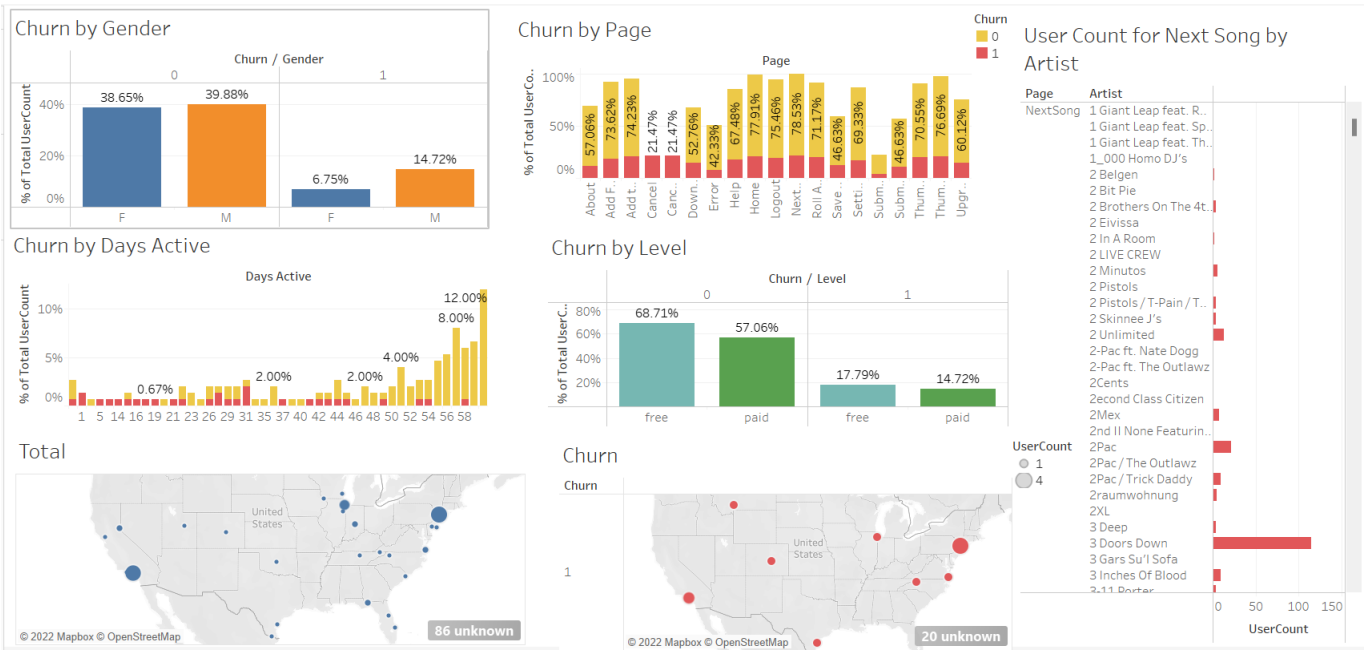

Figure 2: Exploratory Data Analysis: Tableau Dashboard

4.2. Exploratory Data Analysis

In order to explore the vast data and analyse the underlying trends and relations between its various features, exploratory data analysis was carried out on Tableau.

Under exploratory data analysis, a few important features were considered and their impact on the customer churn was evaluated.

1. The first analysis carried out was the impact of gender on customer churn. It was observed that while there is no significant difference in the number of males to number of females who have stayed with the service, a much larger percentage of males have churned or left the service, as compared to females.

2. Analysis of a few factors related to consumer engagement like the most frequently-visited pages by customers and the number of days the user was active on the service, helped gain useful insights such as that there is a lower rate of churn among users who have used the service for a longer period as compared to users who have recently started availing the service.

3. Analysis as per page i.e the page on the app that the user has visited during their session, enabled the observation of the engagement and browsing patterns followed by users that have churned.

4. As per level i.e. if the user was paying or using a free version, it was observed that as compared to users who have not churned, there is a smaller difference between paid and unpaid churned users indicating that the paid users have not churned as often.

5. User count for next song gives a useful insight which helps the service analyse which artist is unpopular among its customers.

6. And finally by location the company can analyse which cities or countries their service is popular or unpopular in and can take measures accordingly.

These analyses can thus help companies change their policies so as to retain as well as attract new customers.

4.3. Data Processing

In order to obtain a more accurate and reliable version of the dataset through steps such as removal of inconsistent or missing values as well as creation of more pertinent features, we implement certain data processing methods.

A Dataproc cluster was created for further data preprocessing and modeling.

The major steps undertaken for data cleaning and feature engineering were as follows:

4.3.1 Data Preprocessing

Some of the most significant data preprocessing steps performed were as follows:

userId	gender	churn	last_level	days_active	last_state	avg_songs	avg_events	thumbs_up	thumbs_down	addfriend
10	M	0	paid	42	MS	84.13	99.38	37	4	12
100	M	0	paid	59	TX	81.27	97.39	148	27	49
100001	F	1	free	1	FL	66.5	93.5	8	2	2
100002	F	0	paid	56	CA	39.0	43.6	5	0	1
100003	F	1	free	2	FL	25.5	39.0	3	0	0
100004	F	0	paid	57	NY	49.58	65.53	35	11	19
100005	M	1	free	18	LA	38.5	54.0	7	3	3
100006	F	1	free	0	MI	26.0	44.0	2	2	4
100007	F	1	paid	58	AR	47.0	57.78	19	6	17
100008	F	0	free	49	CA	96.5	117.5	37	6	17

Figure 3: Engineered Dataset

- Any row with a missing user id was dropped, as there was no way to associate the activity log to a user, making the analysis redundant.
- Any columns with more than 50% missing values was dropped.
- Median value imputation for numerical columns and mode value imputation for categorical columns with nulls was performed.
- Date formatting was corrected.

4.3.2 Feature Engineering

In order to leverage the data to create a more comprehensive and relevant dataset for our problem statement, we carry out several feature engineering techniques. To predict for each user whether the user is going to churn, the following features were extracted:

- Gender, because EDA showed that male and female users might have different behaviors
- Last Level (paid or free), because it's observed that paid level users are more active and also may have different behavior
- Target column "Churn" was created; Value 1 was assigned for all users who have canceled their subscription at some point, otherwise 0.
- Average time a user spends on the service each day was modeled.
- Average events per day to see how active the user is in general
- Average songs per day to see how active the user is in listening to the music
- Number of thumbs up to see how satisfied the user is with the content of the service
- Number of thumbs down to see how satisfied the user is with the content of the service
- Days since the date of the first event to see how long the user has already been using the service
- Last location (state), the popularity may vary depending on advertising in different regions
- Number of add friend events, perhaps the user is less likely to churn if his friends subscribe to the service

The processed dataset was then stored back in the GCS bucket. A new notebook in the dataproc cluster pulled this processed dataset back for modeling.

4.4. Modeling

The engineered features are now used for the creation of classification models using common machine learning models including Random Forest, Gradient Boosted Trees and Naive Bayes. Prior to model creation and parameter tuning, the features are encoded to the form needed for the models, along with the creation of train and test sets using train test splitting. These are elaborated in the following sections describing the major experimentation we have carried out.

4.4.1 Experimentation

To carry out efficient modeling of the multiple features in order to obtain accurate classification models, various experiments were carried out. These are enumerated below.

Experiment 1:

The major steps involved in the first experimentation or modeling process were as follows:

- First, a one hot encoded version of the categorical features (gender, last_level and last_state) was created.
- Then, a unified vector representation of all features using PySpark's VectorAssembler was created.

	Classifier	Train	Test
0	Random Forest	0.836870	0.813589
1	Logistic Regression	0.637836	0.728448
2	Gradient-boosted Tree	0.985075	0.770833
3	Naive Bayes	0.725460	0.776261

Figure 4: Experiment 1 Results

- Split data into training, validation and testing data (60/20/20)
- Experimentation were carried out with multiple models such as Random Forest, Logistic Regression, Gradient Boosted Trees, Naive Bayes, and Support Vector Machines with their default hyperparameters.
- Performance evaluation was done using the weighted F1 score, as the output labels were imbalanced.
- Based on the scores obtained, Random Forest proved to give the best-generalized performance.

Experiment 2:

For further experimentation, multiple data augmentations were done such as:

- Any state value that is not in the top 10 most frequent states, was replaced by "Other", as the frequency of lower ranking states is extremely skewed and introduces bias in the model.
- The numerical features were normalized in order to balance the data scales.
- All columns which were modeled as an average of an original data column such as avg_events and avg_songs were bucketized.
- Random Forest, Logistic Regression, Gradient Boosted Trees, Naive Bayes, and Support Vector Machines were trained on the augmented dataset and evaluated using weighted F1 score.

Experiment 3:

The best generalized performance is displayed by Random Forest and hence it's hyperparameters are tuned using cross-validation:

- Various hyperparameters combinations are experimented with.
  - numTrees [10, 50, 100]
  - maxDepth [2, 3, 5]
  - impurity ['entropy', 'gini']
  - featureSubsetStrategy ['auto','sqrt', 'log2']
- The best cross validation score is obtained for parameter combinations: numTrees=10, maxDepth = 5, impurity = 'entropy' and featureSubsetStrategy = 'sqrt'
- For the tuned Random Forest model, the training set weighted F1 score is 94.6% and validation set weighted F1 score of 89%

4.5. Model Deployment

To create an interactive frontend and deploy the application, we leveraged streamlit and its cloud service. Streamlit is easy to use, user friendly and allows quick app deployment with minimal effort required.
We created a community cloud account for free on streamlit and then created a new app using the github repository that we had already created.
The github repository contains the following:

- The saved model in parquet format, since pyspark models get saved that way.
- A requirements.txt file which has all the libraries needed for the app to run with their configuration mentioned if need be.
- A packages.txt file to include in order to run the app.
- The main file is the app.py which has the code to run the app.

The app.py file has the code which gives the user 2 options to enter user activity parameters. One option is to upload a

	Classifier	Train	Test
0	Random Forest	0.935098	0.778850
1	Logistic Regression	0.658448	0.733777
2	Gradient-boosted Tree	0.992264	0.712958
3	Naive Bayes	0.658448	0.733777

Figure 5: Experiment 2 Results

	f1	numTrees	maxDepth	impurity	featureSubsetStrategy
0	0.744618	10	2	entropy	auto
1	0.744618	10	2	entropy	sqrt
2	0.744618	10	2	entropy	log2
3	0.768257	10	2	gini	auto
4	0.768257	10	2	gini	sqrt
5	0.768257	10	2	gini	log2
6	0.803274	10	3	entropy	auto
7	0.803274	10	3	entropy	sqrt
8	0.803274	10	3	entropy	log2
9	0.774444	10	3	gini	auto
10	0.774444	10	3	gini	sqrt
11	0.774444	10	3	gini	log2
12	0.820868	10	5	entropy	auto
13	0.820868	10	5	entropy	sqrt
14	0.820868	10	5	entropy	log2

Figure 6: Experiment 3 - Hyperparameter tuning

csv file and then clicking on predict, get a result displayed on screen. The other is to enter the parameters manually to get the prediction. The result is displayed as text whether a user with the input activity parameters will churn or not. The app.py file has 2 functions defined:

- The create_features function takes the input data, processes it and returns features created from it.
- The trained_model function takes as input features, loads the trained model and then predicts for the given input.

5. Results

We tried out different models on our preprocessed dataset after feature engineering. The results can be seen in Figure 4. The Gradient boosted tree gave a good weighted F1 score on the test data, but it was overfitting heavily. Naive Bayes and Logistic Regression models both had low weighted F1 scores. The Random Forest Model gave the best generalized performance. From the results of Experiment 2, we can see that the same trend followed. The Random Forest model again gave the best score of 0.778850 on the test dataset. So we finalized the Random Forest model and tuned its hyperparameters in the third experiment. After training the model using the parameters obtained by cross validation, we got an improved accuracy of 94% on training data and 89% on the test data. We saved this model and used it in our frontend application deployed using streamlit.

6. Business Value and Future Scope

The business value as well as future scope for this project is extensive and has been briefly described below.

6.1. Business Value

1. This project provides a comprehensive survey of the importance of various features on customer churn for music streaming services and can thus be used to help such businesses track engagement and browsing patterns of their customer base. This will in turn help the business stakeholders make their product market fit

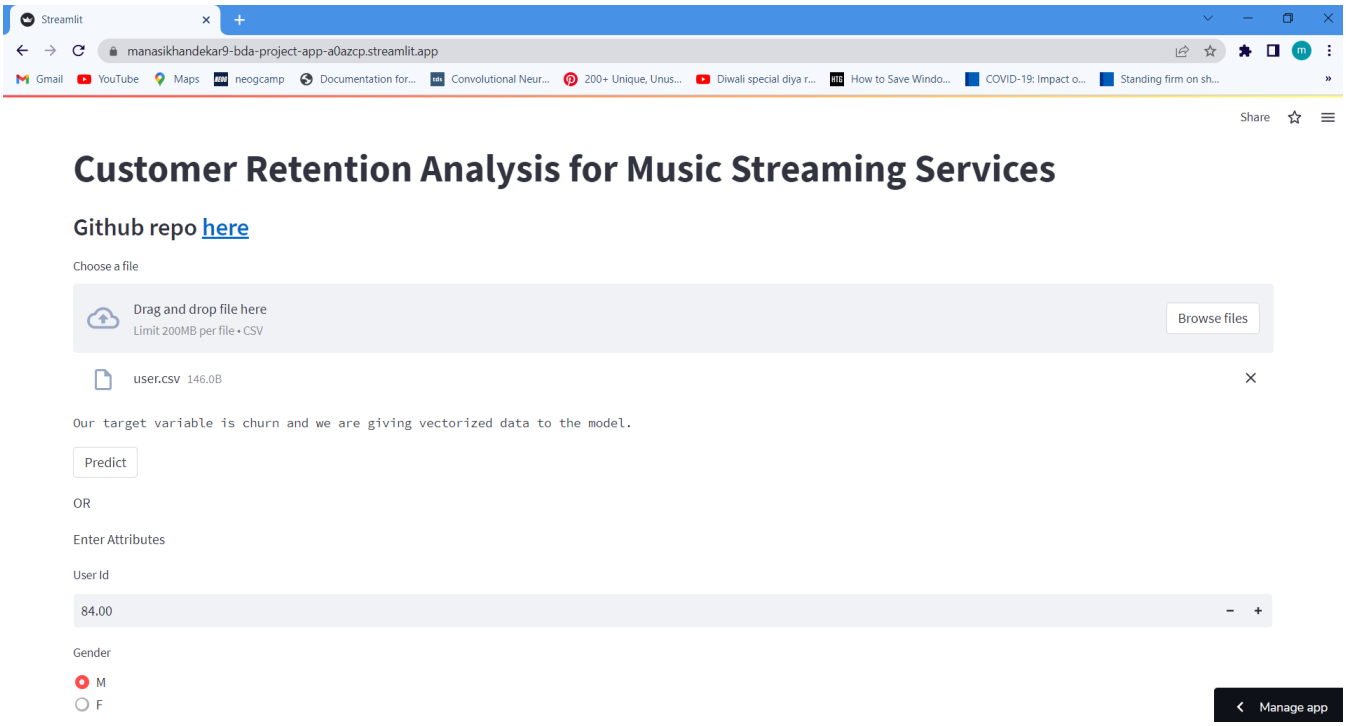

Figure 7: Streamlit App

and make calculations of the actions that they should take in the next quarter.

2. The models and interactive application can help identify if a customer is about to churn and the service can then take corrective measures accordingly by providing certain incentives to such customers.

3. The project can also be used to identify areas of improvement in the business based on the customer usage and churn patterns. Visualization of the data using the interactive dashboard can provide the service with data on a variety of factors such as which locations their service is doing well in or under performing, which artists are popular or unpopular on their application, etc.

   Thus, the business value is vast and this application can be utilized by streaming services to avail a variety of benefits which can help them expand their revenue.

6.2. Future Scope of Work

1. We plan on carrying out several experiments with further feature engineering and data enrichment techniques in order to increase the accuracy of our classification models.

2. Further, we aim to experiment with different ensemble learning methods like boosting and bagging to improve the prediction accuracy by a combination of different models.

3. We also intend on implementing deep learning techniques to this problem statement, using models such as Deep Neural Networks for classification.

4. Additionally, we wish to integrate live streaming data to enhance the results of our models.

7. Conclusion

In today's competitive marketplace, customer retention is one of the biggest challenges for any service. The presence of an abundance of choice for any consumer has a positive impact on the performance of competing companies as they have to constantly update and refine their service so as to retain their customers. Music streaming services too face similar circumstances. Customer churn analytics is an important metric which helps services adjudge why they lose customers and helps them take precautionary measures to avoid this in the future. Our study helps us analyse the factors affecting churn in music streaming services and how the usage of effective machine learning models can helps us predict if a certain customer will churn or not. The project has enabled us to explore and gain knowledge on the usage of various big data tools and technologies and their application and relevance in solving of current and real world business problems.

References

[1] Figueiredo Filho D. B. Rocha E. C. da Nascimento W. da S. Fernandes, A. A. T. Read this paper if you want to learn logistic regression. 2020. 2

[2] Zivari Rahman M. Khodabandehlou, S. Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior. 2017. 2

[3] Santharam Krishnan. Survey on customer churn prediction techniques, 2018. 1

[4] Mantas Matusevičius. User churn prediction within music streaming service industry: A comparison of machine learning models. 2021. 2

[5] C. Molnar. Decision tree — interpretable machine learning. 2021. 2

[6] F. Revert. Interpreting random forest and other black box models like xgboost. 2019. 2

[7] Man Long Wong Sravya Nimmagadda, Akshay Subramaniam. Churn prediction of subscription user for a music streaming service., 2017. 2

[8] Diamantaras K. I. Sarigiannidis G. Chatzisavvas K. Ch. Vafeiadis, T. A comparison of machine learning techniques for customer churn prediction. 2015. 1

[9] Yan J. Yang L. Wang M. Xia P. Zhou, J. Customer churn prediction model based on lstm and cnn in music streaming. 2019. 2

8. Project Contribution

All members - Ajinkeya Chitrey, Aishwarya Sen and Manasi Khandekar have contributed equally to the project.