**Challenging the implicit bias: An application of linear regression with NLP for churn prediction**

100 years old corporate business operations were first built on thousands of years military learnings, where data science started first with medical needs. Although the application areas for both fields are wide-spread today, more than ever, they still reflect the nuances of their predecessors, in one way or another. Those nuances sometimes might be considered favorable, sometimes unfavorable, in other words sometimes it helps for the purpose with leading factors, sometime the opposite is more true, it is a lagging factor which also makes the objective of this paper.

Regulations perspective, financial institutions and mobile operators do have three important common characteristics. First, those industries have entry barriers, therefore they relatively have few players. Second, they run mass business, they deal both consumers and corporations and work with millions. Third, most importantly, they cannot resell their services, therefore they have to manage their customers directly.

Prediction perspective, these characteristics urge them to own a fair size of big data with real time notions. When they do, they expose full colors of estimation world, from regression to classification, form Natural Language Processing (NLP) to recommenders. In practice in mobile business world, almost all business aspects may be represented by this regard, including revenue (ARPU, Average Revenue Per User) forecasts, acquisition/churn projection, customer satisfaction analysis and product/service recommending engines.

The problem in working in this particular field comes in two. First, unlike pure positive science domain, practicality of business world limits the probability of finding a real data. In other words, more data is subject for big data with regulative liabilities in real business world, it is getting more challenging to gather, however the practicality of findings is getting more meaningful. Relevantly, more data (e.g. the ‘famous’ Iris flower dataset) is being provided by NPOs, it is getting more probable to gather, however the usability of knowledge decreases.

In specifics to mobile operators, this brings a coupling, such as a group works with mobile operators that has the best real data and exercise real problems but due to company asset restrictions, they cannot publish and another group that works with special occasion data and can publish, but the applicability diminishes its value.

From science development perspective, this duality has many hurdles against aiming higher in findings production and share for greater good. However, it does not stop stakeholders to challenge this dichotomy. Having more than 20 years ICT experience, where 10 years in mobile business world particularly and holding a PhD in buying behavior, I wanted to honor my past in this regard and scrutinized one of the most compelling dataset in the sake of challenging what is said so far.

**Dataset Discussions**

Association for Computing Machinery (ACM), serving machine learning community since late 90s, pioneers the dataset world with high practicality samples, where ACM calls it Knowledge Discovery in Databases (KDD) initiation. The term Knowledge Discovery in Databases, or KDD for short, refers to the broad process of finding knowledge in data, and emphasizes the "high-level" application of particular data mining methods. It is of interest to researchers in machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization (Fayyad et al, 1996)

One of the most famous KDD datasets (out of over 20) is known as The KDD Cup 2009. It offers the opportunity to work on large marketing databases from the French Telecom company Orange to predict the propensity of customers to switch provider (churn), buy new products or services (appetency), or buy upgrades or add-ons proposed to them to make the sale more profitable (up-selling) (KDD, 2009). To be specific, the data set consisted of 100000 instances, split randomly into equally sized training and test sets. 15000 variables were made available for prediction, out of which 260 were categorical. Most of the categorical variables, and 333 of the continuous variables had missing values.

The applications for this particular competition yielded a final fast-track performance of 0.7651 on churn, 0.8816 on appetency, and 0.9091 on up-selling. Years later, this was challenged with variations of 10% more in accuracy (Niculescu- Mizil et al, 2009).

This study is meant to scrutinize about less discussed aspects of the dataset, in particular to churn aspect, thru four (P) proposals:

1. Scrutinizing an Implicit Bias: Can LinearRegression be used for a classification problems?
2. AUC Optimization: Can decreasing precision or recall help us? Per churn’s Type-I receptive (unlike spam email detection) nature, bursting the ‘false alarm’ by trading off with false negative
3. Battle of Fuzzy Logic vs Binary Classification: In the sake of KDD’s ‘emphasizing the "high-level" application’ of datasets, promoting the utilization of probabilities instead of traditional binary classification
4. Feature Engineering with NLP: Make customer feedbacks a part of features

As a remarkable limitation, the original data is not being published anymore, demonstration was applied on a smaller version with 7.044 records and 21 labels, where the vast majority was noted as self-explanatory: customerID, gender, SeniorCitizen, PartnerDependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod MonthlyCharges, TotalCharges and Churn

**LinearRegression for Classification Problems**

The reason that LinearRegression has no history with classification is implicitly biased. Knowing that binary decisions of all classification models are based on probabilities, there is no technical showstoppers from LinearRegression from being applied for probabilities. As long as the regression predictions are between 0 and 1, results can be utilized to make binary decisions. Number one reason that it was not practiced was not only it was thought as a non-classifier, but also it is tacitly perceived as the least trivial predictor model. On the other hand, although it is doable by all means, in other words binary decision points is a possibility for also regressors, they may not always useful as thought, that makes an incremental discussion point that was entirely handled within the P3 below.

**AUC Optimization**

Lift scores might help with segmentation for better confidence levels, however it is not always helpful. For mass mailing operations, companies may want to choose the 10% with the best accuracy among 10Million+ customers, so they would decrease the budget they need and increase the efficiency. However, unlike segmentation for mass emails, churn job prediction cannot do that. If they do, they will jeopardize the royalty management basics, since they would need to ignore the vast majority of data.

For optimization purposes, manipulating the AUC might yield some useful combinations. However, traditional approach with medical applications would not have a luxury for a secondary decision point. They cannot call someone sick who is really not and the opposite is also true. One of the differentiators of classification comparing to regressors domain is that different problems may require polarized decisions. AUC is known with 4 typologies, while favoring true positive is the ultimate goal, decisions (except medical derivatives which favors F1) on second preferences are also significantly important. As shown in Figure 1 below, churn job may accept false positives as the secondary choice, where spam job would do the opposite, the false negatives as the second choice.

Figure 1: Churn scenario (on the left) vs Spam scenario (on the right)

Churn scenario would accept false alarms while decreasing the false negatives, since treating no-churn customers as churn customers would not hurt company. The only risk they took is spending a bit more on royalty that they normally need, however they mitigate the risk of treating a churn customer as no-churn customer, so they decrease the precision but increase the knowledge they need.

To coupling the situation, spam scenario would accept false negatives while decreasing the false alarms, since treating spam customers as no-spam customers would not hurt company. The only risk they took is having a bit more emails that they normally should have, however they mitigate the risk of treating a no-spam email as a spam email, so they decrease the recall but increase the knowledge they need.

Summarizing our findings, churn applications would prefer false alarms comparing to false negatives, in other words they rather stick Type I comparing with Type II. Doing that would decrease the precision but surprisingly increase the useful knowledge, as favored by KDD.

**Battle of Fuzzy Logic and Binary Classification**

Traditional practices with medical applications need to make decisions where the particular decision is usually about initiating a treatment or not. One of the reasons is that there is usually no way to apply a partial treatment. However, this is not the case for most of the business applications. The graduality is being very well received on most occasions, since there are always different options to offer. From this perspective, even if we apply the classification, unlike the norm, binary form is not necessarily the most useful one most of the time, probabilities might make better fit.

In this approach, the necessity to make decisions about secondary options with AUC manipulation would be dissolved. Since the value of categorization is getting diminished. In other words, 51% probability does not have to make 1 and it does not have to be equally treated with 99%. When we open this door, the churn problem can even be a part of regressors’ world and increase our modelling selections.

**Feature Engineering with NLP**

Traditionally, customer records are usually not graded and they do not make a part of mobile operators’ prediction labels. Surprisingly, although having over thousand labels, this is also true for The Orange Lab dataset that KDD distributed. Knowing that predicting churn is the ultimate fruit of customer satisfaction and this may be a type of data that already exist within the organization, it is notable that this juxtaposition has not been addressed properly, therefore churn prediction has an implicit biased for less successful estimations.

Today, customer records come in two, calls and emails, where the former can be transformed to the latter with an ease, thru common audio-to-text libraries. Even with primitive features with text mining, a compound Vader score can be generated for each customer and can be make a part of the labels. This effort might lead to better accuracies. Probabilities may have a place on customer management screens, call center agent may want to know the churn probability, instead of manual flags or binary classification, so they can offer what was designated accordingly.

Besides the four types of revenue dollar derivatives, the vast majority of labels was categorical, followed by 6 binary typologies, like the target ‘Churn’ column. Due to categorical dominance, Phik correlation was applied thru a notably powerful pandas profiles library. As shown in Figure 2, significant centric interdependencies of revenue labels, as well as tenure and MonthlyCharges were noted.

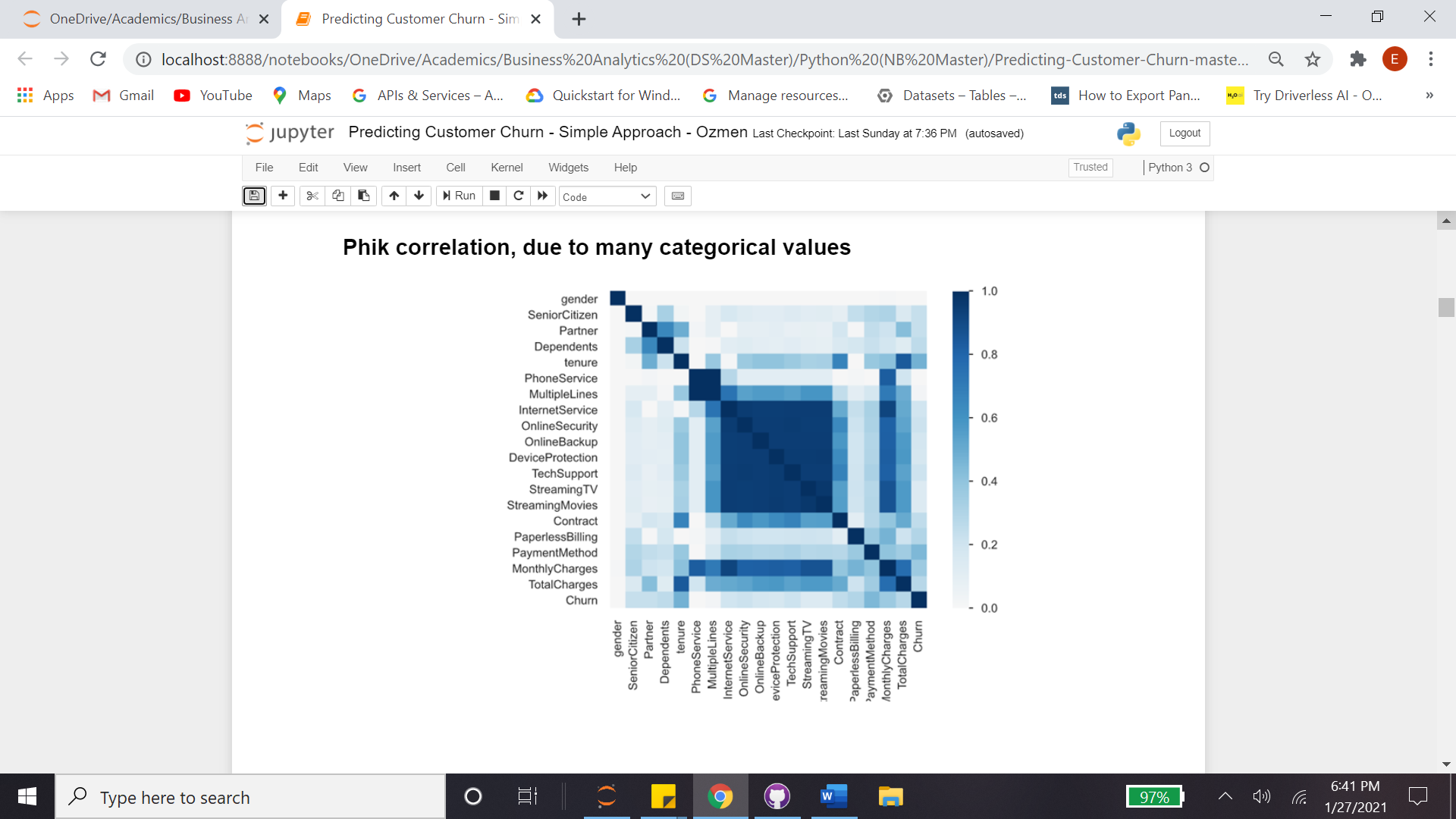
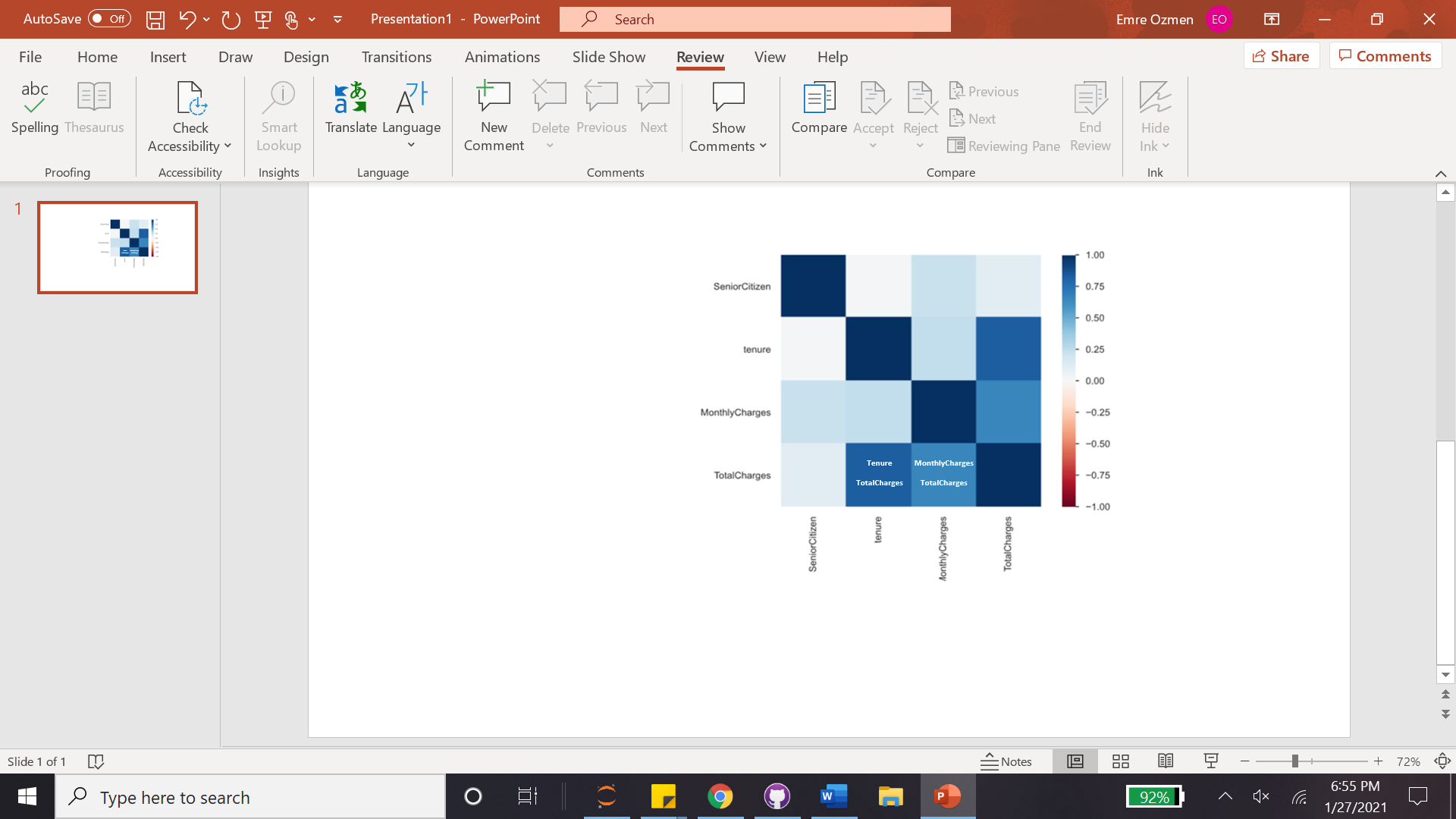
 

Figure 2: Density Phik (on the left) vs Summary Phik (on the right)

Early descriptive findings showed that correlated labels had polarized densities over the ‘Churn’ As shown in Figure 3, highly paying customers were more sensitive to churn, where it implicitly refers to higher maintenance customers with high expectancies. On the other hand, longer a customer stays with the company less likely churns. This coupling raised the dilemma of high tenure customers tend to spend less. In other words, due to the cumulative nature of the latter tenure and total charges significantly were correlated, however this may not be the case in the churn specifics.

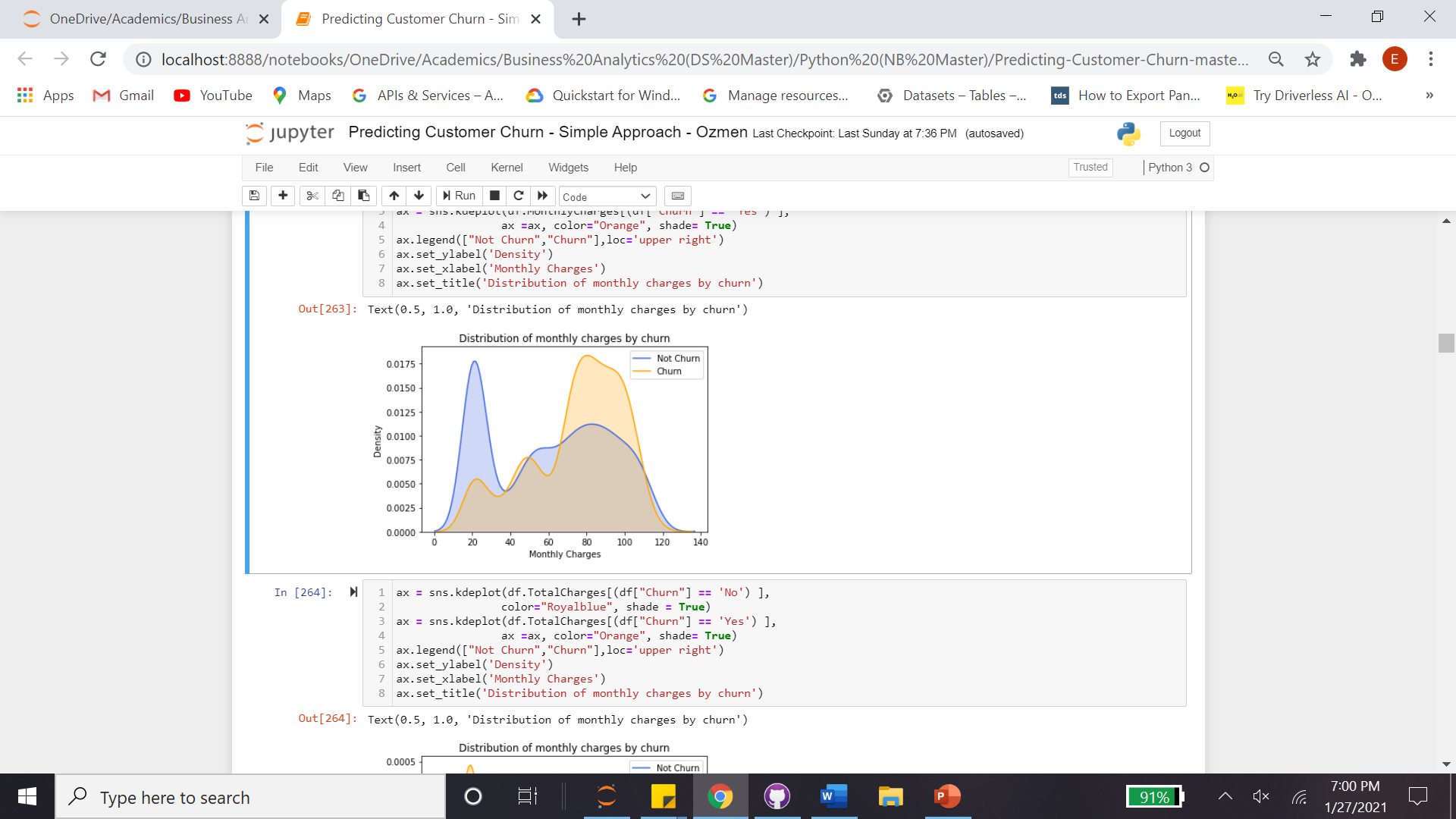
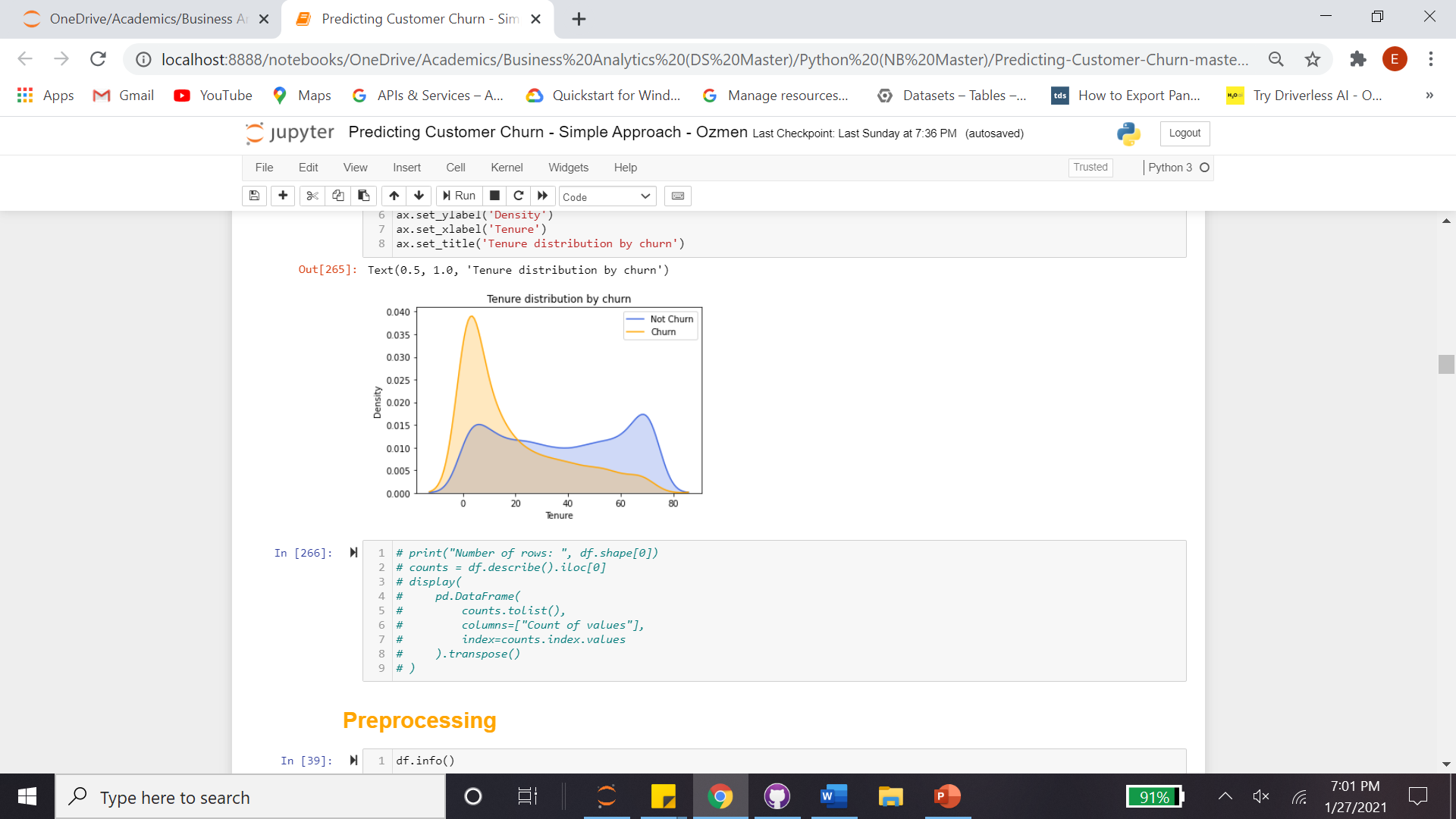
 

Figure 3: Churn by Monthly Charges (on the left) vs Churn by Tenure (on the right)

Preprocessing perspective, both get\_dummies and OneHotEncoder were practiced. For numeric figures, both StandardScaler and MinMaxScaler were applied. Retrospectively, all combinations yielded similar results. Data was split as 80% and 20% for training and test partitions respectively. Cross validation efforts were parked for the optimization stage in the end.

As shown the first bullet below, winning RandomForestClassifier, XGBClassifier and GradientBoostingClassifier were pinned for fast track, where the Gradient Boosting led with 0.80 accuracy, similar to competition champion’s score (Niculescu-Mizil, 2009). The second bullet refers to an underdog, LinearRegression, however surprisingly produced the best result, better than winning model, with 0.81 accuracy. More importantly, it produced a better recall result, 10% more than Gradient Boost, with 0.52. It is notable that none of competition participants practiced LinearRegression, due to classification dichotomy of the problem that we extensively discussed earlier. The third bullet honors the best model, LinearRegression, with AUC optimization favoring better recall by trading off the precision. Doing that, as shown in Figure 4, the number of wrong predictions (280) were not exploited. To be precise, 198/98 was respected as 172/98 by 0.45 tolerance, in other words accuracy was not jeopardized with smaller tolerances, although they will yield better recall figures that we aim. Efforts yield another 10% more comparing to its predecessor with 0.58 recall score. It is also notable that once regressors produce between 0 and 1 both RMSE and Accuracy are mentionable simultaneously. To brief findings:

* GB Classifier, 0.50 Tolerance | Accuracy = 0.80 | Precision = 0.68 | Recall = 0.47
* Lin Regressor, 0.5 Tolerance (RMSE = 0.44) | Accuracy 0.81 | Precision = 0.65 | Recall = 0.52
* Lin Regressor, 0.45 Tolerance (RMSE = 0.45) | Accuracy 0.80 | Precision = 0.61 | Recall = 0.58

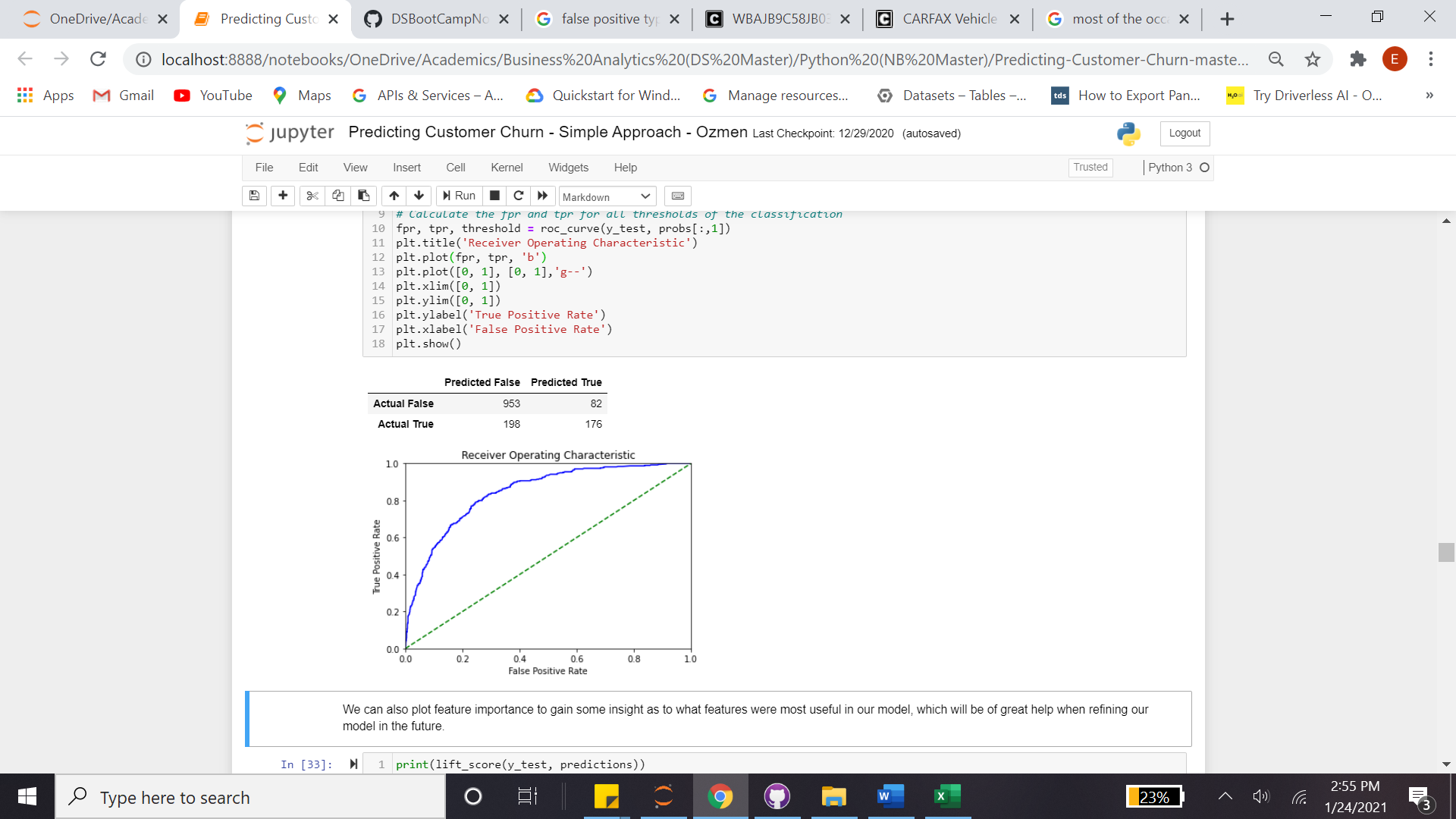
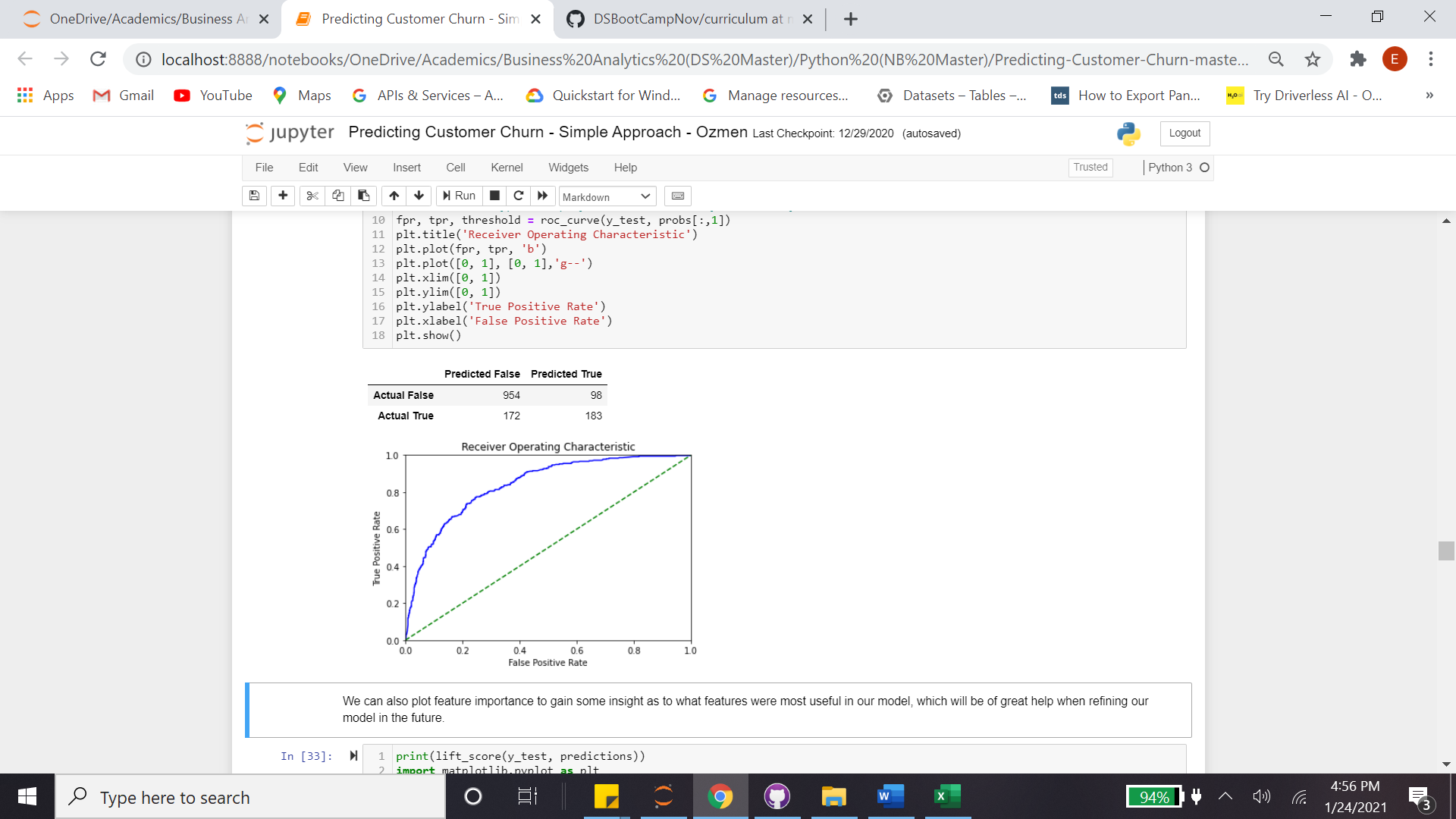
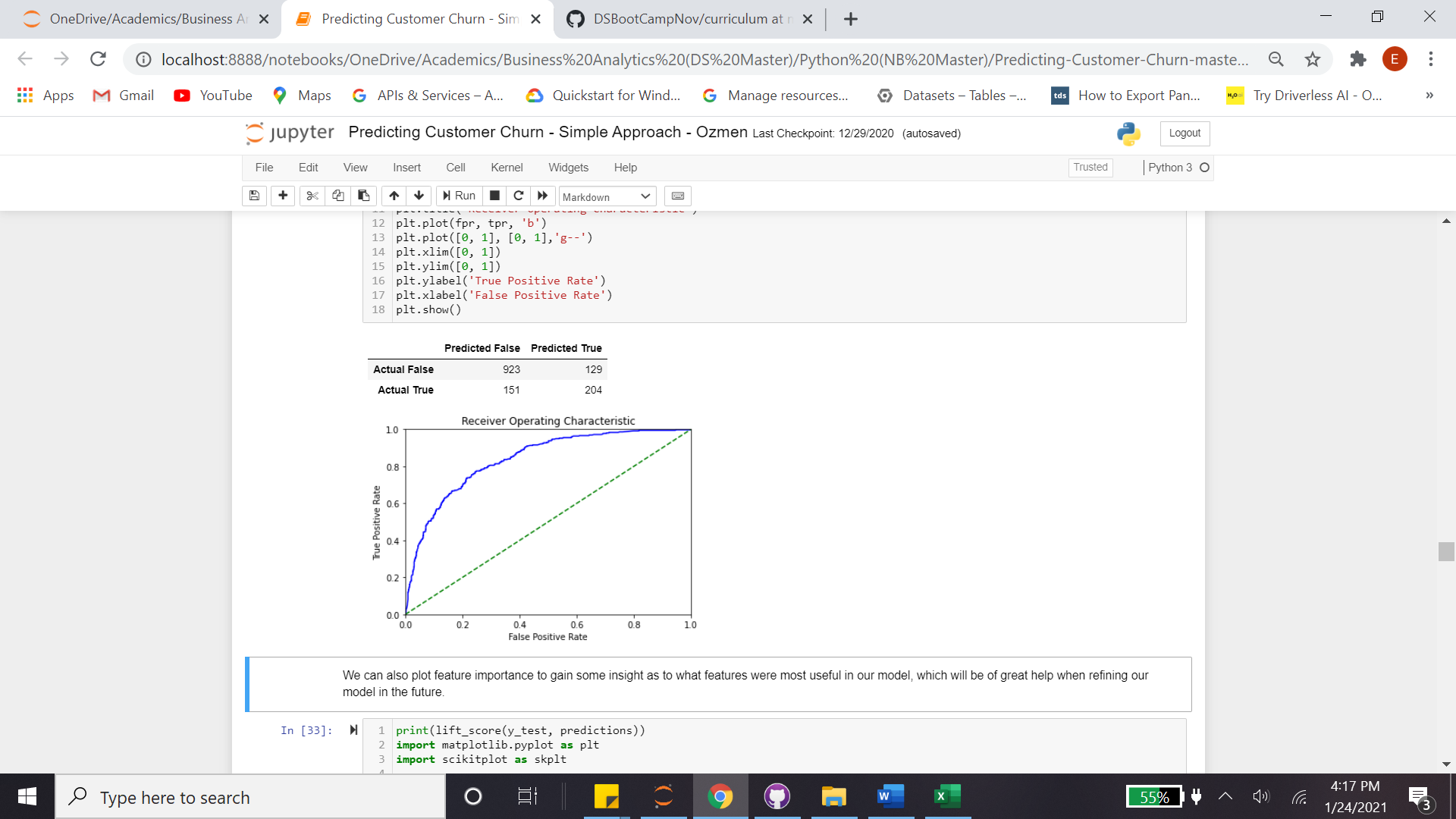
  

Figure 4: 1st Winning Model (on the left) vs 2nd Winning Model (center) vs AUC Opt (on the right)

Practicality perspective, around 50 people switched from missed churn numbers to false alarm, where the latter is preferable. Also, around 30 people transferred to churn numbers, so they can luckily be over-treated. In other words, model user will have around 25% less missed churn numbers and around 60% more false alarm (bearable rounding error) with around 20% penalized (safer from royalty/churn management stand point) in other words more churn numbers.

**Feature Engineering with Text Mining**

Text mining has a reach history. Its roots can be traced back to early library work efforts of first established universities. In years, the variety of its capabilities enlarged form summarization to information extraction/discovery, clustering, context/topic meaning and deep discovery, e.g. sentiments, catching idioms and innuendo. Identify entities and emotions in a sentence and use these to determine if the entity is being viewed positively or negatively (ADS, 2020). Often, looking at words alone is not enough to figure out the sentiment, which also makes text mining complex.

Sentiment analysis is generally a starting point in analyzing a text and is then coupled with other techniques like topic analysis. Sentiment analysis is usually done using a corpus of positive and negative words, in other words some sources compile lists of positive and negative words. Others include the polarity -the degree of positivity or negativity- of each word.

Miner’s algorithm distinguishes the approach into two parts, understanding per sentences and words. Knowing that they are not rivals, as a matter support each other, avenues to utilize both paths worth to explore. From document perspective, study falls into clustering area. From words perspective, it is more associated with Natural Language Processing (NLP).

**Vader sentiment in theory**

Per its official nltk.org/api/nltk.sentiment.html document, a SentimentAnalyzer is a tool to implement and facilitate Sentiment Analysis tasks using NLTK features and classifiers, especially for teaching and demonstrative purposes. In other words, it is weighted word analysis using Vader. Vader contains a list of 7500 features weighted by how positive or negative they are. It uses these features to calculate stats on how positive, negative and neutral a passage is. It combines these results to give a compound sentiment (higher = more positive) for the passage.

Human trained on twitter data and generally considered good for informal communication. 10 humans rated each feature in each tweet in context from -4 to +4.

* Calculates the sentiment in a sentence using word order analysis
* "Marginally good" will get a lower positive score than "Extremely good"
* Computes a "compound" score based on heuristics (between -1 and +1)
* Includes sentiment of emoticons, punctuation, and other 'social media' lexicon elements

**Vader sentiment in practice**

Within this study, three social media listening were made in the sake of customer satisfaction. The average was added as a new feature under column name ‘Compound’:

* Twitter.Com: 2,500 latest tweets with compound score varying between -0.9 and 0.6
* Trustpilot.Com: 1,500 records with compound score varying between -0.7 and 0.5
* ConsumerAffairs.Com: 1,000 records with compound score varying between -0.8 and 0.5

**Repeating process**

To make it comparable, RandomForestClassifier, XGBClassifier and GradientBoostingClassifier were first pinned for the first run, where this time the XGB Classifier led with 0.89 accuracy and 0.58 recall scores. The second bullet refers to LinearRegression, which again surprisingly produced the best result, better than winning model, slightly compromising with 0.86 accuracy, however excelling with 0.61 recall. The third bullet honors the best model, LinearRegression, with AUC optimization favoring better recall by trading off the precision. Doing that, as shown in Figure 5, the number of wrong predictions were not exploited by smaller tolerances in the sake of further recall improvements. Efforts yield almost 15% more comparing to its predecessor. To brief findings:

* XGB Classifier, 0.50 Tolerance | Accuracy = 0.89 | Precision = 0.94 | Recall = 0.58
* Lin Regressor, 0.5 Tolerance (RMSE = 0.38) | Accuracy = 0.86 | Precision = 0.79 | Recall = 0.61
* Lin Regressor, 0.45 Tolerance (RMSE = 0.37) | Accuracy = 0.86 | Precision = 0.73 | Recall = 0.70

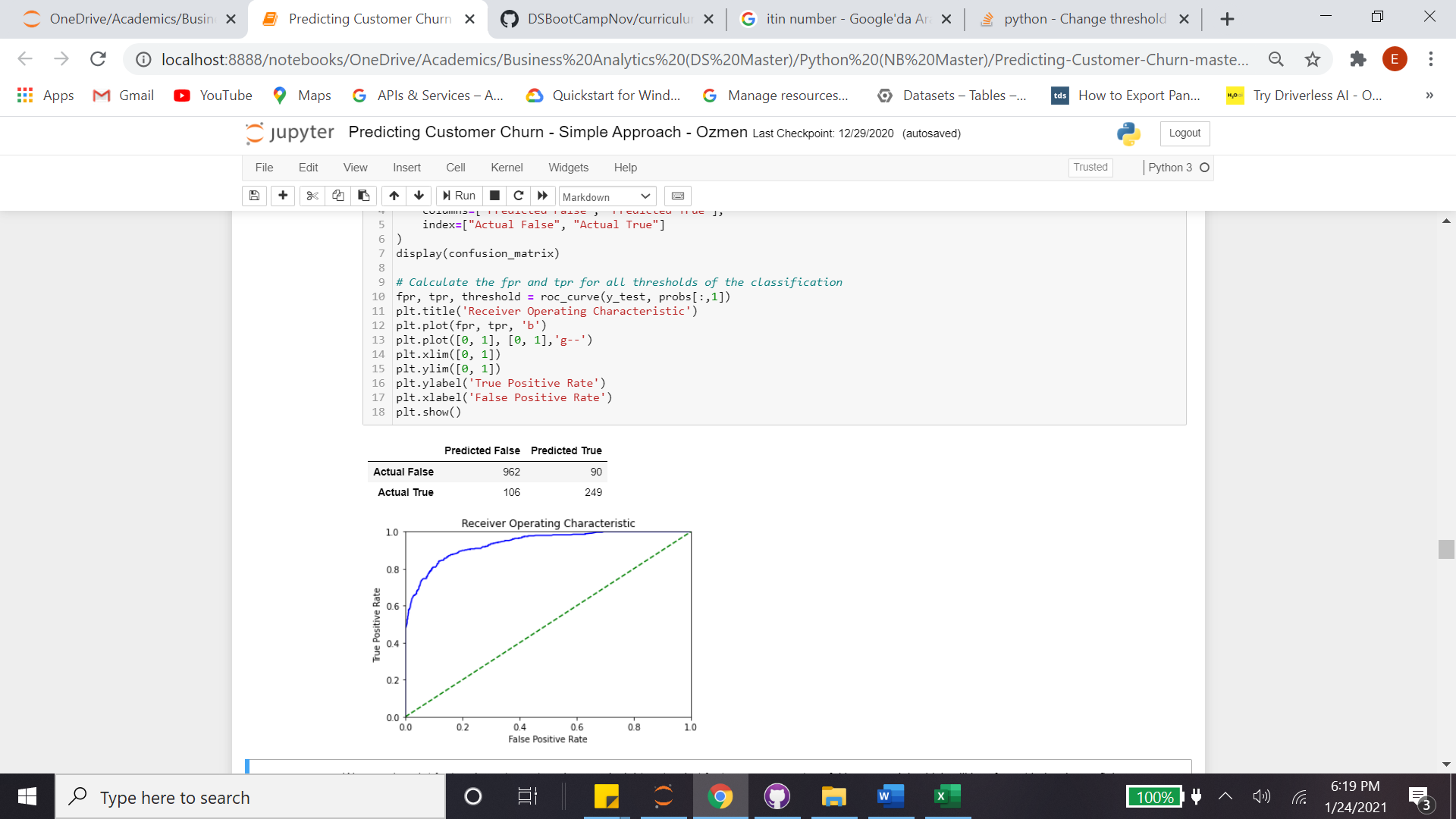
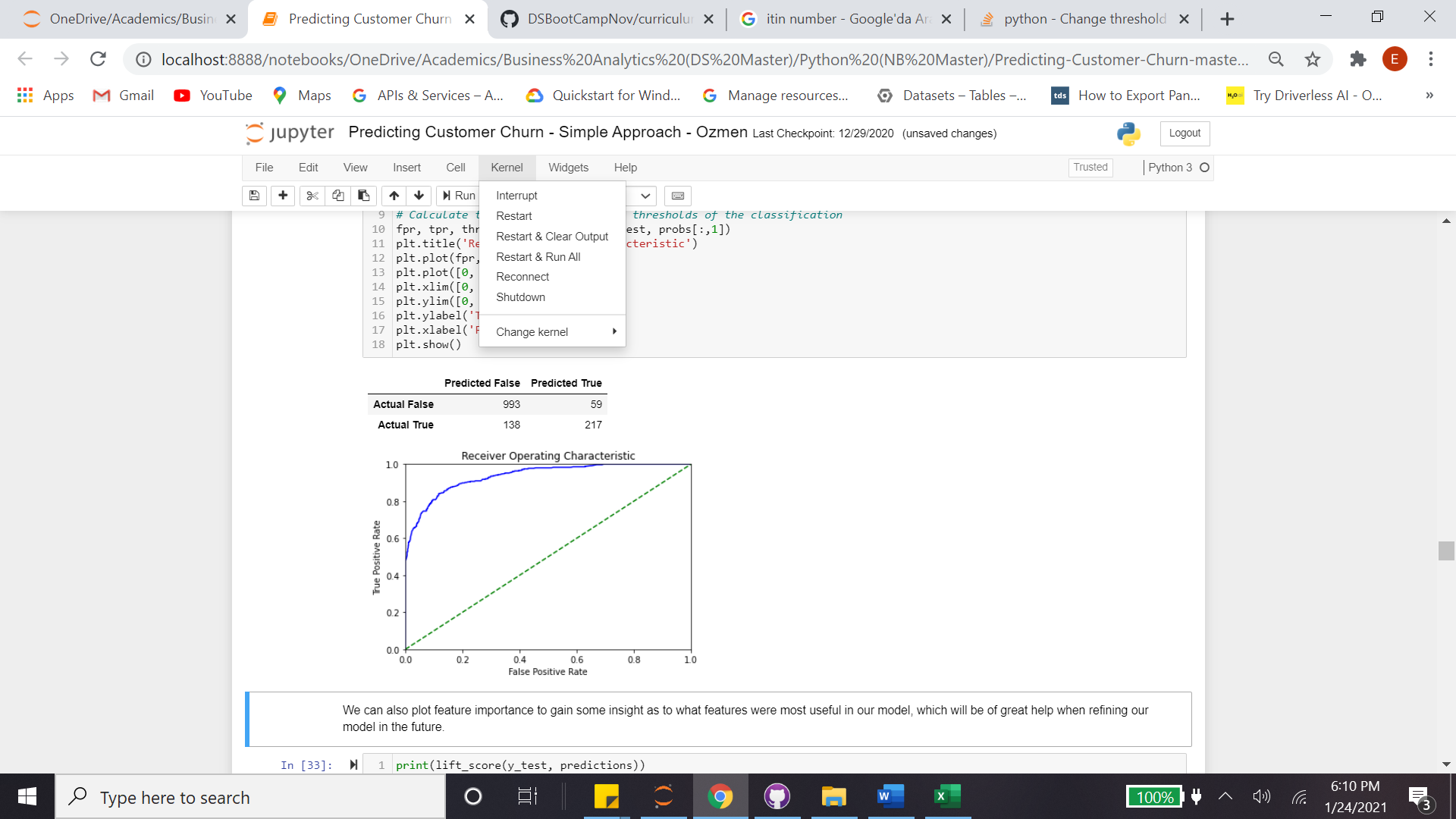
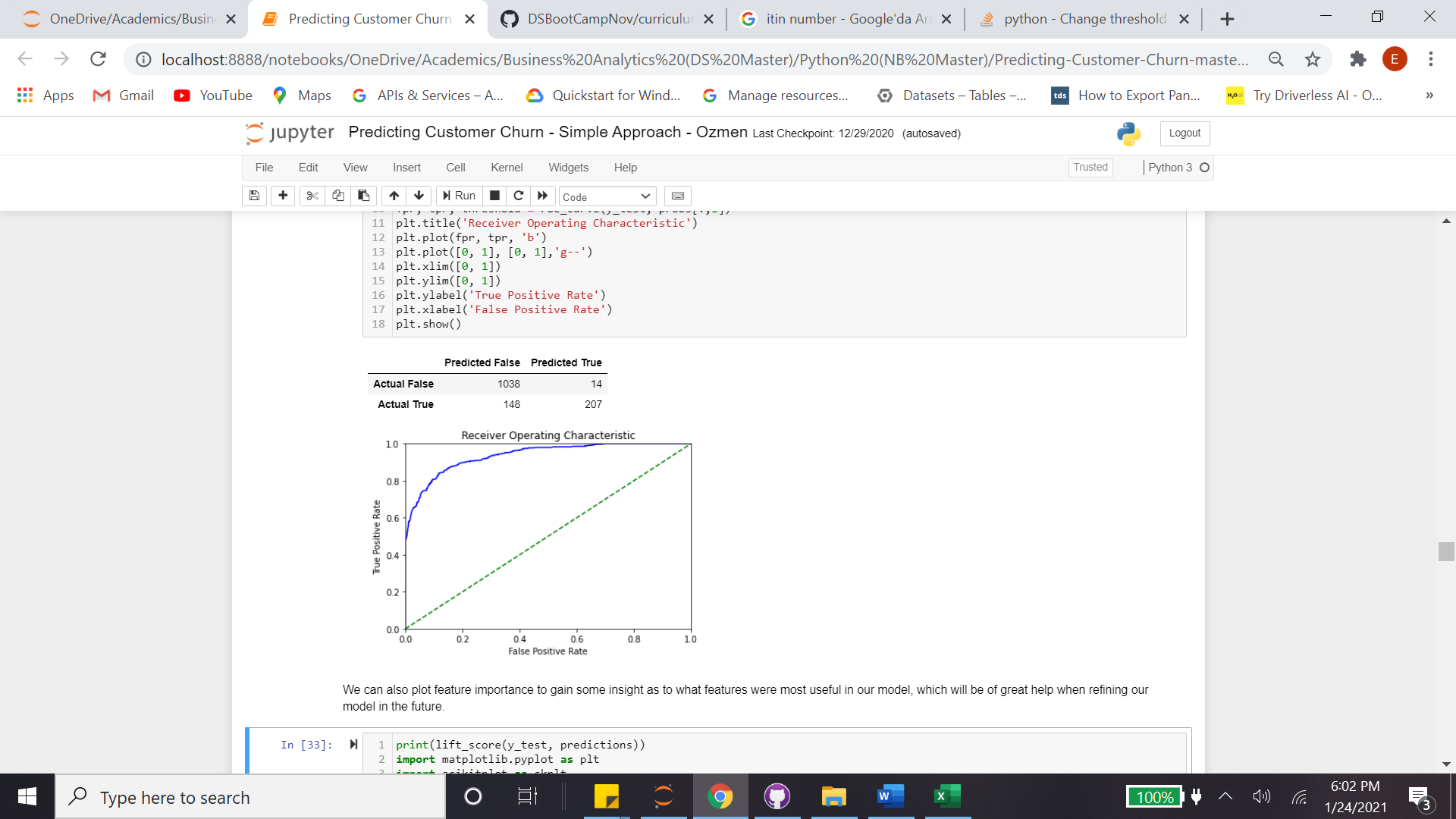


Figure 5: 1st Winning Model (on the left) vs 2nd Winning Model (center) vs AUC Opt (on the right)

Practicality perspective, around 75 people switched from missed churn numbers to false alarm, where the latter is preferable. Also, around 40 people transferred to churn numbers, so they can luckily be over-treated. In other words, model user will have around 30% less missed churn numbers and around 5 times more false alarm with around 20% penalized (safer from royalty/churn management stand point) in other words more churn numbers.

**Comparing to the very first traditional approach**

Traditional -‘so-called’ the winning path- approach that ignores both linear regression and recall performance looks like having the worst results per churn estimations and also unbearable missing churn numbers. Adding the incremental predictive power of customers’ NLP, there are two very significant figures that we can gather in brief:

* Missed churn numbers decreased around 50%, from 196 to 106 customers
* Re-estimated churn numbers were increased around 50%, from 176 to 249
* Technicality stand point, the whole tripartite was augmented, from 8% to 50%
  + Accuracy = 0.80 | Precision = 0.68 | Recall = 0.47 (Traditional “winning” approach)
  + Accuracy = 0.86 | Precision = 0.73 | Recall = 0.70 (Nonconformist approach)

Per the latest best performed model, important factors were drafted as shown in Figure 6, where ‘Compound’ score was dominated, followed by Contract\_Month\_to\_month with 0.17 scaled importance.

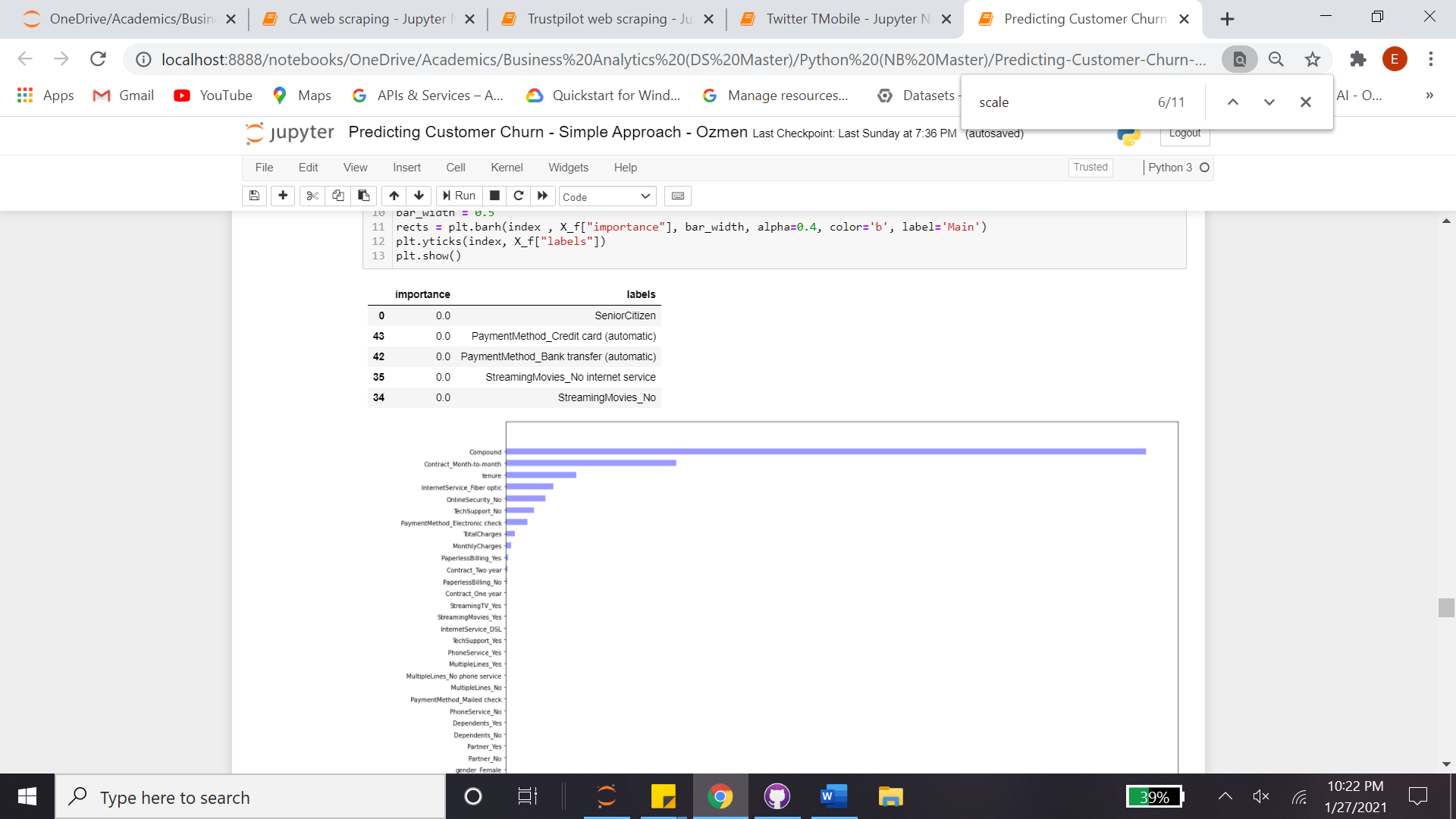


Figure 6: Important Factors

**AML (Auto Machine Learning) Optimizations**

Automated machine learning (AML) concept has a growing reputation with not only having an optimized omni-model environment, but also validation (what was gathered) standpoint (Olson et al, 2016). AML works with many models including Gradient Boosts, Naïve Bayes, Trees, Random Forest, Linear Regression, Gradient Descent Classifier, Logistic Models, Multinomials, as well as Support Vector Machine derivatives (H2O, 2017). It adjusts default parameters in a way to find the best split thru cross validation, finds best algorithms, optimize the entire workflow, except the decision points about how to trade off. There are many open-source attempts in respect to AML, in both open source and commercial arena, to mention few for the former:

* auto-Weka is a Java library, built on Weka
* auto-sklearn is a Python library, optimizes per Bayesian
* TPOT works with Python
* auto-keras is a Python library, has very powerful classification/regression models for not only structured data, but also images and texts
* H20 AutoML is developed with Java, works with Python, R and Scala

Amongst all, H20 AutoML has distinctive features per three aspects, it is explicit in terms of model names (and flexible in terms of inclusions or exclusions), gives confusion matrix if it applies and proposes important factors (LeDell, 2018). To be more specific, H20 AutoML requires only two data and two stopping parameters. On the other hand, it handles a total of 27 parameters to burst the control on user hands (Miner et al, 2012).

Comparability perspective the exact same training and test split was applied with 80% and 20% respectively, however, this time cross validation was added into the process. K-fold 5 was generated accuracies varying between 0.86 and 0.90. To mitigate the overfitting odds, its mode (0.89) was accepted. Comparing to last model, the findings can be listed as follows:

* GB Classifier, 0.43 Tolerance (RMSE = 0.29) | Accuracy = 0.89 | Precision = 0.77 | Recall = 0.78

Table 1: Confusion Matrix with AML Optimization



* Missed churn numbers decreased around 20%, from 106 to 86 customers
* Re-estimated churn numbers were increased around 10%, from 249 to 274
* Technicality standpoint, the whole tripartite was augmented, up to 50%
  + Accuracy = 0.89 | Precision = 0.73 | Recall = 0.70
  + Accuracy = 0.89 | Precision = 0.77 | Recall = 0.78

We earlier examined that gains/lift table do not produce useful knowledge for churn prediction and it does not. Scores were generated for testing purpose, it stated average response rate as 25.36 % and average score as 26.48 %. 15 partitions were led by 99% AUC, 7% more comparing to 93% AUC average.

**Lift Optimizations**

Mega mobile operators with more than 100 Million subscribers will statistically have over 1 Million data subject to churn and may want to work with up to deciles (10-quantiles) data, as shown Figure 7.

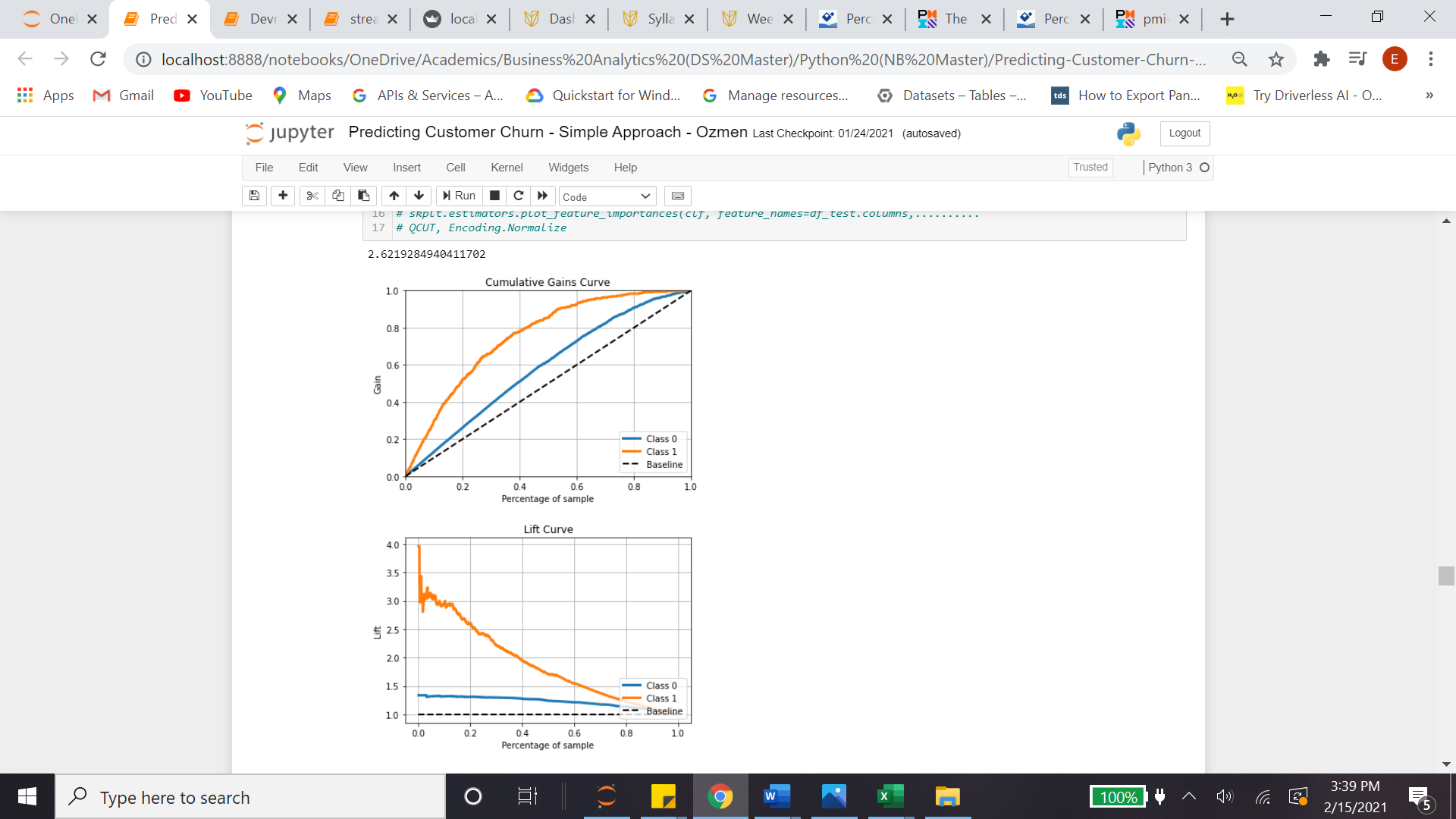
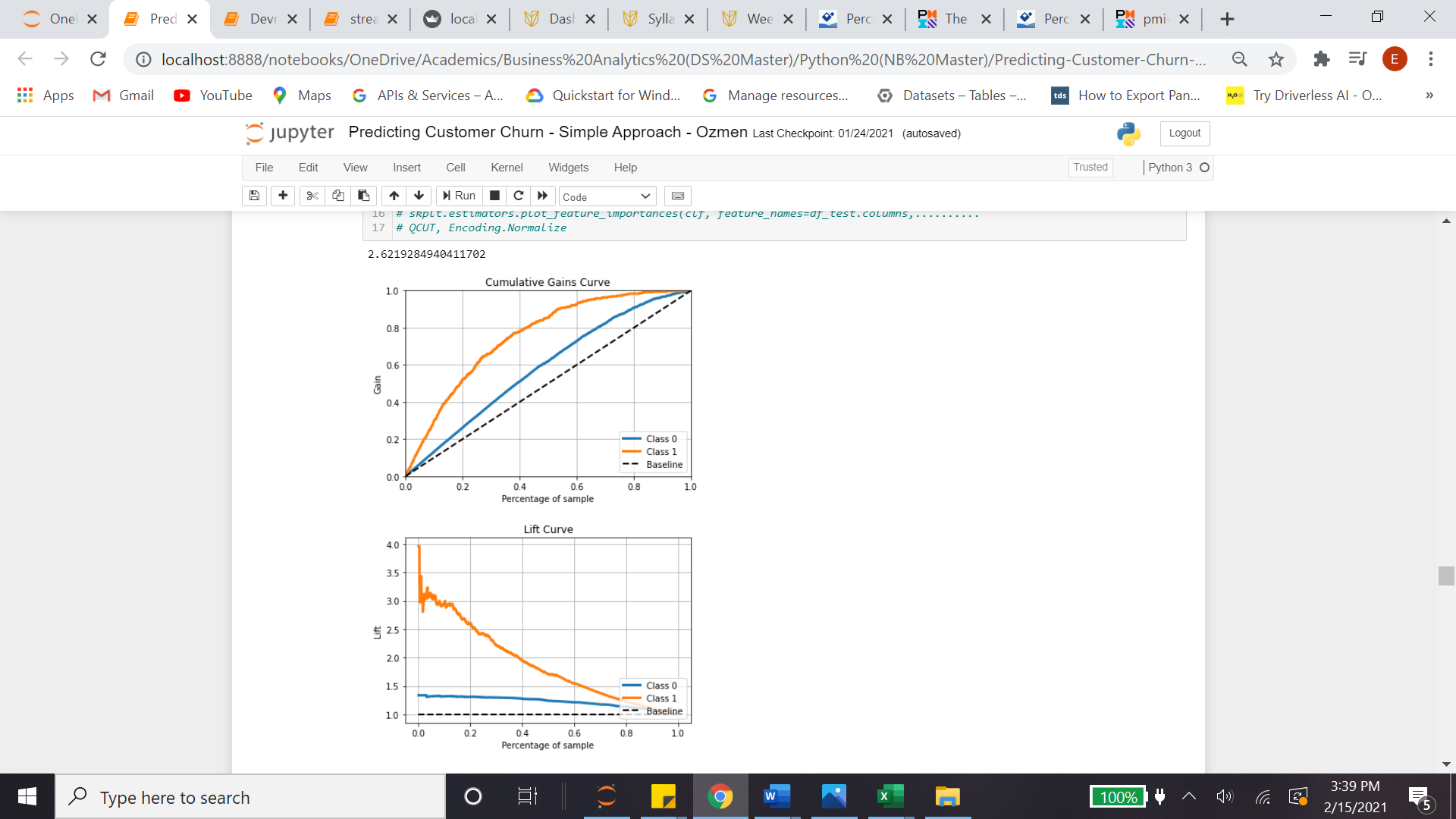
 

Figure 7: Curves for Gain and Lift

**Discussions**

This study yields few frontier ideas thru placing inceptions within typical stages of churn prediction. Four proposals were generated and applied respectively and the winning model was challenged with double digit improvement in each aspect of the classification performance tripartite, namely accuracy, precision and recall. Proposals and outcomes can be revisited as:

* First, the implicit bias was scrutinized, classification problems -in the sake of churn prediction- were challenged with regressors. It was found that regressors might make an option and can also be a leader model. 🡪 11% more recall performance with 1 point more in accuracy.
* Second AUC optimization was revisited. Per churn’s Type-I receptive (unlike spam email detection) nature, bursting the ‘false alarm’ by trading off with false negative truly helped. 45% tolerance resulted 12% more recall performance with no compromising with accuracy.
* Per the "high-level" application’ of datasets, as KDD promotes, utilization of probabilities instead of traditional binary classification may help call center agents, customer satisfaction champions with real time decision making on the phone.
* Although feature engineering with NLP is not an unknown, practitioners perspective it is a rare event. Making customer feedbacks a part of features was attempted thru Vader sentiment dynamics. This is different and more useful than knowing who will churn in binary form, since 0.51 cannot make 0.99. Having a large spectrum of product and services that can address a variation of compartments for probabilities between 0 and 1, five-to-seven categories can be created with ease, so agents may help customers with full colors. Royalty management perspective, this is a priceless fruit that can be gathered.

Churn prediction is not survival. There are degrees in nature of this business question and they all are important. In other words, churn prediction needs more than binary decisions. This churn prediction journey’s findings can be summarized in four hypotheses:

1. Being more receptive in model bias, including regressors, validates the churn model
2. Favoring recall performance improves the churn model
3. Respecting graduality (probabilities rather than 0 and 1) in classifiers improves the churn model
4. Working with customer feedback for predicting customer churn improves the churn model

Practicality perspective an app was presented thru Streamlit library, where only important factors were demonstrated on Figure 8.

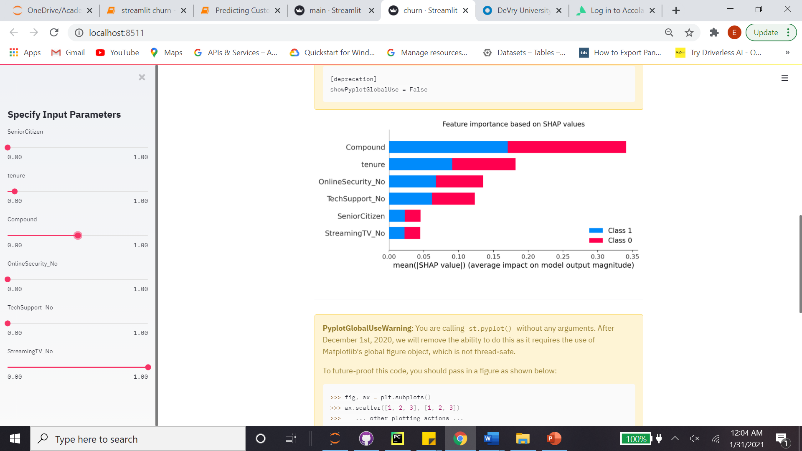
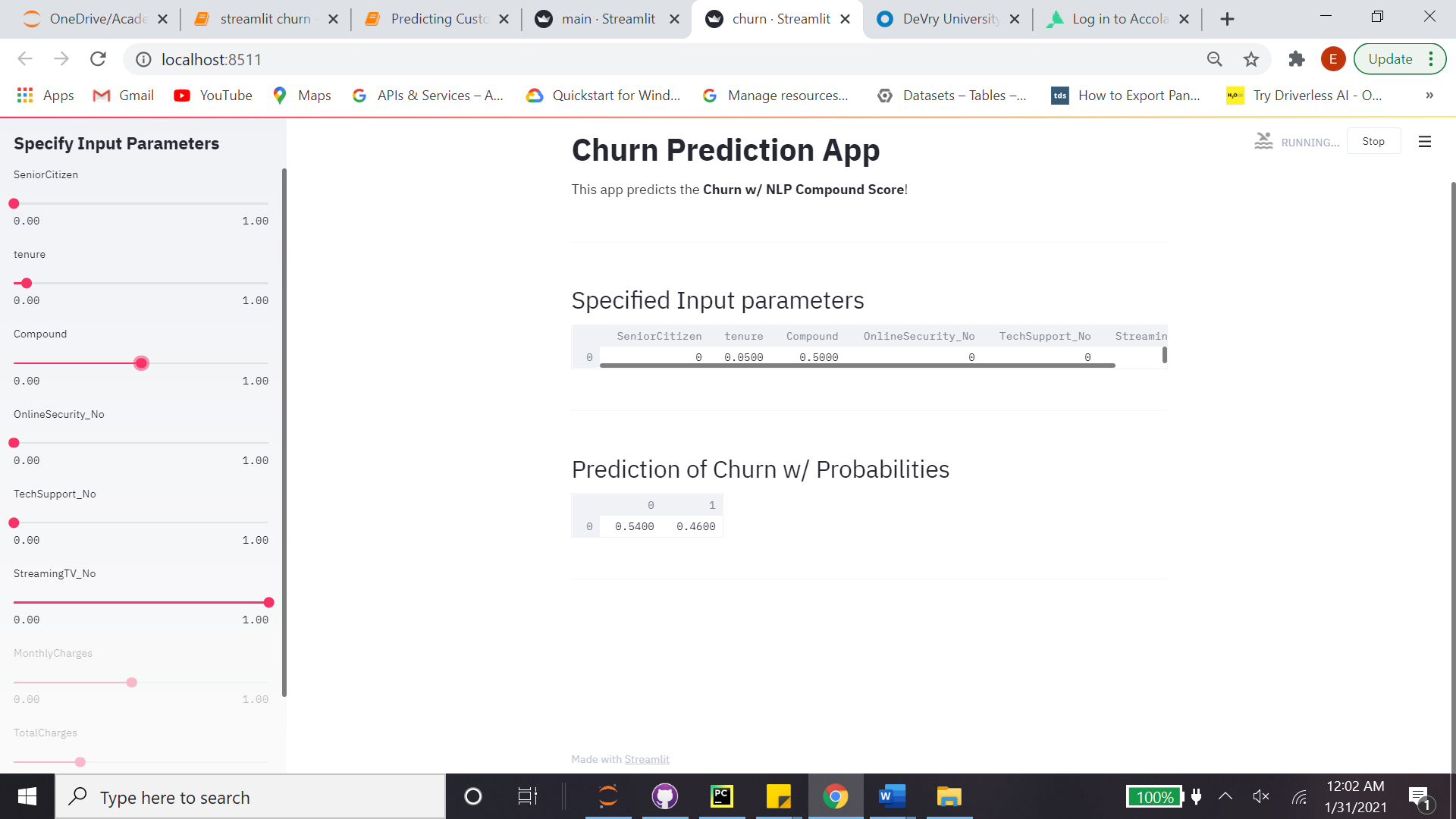
 

Figure 8: Probability based Churn Predictor in Production with Streamlit

Future studies can be directed in both more applications and intrusions in triviality of linear regressions for churn prediction.

**References**

Fayyad, Piatetsky-Shapiro, Smyth, "From Data Mining to Knowledge Discovery: An Overview", in Fayyad, Piatetsky-Shapiro, Smyth, Uthurusamy, Advances in Knowledge Discovery and Data Mining, AAAI Press / The MIT Press, Menlo Park, CA, 1996, pp.1-34

KDD (2018). KDD Cup 2009: Customer relationship prediction, From <https://www.kdd.org/kdd-cup/view/kdd-cup-2009>

Niculescu-Mizil, A., Perlich, C., Swirszcz, G., Sindhwani, V., Liu, Y., Melville, P., Wang, D., Xiao, J., Hu, J., Singh, M., Shang, W., Zhu, Y.: Winning the KDD cup orange challenge with ensemble selection. The 2009 Knowledge Discovery in Data Competition pp. 23–34 (2009)

Olson, R. S., Bartley, N., Urbanowicz, R. J., & Moore, J. H. (2016, July). Evaluation of a tree-based pipeline optimization tool for automating data science. In Proceedings of the Genetic and Evolutionary Computation Conference 2016 (pp. 485-492).

H2O.ai, H2O AutoML. (2017). From [http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html. H2O version 3.30.0.1](http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html.%20H2O%20version%203.30.0.1).

LeDell, E. (2018). The different flavors of AutoML. From <https://www.h2o.ai/blog/the-different-flavors-of-automl/>

Miner, G., Delen, D., Elder, J., Fast, A., Hill, T., Nisbet, R. (2012). The seven practice areas of text analytics. In Practical text mining and statistical analysis for non-structured text data applications (pp. 29–41). Amsterdam: Elsevier, Inc.

Applied Data Science, ADS, Text Mining, Page 1, Columbia University Engineering School & Emeritus