Humana-Mays Health Analytics 2020 Case Competition

Content Table

Executive summary

Research conducted by AHA noted:

- 3.6 million individuals cannot access medical care because of the transportation barriers.
- 4% of all children miss their medical appointment caused by transportation issues.
- Transportation barriers are the third leading cause of missing a medical appointment for seniors across the country.

The project's goal is to develop a model to predict which Medicare members will potentially have transportation challenges. The ROC_AUC paradigm will value these probabilities. This study will provide Humana a better understanding of the members' needs for transportation and help Humana bring the care to the customers promptly.

The metric for evaluating the models is AUC. We tried to lower the False Negative Rate because we would rather have a higher False Positive Rate in this scenario. First, it is an unbalanced data. We did not want the model to predict everything to be 0. Second, we believed Humana would rather be well prepared than the lack of preparation when it comes to bringing timely care to the customers.

The models we tried for the project are XGBoost, LightGBM, Decision Tree & Random Forest, CatBoost. The result comparison will be shown and fully explained in the Modeling section. After testing the different models and doing hyperparameter tuning on these models, the optimal model we got is LightGBM. Here is a snapshot (Figure 1.1) illustrating what our model looks like:

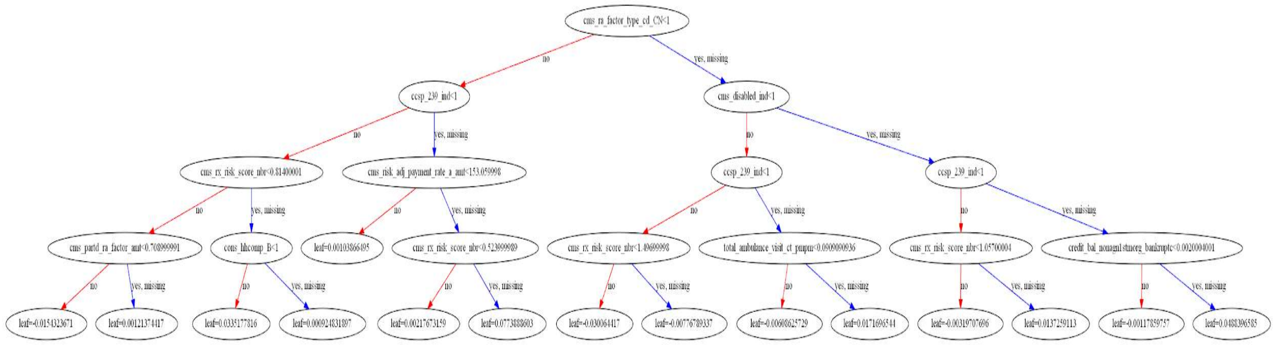

Figure 1.1: Partial Plot of Our Model

Using the prediction model, Humana will be able to target the member groups who have potential transportation problems and provide several plans to help solve the member's transportation issues. With better service, Humana might help the members get timely treatment, stay healthy, and attract more customers to their business.

Data Preparation

Overview

Humana's original datasets are divided into two parts: a training set and a holdout set. Thereinto, the training set includes 69572 recordings from Humana's members and 826 variables, and the holdout data has 17681 members' records and 825 variables. We sorted out the data and found the primary sources coming from the following aspects:

1. Basic information
2. Health issue
3. Credit situation
4. Social consumption
5. Medicare and drug

And our target is to predict the variable of *transportation_issues* in the holdout dataset given the tremendous variables we already have.

The definition of the target variable, “transporation_issues,” is provided as “At a high level, some members have difficulty with transportation and seeking care for a variety of reasons.” In the training set, the target variable is whether a member had a transportation difficulty (0= No, 1=Yes) in 2019. We need to predict that result in the holdout dataset based on members’ various records and information. Though it is a self-reported value, we can still observe its connection with other variables.

In the training set, approximately 85.34% of members claimed that they did not have a transportation challenge in the past year. Only 14.66% of members expressed that they had trouble with transportation last year, so we can see that the target result is imbalanced.

Our team also compared the essential information in the two datasets. According to the training set shows, there are 59.1% female members, 91.9% native speakers, and the top3 members come from FL, LA, and TN, compared with 58.6% female members, 91.4% native speakers, and top3 states are FL, TN, and LA in the holdout dataset. They have the same age distribution as well (Figure 2.1). We then concluded that the basic information in the two datasets is similar and comparable.

Figure 2.1: Age distribution in the training and holdout dataset

Data Exploration

After the preliminary analysis, we started to explore and manipulate the data. Among the variables, we found that 165 of them originally were categorical variables in both two datasets. 661 variables in the training dataset and 660 variables in the holdout dataset (*transportation_issues* excluded) are numeric.

There are 131 variables in the training set with missing values. According to Figure 2.2, three of them miss more than 70% of values. they are: *hedis_ami*, *hedis_cmc_ldc_c_control and hedis_cmc_ldc_c_screen*, which all are categorized into the aspect of "Medicare and drug". There are 95 variables in the training dataset and 99 variables in the holdout dataset with one unique value.

	percent_missing
hedis_ami	99.665095
hedis_cmc_ldc_c_control	78.957052
hedis_cmc_ldc_c_screen	78.954177
cons_homstat	27.712298
cons_ret_y	27.710861
...	...
submcc_ano_dig_pmpm_ct	0.000000
submcc_ano_gu_pmpm_ct	0.000000
submcc_ano_hrt_pmpm_ct	0.000000
submcc_ano_mus_pmpm_ct	0.000000
submcc_rsk_chol_ind	0.000000

826 rows × 1 columns

Figure 2.2: Missing Value Percentage for Training Data

Data Processing & Transformation

- **Get Dummy Variables**

After exploring the data we have on hand, we realized that many categorical variables would not be read by the classification algorithms we applied. Thus, we utilized two different methods (One-hot Encoding and Label Encoding) to transform the categorical variables.

After several rounds of experiments, we decided to convert all the categorical variables to dummy variables. In this way, we can capture the information as much as possible while not adding any unnecessary information to the models.

An example of before and after data transformation is demonstrated as Figure 2.3:

Before			After			
ID	Attribute1	→	ID	Attribute1_EM	Attribute2_AM	Attribute3_CM
1001	EM		1001	1	0	0
1002	AM		1002	0	1	0
1003	CM		1003	0	0	1

Figure 2.3: Transform Categorical Variables to Dummy Variables

- **Handle Missing Values**

As we discovered above that there are many missing values in the numeric variables, we initially filled all the missing values with "–9999" because we believed that these extreme values would be automatically screened out in the process of pursuing the extremum in the model training. Then we also tested to fill the missing data with the average value of the variable. The outcome of both methods is very close. We decided to use the average value to replace all the missing data since it is the most used method, and it will do the least damage in the models' training process.

- **Feature Engineering**

To better understand the data and extract the information for analysis, we decided to create the following features from the existing data:

Variable Names	Description
Total_betos_pmpm	It represents the total of the logical claims for all the betos codes filed per member per month
Betos_pmpm_1	It is a binary category. It will be marked as 1 if the number of Total_betos_pmpm is smaller or equal to 1; otherwise, 0. It represents the first 0.3 percentile of Total_betos_pmpm.
Betos_pmpm_13	It is a binary category. It will be marked as 1 if the number of Total_betos_pmpm is larger than 1 and smaller than 3; otherwise, 0. It covers the percentile from the 0.3 to 0.8 of Total_betos_pmpm.
Betos_pmpm_above3	It is a binary category. It will be marked as 1 if the number of Total_betos_pmpm is larger than or equal to 3; otherwise, 0. It covers the top 20 percent of Total_betos_pmpm.
mapd_amt	The total payment made for medicare advantage plan and part d.
age_group	1: Age under 60 (10% of the sample) 2: Age between 60 & 70 (40% of the sample) 3: Age above 70 (50% of the sample)
vulnerable_group	This group indicates people who possibly have 3 highs chronic conditions simultaneously. It is generated when the columns of rx_gpi2_27_ind, rx_gpi2_36_ind and rx_gpi2_39_ind are all equal to 1. (16.2% people who have transportation issues are in vulnerable group)

The final dataset has 69752 observations and 1882 variables, which are all numeric values. For the purpose of supervised learning and preventing overfitting, we split the data into training set and validation set, given the size of validation set is 20% of the whole data.

Modeling

Model Selections & Rationale

Because the target value, “transportation_issues”, is a binary categorical variable, so we decided to test and apply the following classification algorithms approach the challenge:

- XGBoost
- LightGBM
- Decision Tree
- Random Forest
- CatBoost

XGBoost

XGBoost, also known as extreme gradient boosting, is one of the most popular machine learning algorithms for case competitions. Moreover, it is built on gradient boosting framework principles, whose goal is to minimize prediction errors by combining the next best model with the previous one. According to Kan Nishida, XGBoost “is designed to push the extreme of machines' computation limits to provide a scalable, portable, and accurate library."

LightGBM

According to LightGBM’s official documentation, “LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages: Faster training speed and higher efficiency. Lower memory usage.”

LightGBM is known for its efficient and robust performance on a large scale of the dataset. Usually, people would use LightGBM for data with over 10,000 observations.

Decision Tree & Random Forest

Decision Tree is a very classic classification algorithm for machine learning. Furthermore, Random Forest is just another algorithm that will generate the final output by combining many Decision Trees that are generated randomly in the training process. We decided to use their outputs as our baseline to evaluate how well the other algorithms perform.

CatBoost

CatBoost is one of the most dominating and powerful classification algorithms in recent years. According to Tal Peretz, CatBoost " is the successor of the MatrixNet algorithm that is widely used within the company for ranking tasks, forecasting and making recommendations." During the experiments, we discovered that CatBoost does have a great API and very easy to implement.

Another cool thing to demonstrate the CatBoost's strong ability to learn and process is it reads the categorical variables directly by specifying "one_hot_max_size", which means we could use the original data without doing much transformation.

Individual Model Output & Compare

As we discussed above, we split the data randomly and made 20% of the data validation set to prevent overfitting. We first test each individual model without hyperparameter optimization, and then performed parameter tuning to see how much the model would improve. According to Suzanne Ekelund, the area under the roc curve is a measure of our model's usefulness. Thus, we decided to use the roc_auc score as the evaluation metric for our models.

We started to test each model without tuning anything to find the baseline of the performance of our model. The performance difference among the five algorithms is as large as 0.2 in the term of AUC score. And the results are shown in Figure 3.1:

Models	BaseAUC
Xgboost	0.7219901
Lightgbm	0.6845420
DecisionTree	0.5638988
CatBoost	0.6085949
RandomForest	0.7122100

Figure 3.1: Performance Outputs Without Tuning

After cleaning the data and tuning the parameters for every individual model, we fitted the models with the validation dataset. We then generated a summary table to compare the performances of each model. We used the Accuracy Score, AUCPR, and ROC AUC Score as the measurements. The outputs are shown in Figure 3.2:

Algorithms & Outcomes			
Algorithms	Accuracy	AUCPR	AUC
Xgboost	86.35%	0.180811	0.7497152
Lightgbm	70.08%	0.347931	0.7508940
DecisionTree	85.94%	0.176114	0.7120952
CatBoost	79.86%	0.228834	0.7379980
RandomForest	85.84%	0.151847	0.7289980

Figure 3.2: Models Performances Summary Table

According to the performance summary table, the five models' average performance improvement is 12.69% after tuning the parameters. Decision Tree had the most significant improvement but still has the lowest AUC score among the other models. XGBoost and LightGBM had better roc scores among the algorithms. Even though the accuracy score for LightGBM is about 16% lower than XGBoost, after we studied the confusion matrix for each model, we found it is unnecessary to be treated as a bad thing. XGBoost produced a higher false-negative rate, while LightGBM produced a higher false-positive rate. In other words, XGBoost tends to classify a new/unseen customer as the group without transportation issues. Since our data is hugely unbalanced, XGBoost was likely fooled in the process of training. There is a trade-off. We believe Humana would rather avoid a lack of preparation since there are potentially more customers who have transportation difficulties than the model indicates.

Weighted Average & Stacking Ensemble

To better our final model's AUC performance, we decided to apply weighted average ensemble and stacking ensemble techniques to the existing models. Both ensemble techniques are designed to improve the model's performance while lowering errors. The rationale is very similar to the diversification in financial investment, which will lower the risk by allocating money to different projects. Therefore, we supposed that using the outputs from multiple models to develop a "portfolio" to boost our final model's performance will eliminate the risk that any of them went wrong unexpectedly.

However, the output does not show a significant improvement from the individual model. We assume that this outcome happens due to the following reasons:

- The individual models' outputs are highly correlated, which add no diversity to the stacking model
- We did not tune the parameters for the stacking model as a whole, which means it could not have reached its optimization yet
- The data itself is not able to provide sufficient information to reach a higher score, and both individual models and stacking model are performing on their highest levels

Final model & parameters

After studying many rounds of testing outputs by splitting the data into different approaches, we finally decided to use LightGBM as our final model. It gives the most consistent AUC performance while not showing much of overfitting issues. To optimize the model's performance, we used the one of the APIs on neptune.ai and GridSearch to tune the parameters and the final parameters we applied to LIghtGBM is listed in Figure 3.3:

LightGBM	
PARAMETERS	***INPUT***
objective	binary
metric	auc
num_class	1
is_unbalance	TRUE
boosting_type	dart
learning_rate	0.11
max_depth	12
num_leaves	12
feature_fraction	0.13
lambda_l1	17
lambda_l2	890
max_bin	1017
subsample	0.38
num_iterations	681
min_data_in_leaf	1400
tree_learner	data
AUC	**0.751721**

Figure 3.3: Parameters for The Final Model

Analysis

Important features

We generated two features plots from our final model (Figure 4.1 & Figure 4.2), regarding Top 20 by importance and gain respectively.

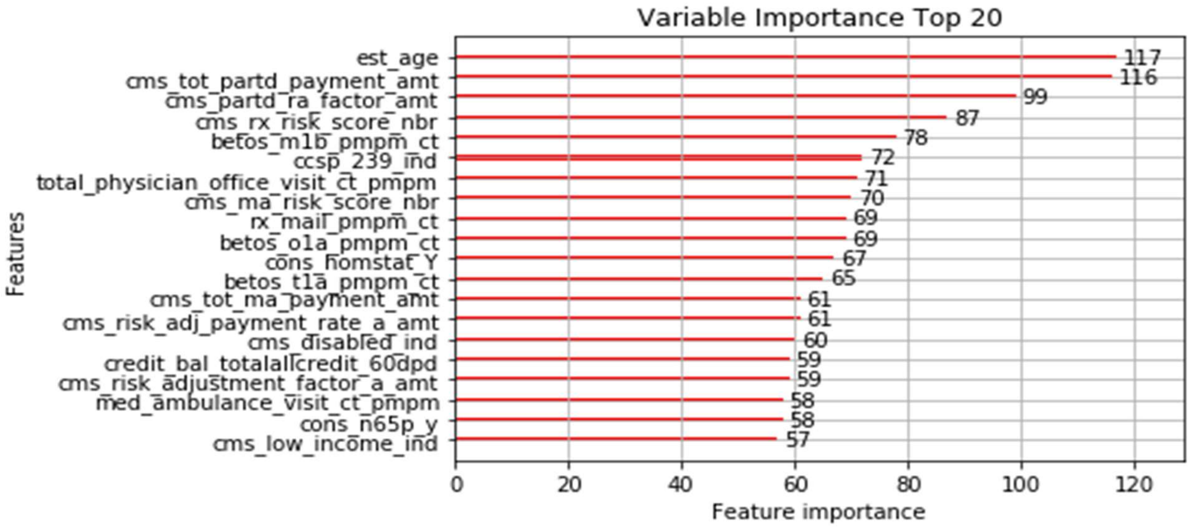

Figure 4.1: Top 20 Features by Importance

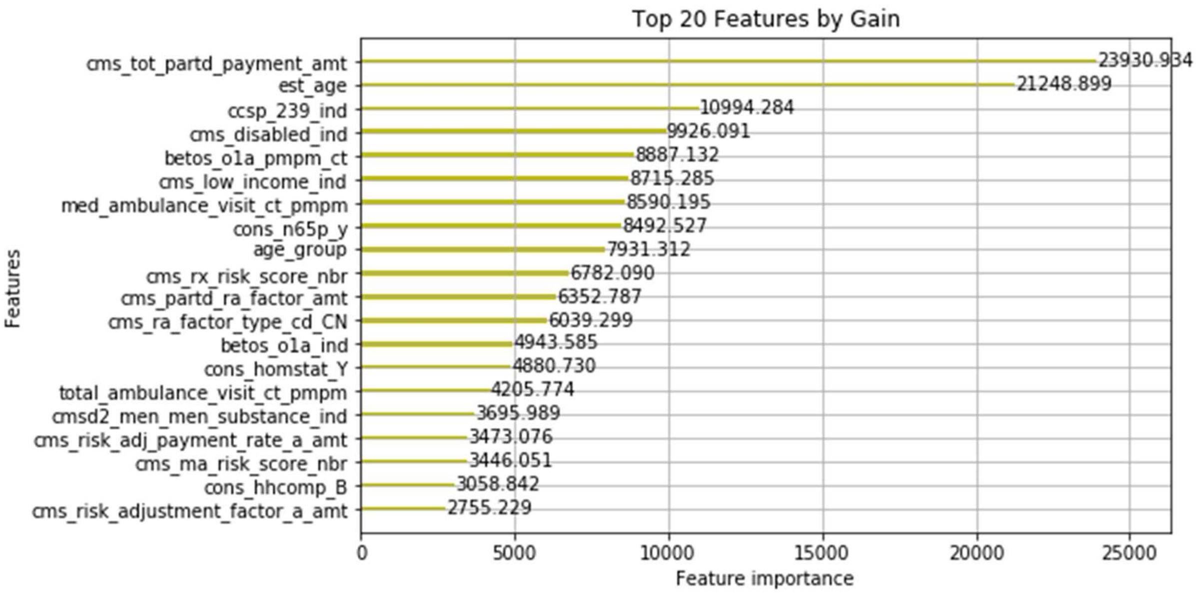

Figure 4.2: Top 20 Features by Gain

As we can notice, there are many overlaps (e.g. cms_tot_partd_payment_amt & est_age), indicating Humana should take a serious look into those features that are shown on both charts. They could come from different social classes or professional industries, but their paths somehow come across in the terms of transportation challenge.

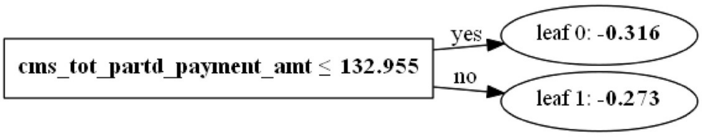

Figure 4.3: Illustration of The Final Model

Figure 4.3 shows how the model interacts with the essential variables and how it can predict the potential transportation issue.

Studying from the plot above, we can conclude that if the customer makes less than or equal to $132.96 on the Part D payment, the model will classify the customer as 0 on the leaf; otherwise, the customer will be marked as 1 on the leaf. The data shown in Figure 4.4 can be strong evidence to support the insight derived from this plot.

The average Part D payment made by the entire sample is about $145.19. However, when we broke the sample into two groups by identifying if they had claimed they had transportation issues, the average payment difference became big. We can see that people with transportation issues would pay $65.94 more on average. That is approximately 48.41% of what people without transportation difficulties would spend.

	Info_partd_payment	Info_partd_Yes	Info_partd_No
count	65822.000000	8967.000000	56855.000000
mean	145.199206	202.156691	136.216043
std	107.731191	132.460194	100.384984
min	0.000000	0.000000	0.000000
25%	81.700000	94.335000	80.440000
50%	101.440000	147.840000	99.370000
75%	184.670000	294.765000	141.105000
max	857.910000	857.910000	857.910000

Figure 4.4: Statistics of PartD Payment for Groups by Transportation Challenges

Of course, the real model will be more complicated than the plot shows. Furthermore, there will be many more nodes between the root and the leaves. Nevertheless, Figure 4.3 is an excellent illustration of how the model functions.

There is another interesting feature we felt worth talking about is Total_betos_pmpm. This is a feature we created during the process of feature engineering. It is a sum of all the logical claims filed for BETOS code per member per month. Figure 4.5 shows that the average claims per member per month are approximately 0.51 times more for people with transportation issues. They are the kind of people who will visit the office or hospitals and use ambulance or medical services more than anybody else. However, they are also people who had transportation barriers. Therefore, by paying more attention to this group of customers, Humana will be able to discover their difficulties and bring care to them promptly.

	None	Issues
count	59375.000000	10197.000000
mean	1.949440	2.462947
std	1.713437	2.252997

Figure 4.5 The Average of BETOS Claims by The Two Groups

Recommendation

From the important features of the model we built, some member groups need to be paid more attention to. For instance, the top 3 important features are related to age and Medicare history, and they are est_age, cms_tot_partd_payment_amt, total_ambulance_visit_ct_pmpm. Therefore, we listed the following member groups with a higher risk of transportation issues and put forward some advice for Humana.

- “Basic information” : Elder and disabled members.
- “Credit status”: Bad credit record and low-income members.
- “Medicare and drug” : High CMS risk score and those who had used ambulance service members.

From Humana’s website, Florida Medicaid provides members free rides to their doctor, the pharmacy, or a healthcare visit by planning a ride. Members can both schedule the trip online or make a phone call. However, the service may still be insufficient. For example, members need to call or schedule online at least 24 hours before the doctor’s visit, which means the customer who has urgent needs will not get the service. The last updated time of the free rides service in Florida was two years ago. Given the pandemic situation we are experiencing right now, we suppose there will be more demands for the timely transportation services than ever. So, we suggest Humana to track this information and make the prompt updates.

Recommendation & Business Insights

Humana may cooperate with Lyft or Uber, providing a service as “ Humana Call.” Divers who join the project will arrive at the pickup place timely and give passengers safe rides. This program should allow Humana members to get rides immediately and schedule round trips after their office visit. Members in the US can easily access this service at a lower cost or even get covered by the insurance plan for certain types of diseases, like people with a disability or in the 3 highs group.

For those members who have a good credit record but low income, since they will become the potential valued customers (such as Ph.D. students), we encourage Humana to develop and expand more services in more areas to solve their transportation difficulties or render services to them at a lower rate.

One of the solutions is cooperating with banks and car dealers to help members acquire loans and cars more quickly and easily. Humana can also launch programs with local DMVs and Department of Transportation to ensure a smooth channel to complete related applications for the customers with difficulties. For example, due to Covid-19, driving seems a safer way to routinely commute than other public traffic means. If Humana could cooperate with car dealers and offer a lower price for those members with transportation barriers, it would be the service that will win a good reputation for Humana as well.

Reference

Sharma, A. (2020, May 12). Decision Tree vs. Random Forest - Which Algorithm Should you Use? Retrieved October 10, 2020, from https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/

Brownlee, J. (2020, April 21). A Gentle Introduction to XGBoost for Applied Machine Learning. Retrieved October 10, 2020, from https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

Dwyer | February 5, K. (2018, February 5). Workers' Comp Patient Transportation. Retrieved October 11, 2020, from https://riskandinsurance.com/service-spotlight-qa-one-calls-linda-colsen/

Ekelund, S. (2011, January). ROC curves – what are they and how are they used? Retrieved October 10, 2020, from https://acutecaretesting.org/en/articles/roc-curves-what-are-they-and-how-are-they-used#:~:text=As the area under an,compare the usefulness of tests.&text=In addition, the area under,test(s) in question.

Güneş, F. (2017, May 18). Why do stacked ensemble models win data science competitions? Retrieved October 11, 2020, from https://blogs.sas.com/content/subconsciousmusings/2017/05/18/stacked-ensemble-models-win-data-science-competitions/

Heath, S. (2019, November 20). Overcoming Transportation Barriers to Drive Patient Care Access. Retrieved October 10, 2020, from https://patientengagementhit.com/news/overcoming-transportation-barriers-to-drive-patient-care-access

H. (2018, October 1). Medicaid transportation services. Retrieved October 11, 2020, from https://www.humana.com/medicaid/florida-medicaid/medicaid-extras/transportation

Nishida, K. (2017, March 21). Introduction to Extreme Gradient Boosting in Exploratory. Retrieved October 10, 2020, from https://blog.exploratory.io/introduction-to-extreme-gradient-boosting-in-exploratory-7bbec554ac7

Peretz, T. (2018, October 18). Mastering The New Generation of Gradient Boosting. Retrieved October 10, 2020, from https://towardsdatascience.com/https-medium-com-talperetz24-mastering-the-new-generation-of-gradient-boosting-db04062a7ea2#:~:text=CatBoost is an algorithm for,tasks, forecasting and making recommendations.

Welcome to LightGBM's documentation!¶. (n.d.). Retrieved October 10, 2020, from https://lightgbm.readthedocs.io/en/latest/#:~:text=LightGBM is a gradient boosting,Lower memory usage.