**Saving the Titanic**

Saving the sinking ship is a reflection of my first big enterprise machine learning project attempt, mistakes made, course corrected and lesson learnt. This blog will offer you an opportunity to learn from my mistakes. Happy reading!

March 2019 - new organization, new team, new domain and 1 week down the path, I was offered first enterprise machine learning project to spot items in the warehouse that had highest chance of getting inventory adjusted. Certain number of items should be picked for audits on daily basis across 30+ warehouses. This process will replace existing logical SQL which picked similar items every day.

Project didn’t offer much business background or resources like project managers, business analysts, data analysts or any other organizational data experts. We were directed to a set of database tables with 80+ attributes and no data dictionary to look back to. With the data provided, expectation was to build a machine learning model and pilot 10 branches and advance to 30 branches in progression.

We met the nearest stakeholders who were owners of the warehouse database and table. They were only able to provide high level meaning of the data for the attributes.

With limited knowledge of data in hand, we started with Exploratory Data Analysis (EDA) to gains preliminary insights on the data. Using the insights and assumptions, we built Light Gradient Boost Model (LGBM) model with F1 score of 0.75 with recall of 1 (items having higher chance of getting adjusted) as 0.80 which seemed performant only for few branches. Our end customer was happy with the results and agreed to pilot.

Fast forward, by July 2019, we implemented first branch in development environment. Existing processing had 2% success rate of finding items, out machine learning model had a success of up to 5%. From percentage, it sounded impressive but from numbers it basically picked 3 more items in the scale of 100. Eventually that percentage decreased to 0.5 %

For next 2 months, we tweaked the training data and model and gained some success and eventually it started failing again. By end of November 2019, we were facing the following:

* We were still at 1 branch (out of 10 expected) and not able to stabilize model performance. Machine learning model was performing poorer than current process.
* We burnt almost 9 months of organization money on experimentation.
* We still were not confident on the attributes and data we were using.

Customers, senior management and out team were extremely upset with model performance and cost incurred. We collectively took a call to close the project in 2 weeks’ time if nothing changed.

**How we saved the ship?**

I took a pause to think on what are we missing, what can we do to understand the data better.

First thing that came to my mind was to schedule a visit to the warehouse where actual audit process is taking place. This proved out to be a game changer. At the warehouse. we learnt the principles of how items arrive and get shipped, how items are prioritized in isles and locations, what goes to back of warehouse, what stays in front, what goes into priority freezer, which items are picked by robot and which are manual and how audit process takes place. I came back with enlightening experience and detailed notes (and a $200 cop citation in between for driving over 10 miles in 30 miles/hr. zone 😊).

Now, the data started making more sense. Luckily, we also found a senior team member who spent significant time in warehouse operations and prolific with data. We discussed our notes and came up with essential attributes that signifies where and how an item is placed in warehouse. Additionally, we agreed on essential filter criteria’s that made out data more focused.

Another game changer feature engineering idea came to mind for unique fields like item# and location# where I grouped the values , got their counts to create a feature called popularity t-shirt size (S, M, L and XL) depending on how many times that item was picked for audit in the past. Used this similar logic for 4 other fields. Now we had extremely strong features.

Lastly, we came up with additional interesting feature call $ value of audit. It’s a multiplication of unit price and quantity.

Now we had 14 strong attributes. I fed them thru decision tree, random forest, XGB, LGB and Adaptive boosting classifier and guess what? 90% + F1 score for almost all model. We picked random forest since recall of 1 (items having chance of adjustment) was best amongst all. New Features we built were toppers in feature importance. We performed 15 separate validations including different dates, different data combinations, forcefully introducing imbalance and model worked perfectly in each of them with consistent F1 score.

Fast forward, Feb 2020, we built, tested and validated 5 branches and all models were stellar. We put them into dev for pilot and measure the performances over a month. Now the items being audited thru ML were in multiplication factors. They performed 7 – 10 times better.

May 2020, we implemented our solution across 50% of our warehouses in production. Our model is worth over $2 million in terms of inventory saved. We walked away with smile, happy customer, management and bright future ahead of us.

**What did we learn?**

* Upon start of a project, always request for a business partner/domain expert who can help you understand not only the data but also the context of the data from business angle.
* Finding correct stakeholders are game changers for machine learning projects especially the ones who are on ground. Reach out to them to walk through the end to end process.
* Never hesitate to ask questions on the data you are using, no matter if you are new or pro in an organization.
* As you spend more time with ML project, build out good relationship with all stakeholders. They will help you answer many questions outside of scheduled meetings.
* Database and table owners may or may not be correct stakeholders for your questions. Gather as much information as you can but definitely bounce it against business partner/domain expert.
* Exploratory Data Analysis a must. This will open your vision on which features to select and use. Upon completion of EDA validate your findings with business partner/domain expert.
* Check for data boundary. For the given requirements, ask stakeholders on what data should we exclude for our requirement.
* Feature engineering is very important.
  + Do not ignore columns with unique data. See if you can group them and the count as can be used as a categorical variable.
  + For columns with continuous value, see if you can bin them and come up with categorical values.
  + Prices are usually important. Understand how they are calculated
* Clean data is more important than best in class models. If you have right data, simplest of the models will work. If classification, start with random forest and go ahead.
* Failing fast is important. Don’t keep on experimenting. Ask for help on process and features as soon as you can.
* Single model might not work for all test cases. Think of building individual models for different scenario and ensemble.
* Models are not black box. Make use of ELI5 and SHAP to validate your model and see how your model reacts to different rows.
* Getting pigeon holed into our thought process in very natural in ML projects. Take a break. Bounce your thoughts with peers and seniors.
* Keep excellent documentation trail of your work from day 1 (Business Requirements, Data dictionary, Validation Logics, Model Version Log etc).

Congratulation on reading this far. I hope you are walking away with invaluable insights for a real-life project. Happy reading!