Capstone Team 15 Final Report

Amber Chen, Shuming Chen, Tairan Deng, Kaiwen Zhang, Ali Daanesh Sayyed

Professor Tej Anand & Dan Mitchell

Capstone Spring 2021

University of Texas Austin

May 15, 2021

The University of Texas at Austin
Texas McCombs
MS Business Analytics

Executive Overview

The Capstone Team 15 worked with Walmart Global Tech support team to look at Walmart employee tech support data. The overarching goal was for us to analyze the data and find patterns that stood out with the goal of improving and enhancing their current tech support system especially since WFH had overwhelmed the current support structure.

The overarching business question, as mentioned before, was how could the team improve tech support efficiency for Walmart employees. There were two questions that we solved

1. How can the team improve Walmart's current SLA
2. How could the team optimize ticket reassignments for the support team

Those are important because the current process is time consuming, causing delay, and missing the employee expectation. SLA & Ticket reassignments were the two areas we focused on after data exploration analysis in order to reduce process time and potentially increase the Tech Support capacity, hence why the team dived deeper into these two topics.

From a practical standpoint, we bring value to Walmart by optimizing their tech support teams ability to efficiently deal with tickets as well as financially benefit the company through reducing the frequency of ticket reassignments.

The team dealt with a dataset made of 141,032 tickets with 193 attributes per ticket (though it is worth noting that many have missing values). The dataset was only for the United States and included data from January 2020 to February 2021.

The analytical findings of our team consisted of visualizations as well as a variety of models. The team utilized tableau to create intuitive visualizations for Walmart. As far as models go, we created a predictive model to address the SLA business question by predicting the amount of tickets the different support departments would have and also built a classification model for ticket reassigning (thereby mitigating the issue by the root source).

We recommended to Walmart the following:

- Automate lower priority tickets within the queue
- Use predictive modeling to predict the ticket traffic and have staff on hand accordingly
- Use classifications models to classify tickets based on certain criteria, thereby reducing reassignment occurrences significantly
    - Potentially build a chatbot based on the classification model to help employees with low priority issues that could be resolved with FAQ hyperlinks

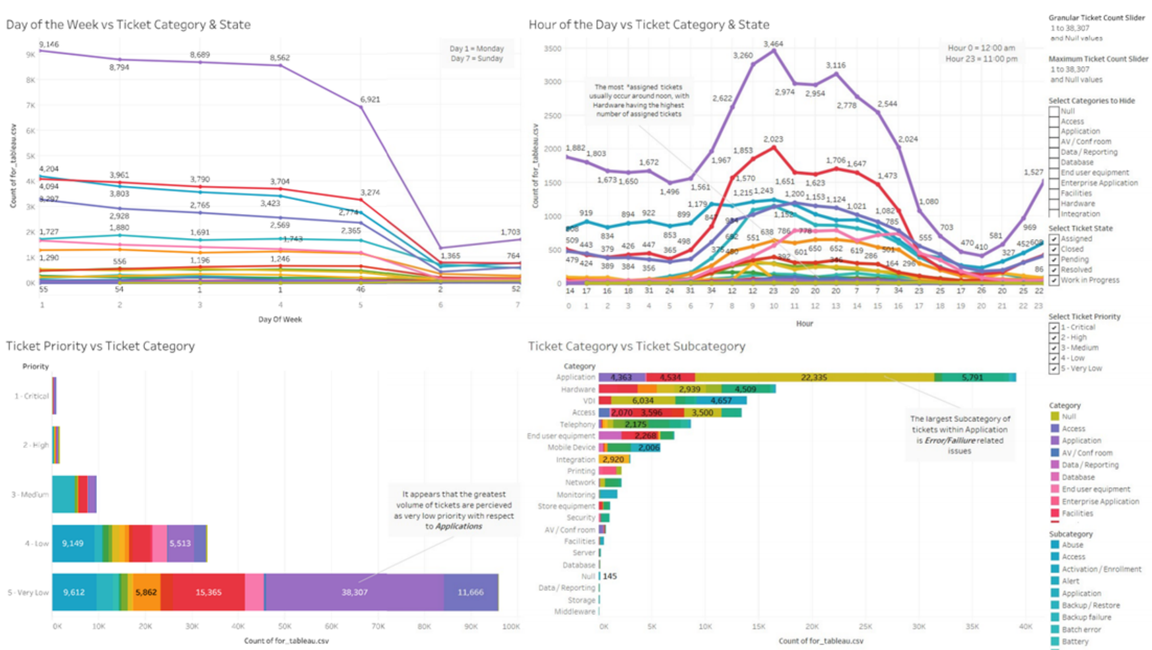
All of the aforementioned recommendations help increase Walmarts employee tech support teams efficiency, which in turn also financially benefits Walmart.

Source Data Exploration

The source data provided by the client was an excel spreadsheet consisting of 141,032 tickets and 193 attributes. The data covered the United States only, and spanned January 2020 to February 2021. The data essentially contains all specifics regarding the ticket like: duration of ticket, department ticket was assigned to, ticket reassignment, who worked on the ticket, etc.

Some notable findings from the data included the following (Below is the visualization built from the data via Tableau) :

- ❖ 68% of all tickets were very-low priority
- ❖ 40% of all very-low tickets were application related
- ❖ The top ticket categories were application, hardware, VDI



The biggest limitation we faced in the data was that a significant amount of the data was omitted for privacy reasons. This forced us to be more selective with the attributes we analyzed. Aside from omitted data the dataset was usable, and provided us an ample amount of insights. As far as useful information goes, our findings are beneficial to Walmart, as we found several concerning trends that, if addressed, will help improve Walmarts efficiency.

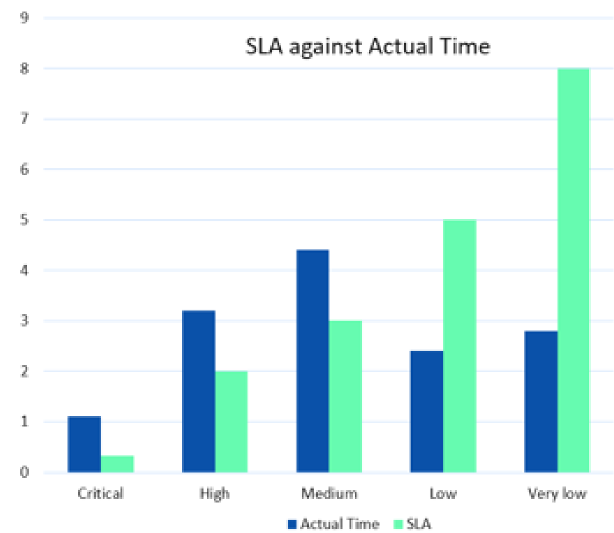
As far as data engineering and transforming goes, our data was relatively fine to work with as is. The team really only truncated the dataset to simplify the visualization processes. Any other transformations were general manipulations used for reaching the insights that the team reached. A simple transformation involved cleaning up certain attributes that involved text (i.e. ticket comments and descriptions) into a format that can be used by our natural language processing packages and classification models.

Explore Causes of Current Problems

Ticket processing prioritization

In this project our main concern is the efficiency of ticket solving. According to Walmart`s expected SLA, from priority high to low, tickets are allowed 8 hours, 2 days, 3 days , 5 days and 8 days to solve. However, on average the actual results are far behind the SLA standards. Critical, high and medium priority tickets are at least 50% over time. Please see the graph and chart below.

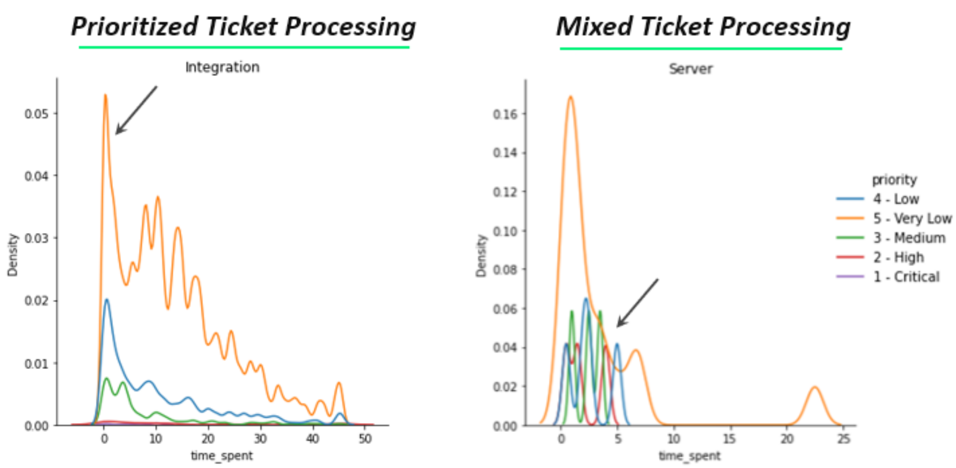
Priority	Actual	Standard
Critical	***1.1 days***	***8 hours***
High	***3.2 days***	***2 days***
Medium	***4.4 days***	***3 days***
Low	2.4 days	5 days
Very Low	2.8 days	8 days



As we can see, the critical, high and medium critical tickets are over time, while the low and very low priority tickets have extra time to finish on time. These two categories of tickets composed over 70% of the 140,000 cases processed in 2020. Imagine that if Walmart can allocate technician resources more on critical, high and medium priority cases, Walmart can expect cases of all priorities could be solved within expected timeframe.

After our team took a closer look at tickets processing in different categories, we have noticed that prioritizing problems exist in certain tech-support teams. Normally critical/high priority tickets could cause higher business loss and need to be solved at higher priority, this is also the reason why the SLA is set to 8 hours/2 day. Ideally tech-support teams would process new coming high priority cases in the first place but from our data exploration, some categories tech-support teams are operating under “first come, first serve” rule and have a significant size of backlog cases under critical/high cases.

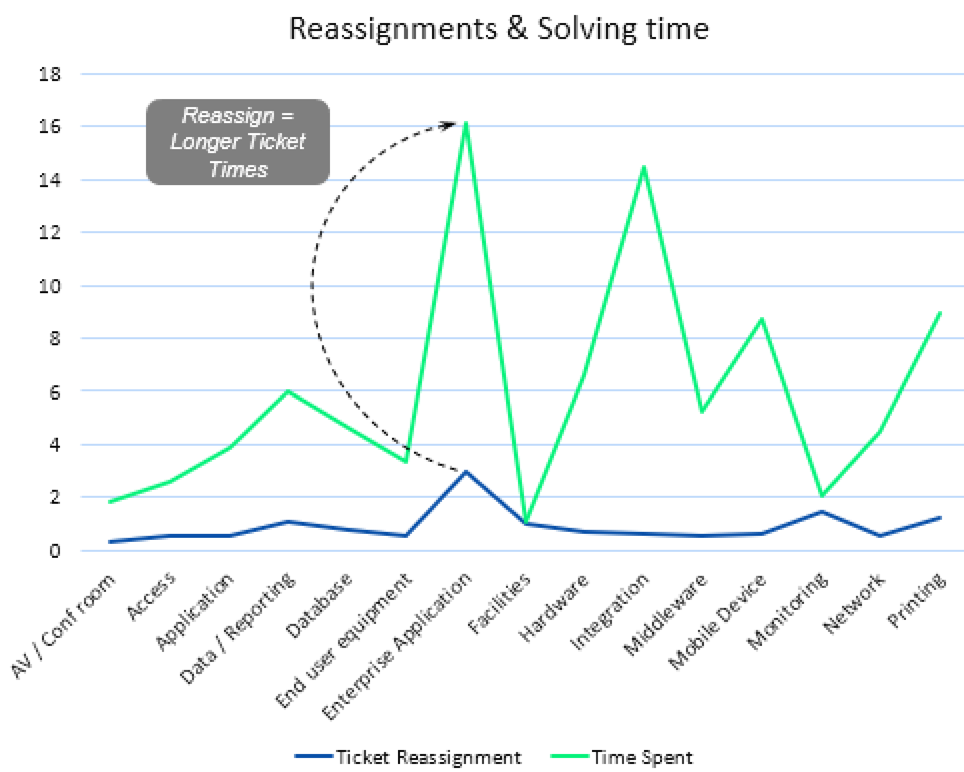
The following graph is a plot of density against average time spent on cases in different priorities. The integration tech-support team displayed good prioritizing operation as shown in the plot on the left. Low/very low priority tickets are processed in a wide timeframe, on the contrary, most medium/high priority tickets are processed in a shorter time (within 3-5 days). This indicates the technicians put high priority tickets in the first place disregarding the sequence of tickets creation. However, when looking at the server tech-support team (on the right), it is obvious that except for very low priority tickets, all other tickets` solving time are evenly distributed within 5 days. This could be explained by the “first come, first serve” operation process they're utilizing. Our team would like to recommend more employee training regarding prioritizing tickets.



The following tech-support team has been identified to have prioritization problems: server, data/reporting, printing and telephony.

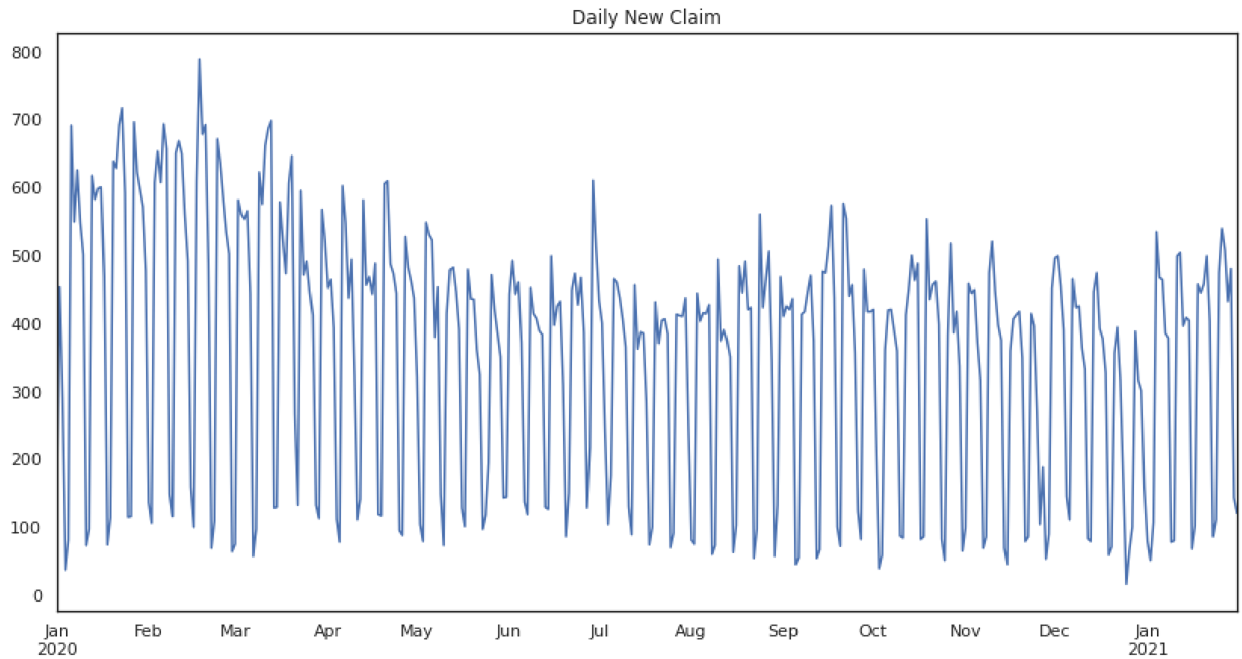
Classification Error

During our exploration of data patterns, our team also noticed that over 50% of tickets have been reclassified into another category. When a new ticket is created in the system, a technician will look at the ticket and assign a category to it. When the ticket has been worked on by the assigned category team and was found to be in the wrong category, this ticket will be sent to a different team`s queue to get solved. This would double the backlog time and significantly increase ticket processing time. From the below graph we can see for some categories, processing time peak when tickets have high reassignment counts. For instance, data/reporting, enterprise application and printing. The high reassignment count could be a reason for severe overtime in the mentioned categories.



Model Building

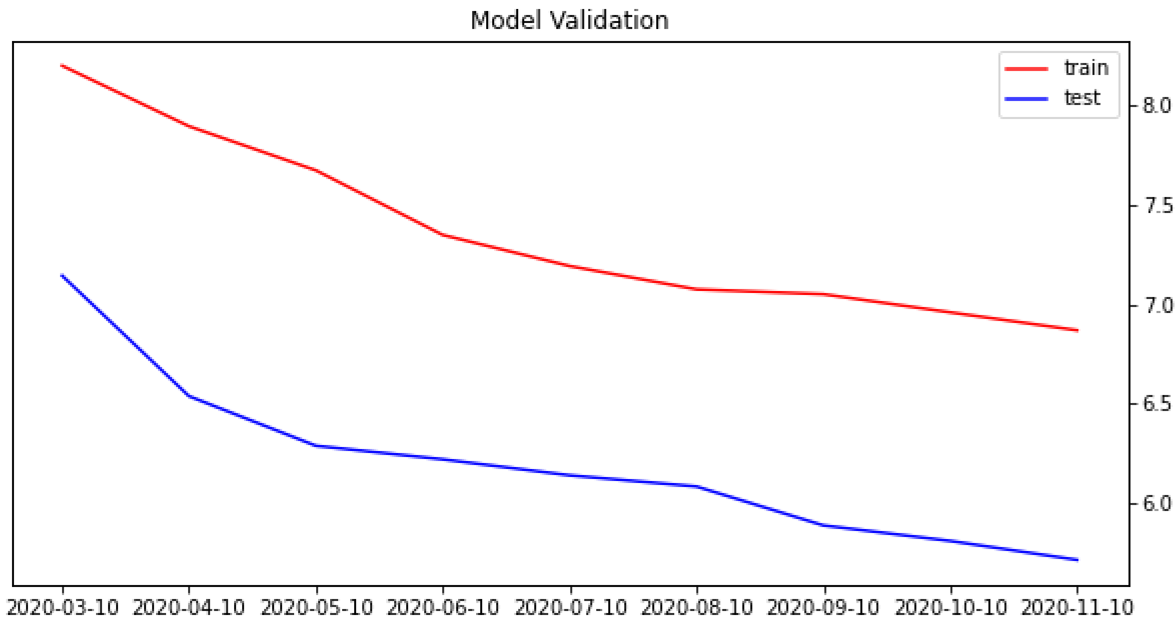
Time Series Analysis



Above is a graph summarizing the daily new claim across the time horizon. And we notice there is a weekly pattern existing. An autoregression model is then built, regressing the hourly ticket volume on a number of time series factors as described below. The purpose of the time series model is to accurately predict ticket incoming based on historical data to provide better guidelines to the staffing need and schedule.

Variable Names	Calculation / Formulation
1-24 hour(s) prior ticket count	Hourly ticket count x hours prior
1-7 day(s) prior ticket total	Daily ticket count x days prior
Day of Week	7 days in each week, extracted from the datetimeIndex
Hour of Day	24 hours in each day, extracted from the datetimeIndex

The graph below is presenting the train & test RMSE in model validation. The result is improving as the sample size gets larger.



Supervised Learning: Classification

As the exploratory analysis suggested, reassignment of tickets could be one of the potential reasons for inefficiency in the current system. According to the data set, there are a total of 22 different categories and about 50 thousand tickets were reassigned to a different team at least once. This indicates that about half of the tickets were classified incorrectly initially. The approach the team took to this problem is to build a multi-class classification model that accurately classifies the tickets into the correct category which reduces the probability of ticket reassignments. This model can be used to potentially reduce reassignment count caused from manual labor. Walmart tech support then can use this model to implement some sort of automation such as a chatbot system.

The team conducted 2 classification tree models, XGBoost and CatBoost. Both models use the short description of tickets provided by the employees that submitted the tickets as the independent variable and the category of tickets as the dependent variable.

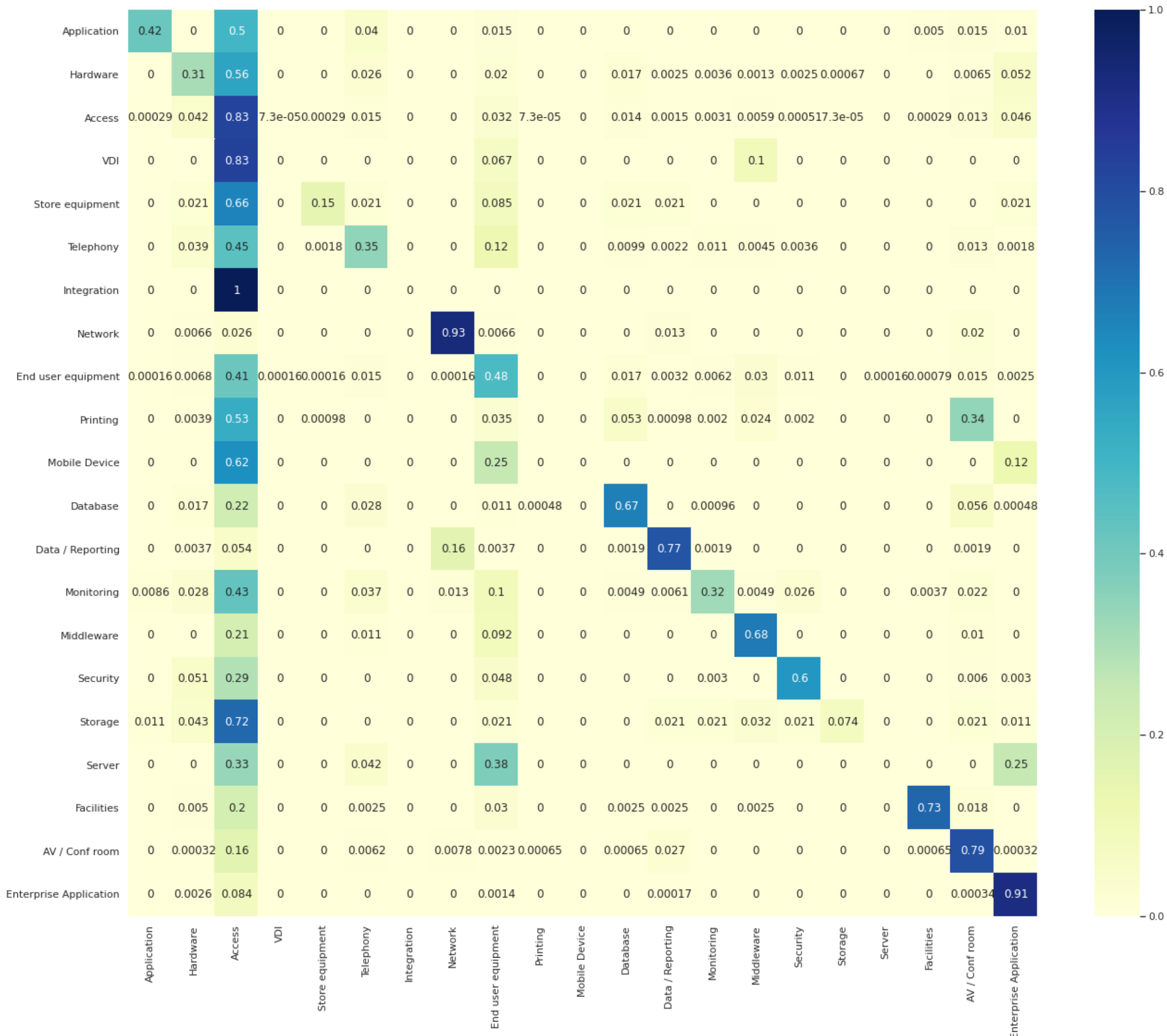
XGBoost

The independent variable needs to be preprocessed before it is passed into the XGBoost Classification algorithm. Lemmatization, tokenization was applied to the text to better understand the meaning and context of the messages left by the employees. TF-IDF vectorizer then is applied to the text to transform it into numerical values.

Before model training, we splitted the data set into training and test sets. The training set is then further split into validation and training sets. We built our own grid search cross-validation algorithm for hyper-parameter tuning. The algorithm applies TF-IDF vectorization on the data for every iteration of cross-validation to ensure the model is actually looking at the correct and original content of the short description as compared to vectorized before grid search which changes the form of the data before training starts.

	precision	recall	f1-score	support
AV / Conf room	0.87	0.42	0.56	201
Access	0.63	0.31	0.41	4452
Application	0.56	0.83	0.67	13747
Data / Reporting	0.00	0.00	0.00	30
Database	0.41	0.15	0.22	47
End user equipment	0.59	0.35	0.44	2232
Enterprise Application	0.00	0.00	0.00	2
Facilities	0.54	0.93	0.68	151
Hardware	0.74	0.48	0.58	6306
Integration	0.00	0.00	0.00	1025
Middleware	0.00	0.00	0.00	8
Mobile Device	0.75	0.67	0.71	2075
Monitoring	0.73	0.77	0.75	536
Network	0.66	0.32	0.43	815
Printing	0.65	0.68	0.67	881
Security	0.63	0.60	0.61	333
Server	0.64	0.07	0.13	94
Storage	0.00	0.00	0.00	24
Store equipment	0.95	0.73	0.83	400
Telephony	0.74	0.79	0.77	3080
VDI	0.86	0.91	0.88	5825
accuracy			0.65	42264
macro avg	0.52	0.43	0.44	42264
weighted avg	0.65	0.65	0.63	42264

This model achieved an overall accuracy of 0.65 after performing hyper-parameter tuning. Since this dataset is imbalanced, F1 score is also a very important measure to consider. F1 score is a harmonic average of precision and recall. Precision measures the number of correctly predicted instances out of all prediction instances for a specific category. Recall measures the number of correctly predicted instances out of all data points of a specific category. This model achieved a weighted F1 score of 0.63.



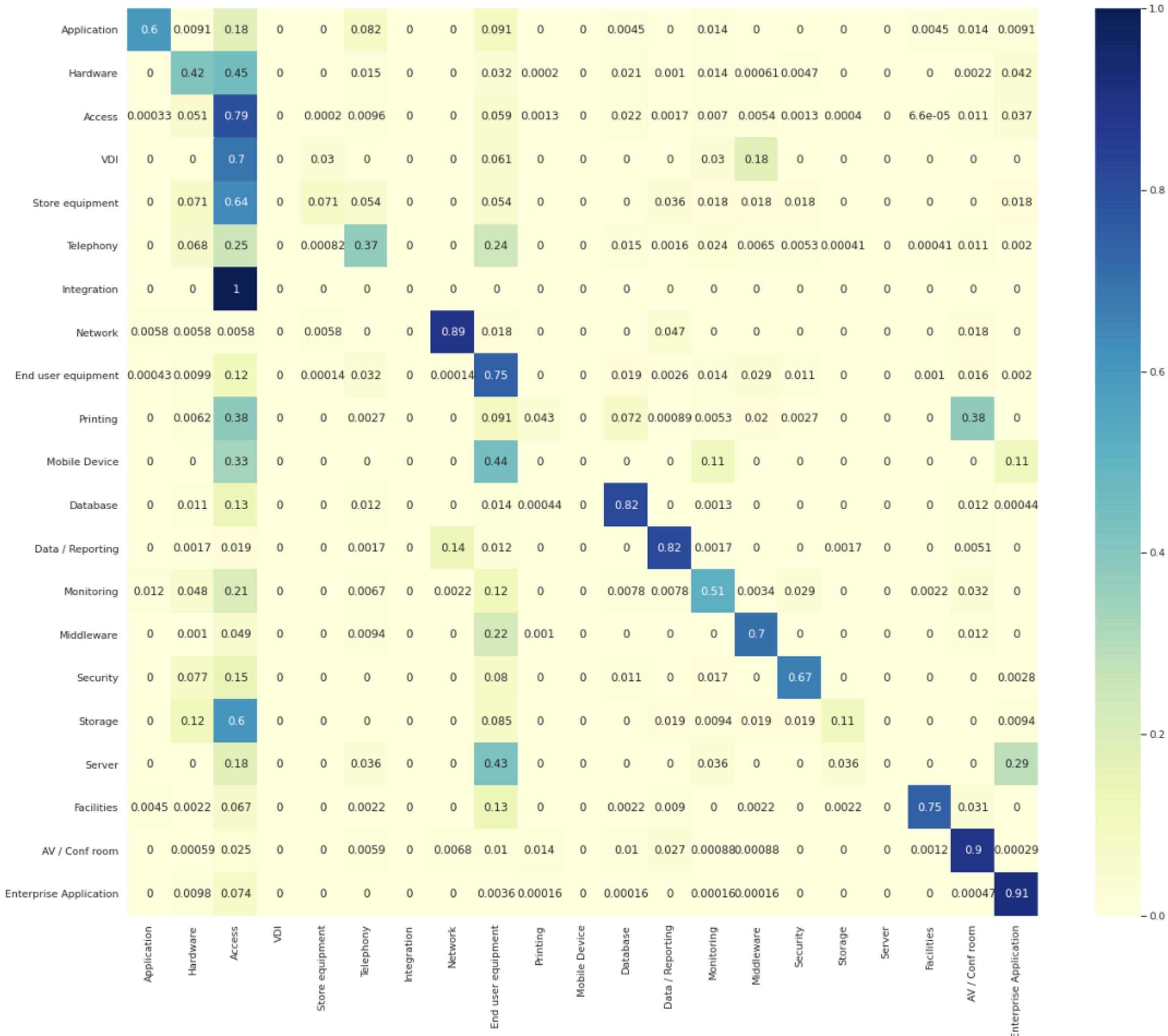
Looking at the confusion matrix, we can see that the model performs well for categories such as Access, Network, Data/Reporting, Facilities, AV/Conf room and Enterprise Applications. The model performs poorly for categories such as VDI, Integration, Mobile Device and Server. VDI stands for virtual desktop integration and is mainly classified as Access. VDI could be considered as a type of access to a specific system. Similarly, Mobile Device and server were mainly classified as End user equipment which both are types of equipment. This indicates that our model is not granular enough.

CatBoost

The next model the team conducted was a CatBoost classification model. Train, validation, test split is also conducted before model training. Since the CatBoost package is able to take in text data as input, text preprocessing was not necessary before training the model. Just like XGBoost, the team also implemented a grid search cross validation strategy for hyper-parameter tuning. The resulting parameters are depth of 4 for the classification tree, 12_leaf_reg of 1 and learning rate of 0.1. This model achieved an overall accuracy of 0.72 and a weighted F1 score of 0.70.

	precision	recall	f1-score	support
AV / Conf room	0.86	0.60	0.70	220
Access	0.63	0.42	0.50	4918
Application	0.69	0.79	0.74	15107
Data / Reporting	0.00	0.00	0.00	33
Database	0.33	0.07	0.12	56
End user equipment	0.63	0.37	0.47	2452
Enterprise Application	0.00	0.00	0.00	2
Facilities	0.59	0.89	0.71	171
Hardware	0.69	0.75	0.72	6940
Integration	0.41	0.04	0.08	1128
Middleware	0.00	0.00	0.00	9
Mobile Device	0.72	0.82	0.76	2265
Monitoring	0.74	0.82	0.78	588
Network	0.56	0.51	0.54	893
Printing	0.67	0.70	0.68	955
Security	0.60	0.67	0.63	362
Server	0.55	0.11	0.19	106
Storage	0.00	0.00	0.00	28
Store equipment	0.95	0.75	0.84	445
Telephony	0.78	0.90	0.84	3394
VDI	0.88	0.91	0.90	6418
accuracy			0.72	46490
macro avg	0.54	0.48	0.48	46490
weighted avg	0.71	0.72	0.70	46490

Looking at the confusion matrix, the model performs greatly on categories such as Access, Network, End user equipment, Database, Data/Reporting, Middleware, Facilities, AV/Conf room and Enterprise Applications. On the other hand, the model performs poorly on categories such as VDI, Integration, Mobile Device and Server which coincides with the result of the XGBoost model. The team believes that this is due to the same reason as the XGBoost model.



Furthermore, one of the potential reasons for inaccuracy in these categories also lies in the data set itself. The data set is imbalanced and the categories that performed poorly lay on the tail of the distribution. The team believes that if more data on these categories were provided, performance on these categories for both models would potentially increase immensely.

Using CatBoost for improvement analysis

The team first found the model prediction accuracy of each ticket category (based on the confusion matrix of the Catboost model), and then compared that with their respective reassignment rates as a measure of the "no-model prediction accuracy" of a ticket's correct category. Based on these two metrics, the team determined the categories for which our model did better than the no-model scenario. These will be our categories of focus, and the team will only be including these categories in our analysis from this point on.

Dividing our model accuracy by the “no-model accuracy”, the team calculates an “improvement” ratio for our categories of focus. the team use this improvement value to be divided by each ticket’s reassignment count to get its “reduced reassignment count” (e.g. if a ticket belonging to a category of focus and was reassigned once, and the improvement ratio for that category was 2, then the team say that that ticket will have a reduced reassignment count of “half” using our model). The difference between the original reassignment count and the reduced reassignment count will be the amount of reduction.

	category	reassignment_count	hour	calendar_duration	improvement	reduced reassignment_count	reassignment_count reduction amount	reassignment_to_duration_coefficient
8	Access	0	23	5028	1.142340	0.000000	0.000000	410205
9	Access	0	23	5149	1.142340	0.000000	0.000000	410205
12	Access	0	23	20697	1.142340	0.000000	0.000000	410205
13	Access	0	23	21293	1.142340	0.000000	0.000000	410205
16	Access	0	22	7021	1.142340	0.000000	0.000000	410205
...	...	...	...	...	...	...	...	...
141008	AV / Conf room	0	8	66	1.149429	0.000000	0.000000	444807
141009	AV / Conf room	0	8	61	1.149429	0.000000	0.000000	444807
141010	AV / Conf room	0	8	203	1.149429	0.000000	0.000000	444807
141015	Network	1	7	1733585	1.227849	0.814432	0.185568	359642
141021	Monitoring	1	4	17212	3.853112	0.259530	0.740470	89438

28453 rows × 8 columns

Now, the team would like to find a way to convert the reassignment count reduction to duration reduction; finding the reduction in the number of relays a ticket goes through is good, but it is far more business-relevant to determine how much reduction in the expected duration of a ticket would be possible through our model. For this, the team use a simple linear regression of ticket duration (measured in seconds) regressed on reassignment count -- done by each partition of category and hour of the day (e.g. one regression would be done only on the subset of data of category Mobile Devices and time 0:00, another of category Applications and time 14:00, etc.). Due to limitations of data such as no data for certain combinations of category and hour, or only one value of reassignment count for a certain combination, not all category-hour combinations have viable coefficients. To avoid this issue, the team decided to regress duration by reassignment count only by category and not by category and by hour. Using the coefficients from these regressions, the team can convert our determined reassignment count reduction amount to duration reduction amount. For the sake of simplicity and the two limitations mentioned above, the team assume that this coefficient / conversion factor should be applied to all hours for a category in our dataset (e.g. the team multiply all the reassignment count reduction amounts of tickets in Mobile Device by 748308, regardless of what hour of the day the ticket started in). With these coefficient values, reassignment reduction amounts can be converted to duration reduction amounts by being multiplied by the coefficients. The duration reduction amounts can then be subtracted from original durations to have the estimated reduced durations.

	Coefficients
Category	
AV / Conf room	444807
Access	410205
Data / Reporting	173611
Database	524508
End user equipment	543357
Facilities	0
Middleware	65292
Monitoring	89438
Network	359642

	category	reassignment_count	hour	calendar_duration	improvement	reduced reassignment_count	reassignment_count reduction amount	reassignment_to_duration_coefficient	reduced calendar_duration
8	Access	0	23	5028	1.142340	0.000000	0.000000	410205	5.028000e+03
9	Access	0	23	5149	1.142340	0.000000	0.000000	410205	5.149000e+03
12	Access	0	23	20697	1.142340	0.000000	0.000000	410205	2.069700e+04
13	Access	0	23	21293	1.142340	0.000000	0.000000	410205	2.129300e+04
16	Access	0	22	7021	1.142340	0.000000	0.000000	410205	7.021000e+03
...	...	...	...	...	...	...	...	...	...
141008	AV / Conf room	0	8	66	1.149429	0.000000	0.000000	444807	6.600000e+01
141009	AV / Conf room	0	8	61	1.149429	0.000000	0.000000	444807	6.100000e+01
141010	AV / Conf room	0	8	203	1.149429	0.000000	0.000000	444807	2.030000e+02
141015	Network	1	7	1733585	1.227849	0.814432	0.185568	359642	1.666847e+06
141021	Monitoring	1	4	17212	3.853112	0.259530	0.740470	89438	-4.901411e+04

28453 rows × 9 columns

The team then can, for the subset of data with our categories of focus, plot the sum of duration aggregated across the 24 hours, as well as the same thing but with the sum of reduced duration. These two plots provide a literal and visual representation of how the total duration of tickets originated at each hour of day gets reduced, by how much, and for which category. The sum of duration can be aggregated across the categories only, without separating out into the 24 separate hours, in order to provide comparisons of the relative and absolute reductions in duration experienced by the different categories. Notably, the Access category had a large absolute duration reduction of approximately 12.56 years, while the Monitoring category had a large relative duration reduction of approximate 27.8%.

This analysis does have limitations, one of which includes the regression model used to predict duration from reassignment count. Linear regression is the simplest of prediction models, and often simple is appropriate. However, there definitely can be more complex, better fitting models, given the nature of values in variables such as reassignment count and hour.

The type of values present in the dataset variables is a separate model limitation in and of itself. Reassignment count is considered a censored variable in the sense that it does not take on negative values, and 0 is the overwhelmingly most common value. As long as the independent variable, duration, does not have this characteristic (which is the case), then linear regression is a tolerable model, but there remains to be researched what limitations a censored independent variable can present to the model's validity.

From the team's analysis, the team found that the categories for which the model provides better ticket-category-identification accuracy are: AV / Conference room, Access, Data / Reporting, Database, End User Equipment, Enterprise Application, Middleware, Monitoring, and Network. The team were then able to quantify the savings in time that can be made. As a frame of reference, 20% of all tickets in the dataset belong to one of these categories. Two things are immediately clear, if the dataset is representative of Walmart's internal I.T. operations: the model has the ability to out-predict any categorization/prediction process used in the dataset for a significant percentage of the tickets, and that ability translates to higher efficiency for a majority of the technical issues that arise throughout Walmart's corporate teams.

Recommendations

Category	Decrease in Ticket Time Duration (Years)
AV / Conference Room	5.5% (0.36 years)
Access	5.9% (**12.56 years**)
Data / Reporting	8.3% (0.26 years)
Database	14.7% (0.67 years)
End User Equipment	3.7% (**5.07 years**)
Middleware	1.3% (0.01 years)
Monitoring	27.8% (**5.48 years**)
Network	4.8% (3.14 years)
Overall	***6.1% (27.57 years)***

$$\frac{\textbf{Salary}}{\textbf{Year}} * \frac{\textbf{1 Year}}{\textbf{2080 Work Hours}} * \textbf{27.57 years} * \frac{\textbf{8760 hours}}{\textbf{year}}$$

Low ($20k Salary) = > $2.3 Million

Medium ($35k Salary) = > $4.0 Million

High ($50k Salary) = > $5.8 Million

As shown above, the Catboost model, paired with attention to how reassignment count could be contributing to operational inefficiencies, can help deliver material savings in both time and money for the Walmart Global Tech Team. Walmart, with the assistance of this team, can dissect the coding process through which text corpuses can be fed into the developed model.

Walmart should implement this Catboost model using software engineering that automates this process in real time and produces near-instantaneous results in response to typed input. In other words, Walmart can create an interactive "chatbot"-like platform that will return one or multiple possible categories of issues for the text description of the I.T. issue that it detects. Employees of the Global Tech Team should be notified of the change, which should take place gradually through pilot programs and focus groups, which will allow a period of calibration.

The team set out to optimize ticket reassignments for the support team. Not only have the team done this by estimating the reassignment reductions produced by the Catboost model, but it was possible to exceed this metric and propose monetary savings of $2.3 million, conservatively speaking

Conclusions

From the findings the team concluded that Walmart can increase tech support efficiency. There are three concrete recommendations the team has for walmart

1. Automate very-low and low priority tickets
2. Use predictive models to anticipate ticket volume in order to preemptively prepare tech support departments.
3. Use classification models to preemptively classify tickets thereby reducing the time lost in reassigning tickets to their appropriate locations.

This will be accomplished by collaborating with Walmarts data science teams, building upon the groundwork the MSBA Capstone Team 15 has provided.

References

Moscariello, J. (2021, April 23). Personal Communication [Zoom]

Moscariello, J. (2021, April 30). Personal Communication [Zoom]

Overview of CatBoost. CatBoost. Documentation. (n.d.). https://catboost.ai/docs/concepts/about.html.

XGBoost Documentation - xgboost 1.5.0-SNAPSHOT documentation. (n.d.). https://xgboost.readthedocs.io/en/latest/.

Acknowledgments

The MSBA Capstone Team 15 would like to acknowledge the following individuals:

Tej Anand

Foster Cryer

Jesse Iyalekhue

Sami Kibria

Dan Mitchell

James Moscariello