**ML Churn classification project (Magnus Jensen)**

The dataset is (a supposed!) data of customer info related to if they would end up exiting the bank.There is some brief info of the different categories on the Kaggle site and it is , especially in hind sight , formulated a bit weirdly. The info for each feature is posted in one the cells in the notebook, but what I reacted to was how declarative some of the descriptions were. Like the one for surname: “Surname—the surname of a customer has no impact on their decision to leave the bank.” Why would this be said if the task is to investigate if any features have a relationship to the outcome of exited?

So we have a binary classification task. There is not much info surrounding the features. In a real world scenario, you would try to contact the bank (i.e. your client) to get more information on the features, and maybe even ask if you could get more information from them. If we are thinking of a bank. You would have lots of more information than was given in this datset. Like transactions per month. Monthly spending, if more than one user has access to an account. If the user has more than one account and if he got savings (fonds or stocks) at the bank. If he has had any violations or unpaid bills on his account, or how many of his/her bills are on auto payment, so much more.

So if this is a real situation, where a bank want to know which customers are at risk of exiting, then I don’t get why they would only select these features. Maybe it has to do with what information they can make available publicly even if the information is anonymous. Anyways, the task was to use the available features to get information of the bank and customers and see if we could predict future exiting.

**Notebooks Order**

The notebooks are meant to be read/executed in the following order:

EDA\_vis\_single  
 EDA\_vis\_multi  
 featureTransformation  
 featureSelection  
 modelTest

**Dataset Features**

The dataset arrives without any missing values nor any real outliers (at least clear outliers). It consists of a mix of categorical, numerical, and Boolean features.

A correction for the notebook. I keep talking about continuous features, but I mentioned some that are not really continuous features along side the real continuous features. The only continuous features are Balance and EstimatedSalary, but how I thought of it during the project was if they may benefit from being scaled.

I go through each feature in the notebook and look at its distribution, value\_counts() and relationships with other features, so I wont note that info here as well.

I did a fairly simple visualization. I tried pairplots but they did not show any useful information. I did use pandas\_profiling after and it just told me things I had already found out. I was thinking about trying other autoviz libraries , but that did not end up happening.

**From the notebooks**

It would be clearly ordered if I commented in order of the notebooks , but in the first (EDA\_vis\_single), I think I comment most of the important stuff.

In EDA\_vis\_multi i remove the Complaint feature since it is so heavily correlated with Exited. If this was an authentic dataset then I would make sense to keep it in if the bank had this information prior to the customer exiting the bank. Now after the fact, I should probably have dropped estimatedSalary since it does not follow any clear pattern.

It was difficult to think of the right approach when so many features seemed un-important and also that there were not lots of features. That was the reason I did not simply drop the low correlated features quickly, I wanted to see if I could learn if they would help the model anyways.

I am dropped the ball on exploring how cardType and Country related more with exited and a realized to late. Though I am pretty sure that it would not have mattered which I will get into later when I make more general evaluation of this project. Same with high numberofProducts (3 and 4) with exiting.

I also think I could have gotten more out of the gender feature. It does show up as having some effect when I look at features but the feature importance form the randomForrest model says it not significant. I could have explored this feature more.

During EDA\_vis\_multi we discover lots of patterns that seems off.

I regret not having focused feature selection before feature transformation. My thinking was that I wanted to see if I could increase the importance of some features , but instead I think may have gotten a slightly less clear picture of the importance since I did it in reverse order.

In the featureTransformation, I could perhaps have gotten more from the country encoding. I think I gave up on it a bit to soon. Since the difference between the countries would be less than if they were just ordinally encoded.

When I did the first randomForest model and feature importance ranking I should have realized salaryEstimate was probably a fake feature, though it did show up as having pretty high mutual information. I don’t know how to interpret that.

“When a feature shows low importance in a tree model but high mutual information, it implies that the feature might not be easily captured by the splitting criteria used in the tree model. The model fails to leverage the feature effectively in its decision-making process, resulting in low importance”. Could this be interpreted as estimatedSalary may be useful if transformed?

Going through the code I realized that I split df into X\_traindDF etc. TWISE! Which is not good. I should have saved to csv the first time I did it.

In featureSelection I am surprised how little HasCrCard matters. This makes me think it may also be a fake feature. Or being a measurement of something very different

I have to say that the PCA test indicates a bit that the features are still used a bit, but they don’t show up as important.

Proof reading the code , I would say that I should not have done the model testing at the end of featureSelection. I think training after using PCA was a good idea. I think there is a mistake with the last one that gets a higher score. I will get write more on this at the end since its one of the major things I did not get to.

In the featureEngineering notebook I though that some features would end up changing the model predictive power in some way , at least to make it worse. I do think the feature engineering was pretty well done on the part of combining features but I dropped the ball on the clustering. I could have done more with it but I am surprised of how long everything takes to get done. I don’t know, maybe I am just to slow…

In the modelTest notebook I think I had a reaosnble approach, but again it is quite hard to judge since I did not manage to change much when trying different pipelines and different hyperparameters. For example, I would have to change learning rate to absurd amounts for it to change the classification report for the validation set. Perhaps I did some mistake here that I did miss. I know some names are dublicates.

The reason I tried to name all gridserach objects differently was not due to being able to access them later (which is a useful thing to do and that should have been my main reason), but I have had issues with noteboosks and grid search before.

I don’t know why the notebook turned out so messy. I tried to take it one thing at a time and have a natural flow , but I don’t think I succeeded with that.

**Improvements & Mistakes**

There area lots of things that I did not have the time to get to , or realized to late , or just don’t have the understanding of the subject of machine learning yet to fully understand.

I find it pretty hard to judge how good of a job I did since it seems that my feature engineering and model tuning had very little effect on the prediction power. It could be that I the default randomForest is just very good at this particular dataset, or it could be that I have misses something important.

I am confident that my EDA was decent and that I found some anomalies and correlations , but since I don’t know if I manage to use any of those insights into improving the model it feels rather hollow.

I coud have used the gpu when using gridserachCV I would have used catboost instead since it seemed to get slightly better results compared to xb\_boost. I had planned try a voting classifier between the 3 best models but I ran out of time for that as well, same for creating a neural network.

The biggest mistake that I should have been able to implement at the end was just creating a new model and remove some of the features and see how much I could remove before the prediction power dropped. Especially since one PCA test showed great results.

I also should have explored and research more how to do feature engineering with binary features. Since you cannot do / with 0 I would had to do something else. I decided to get back to it later which was a good chose since I just barely made the deadline.

Part of me wished that I would have used dataset that needed some cleaning since then I could have shown more of my actual skills…

I know that the diving of the notebooks is a bit off , but I am not sure how to make it clearer. In theory you an divide between categories but in practice you would want to try out more how altered features effect a model and stuff like that.

I tried to comment and explain in my notebook. Some of it is gpt help comments and some of them are mine. I think missed cleaning up some of the mistakes so that the comment does not reflect what is going on.

The clustering was not done well. I thought some of them would have some effect, but they ended up not being used by any of the models. A better approach would have been to cluster some related features such as creditScore and Balance and see if that showed to have any predictive power.

I spent quite a bit of time creating dbscan clusters but then I found it that the dbscan model does not have a predict method. I did found out that there is an alternative to dbscan that can predict , but I did not have the time to tinker with it.

I think it is possible to reuse the preprocessing stage of a pipeline but with a different model but I did not manage to get it to work hence why I create so many pipelines.

I had planned to end the notebook with some illustrations but I did not manage to think of anything. Both randomForest and xgboost creates lots of trees so I am not sure how you are supposed to plot that.

To end this report I will say that I did learn some things but that it was very frustrating since I never got any sense of momentum since nothing I did seemed to improve the models. I will probably try a few Kaggle challenges in the summer to get my confidence back up.

I wish you a happy summer!