*Avrupa Bilim ve Teknoloji Dergisi*
*Sayı 31 (Ek Sayı 1), S. 399-408, Aralık 2021*
*© Telif hakkı EJOSAT'a aittir*
***Araştırma Makalesi***

www.ejosat.com ISSN:2148-2683

*European Journal of Science and Technology No. 31 (Supp. 1), pp. 399-408, December 2021*
*Copyright © 2021 EJOSAT*
***Research Article***

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

Emre Ozmen[1*]

[1*] Nişantaşı University, Institute of Graduate Studies, İstanbul, Turkey, (ORCID: 0000-0001-5541-1155), emre.ozmen@nisantasi.edu.tr

(First received 29 September 2021 and in final form 22 December 2021)

(**DOI:** 10.31590/ejosat.1002211)

**ATIF/REFERENCE:** Ozmen, E. S. (2021). Challenging the implicit bias: An application of linear regression with NLP for churn prediction. *European Journal of Science and Technology*, (31), 399-408.

## Abstract

The problems with working in churn prediction are twofold. First, unlike pure science, the practical applications of data in the business world limit the probability of collecting real data—that is, more data is subject to big data, more regulative liabilities in the real business world occur. These results in data collection becoming more challenging despite the increased practicality of the findings. Despite a limited version of KDD (Knowledge Discovery and Data Mining, as an Association of Computing Machinery initiation) competition data, this study introduces unique ideas by placing inceptions within typical stages of churn prediction. As part of this study, four proposals were generated and applied, and the winning model was challenged with double-digit improvement in each aspect of the classification performance trio—namely accuracy, precision, and recall, where favoring the latter the most. Proposals can be summarized as validations with regressors, recall-biased metrics, probability-favoring optimizations and customer sentiments-empowered results.

**Keywords:** Churn predictions, NLP optimizations, Recall-biased systems, Linear regression as classifier

# Lineer regresyon ile NLP uygulamasının müşteri kaybı analizine adaptasyonu

## Öz

Müşteri kaybı tahmini çalışmanın önünde muhtelif engeller vardır. Birincisi, salt pozitif bilim alanından farklı olarak, iş dünyasının doğası gerçek bir veri bulma olasılığını sınırlamaktadır. Başka bir deyişle iş dünyasında daha fazla veri düzenleyici bulunmakta, yükümlülükler paylaşılmasını giderek zorlaştırmakta ama buna mukabil bulguların pratikliği daha anlamlı hale gelmektedir. Bu makale ile, KDD (Bilgi İşleme Derneği tarafından yönetilen, Bilgi Keşfi ve Veri Madenciliği oluşumu) yarışma verisinin sınırlı bir versyonu ile çalışılmasına rağmen, dört öneri oluşturulmuş ve uygulamaları sergilenmiştir. Öneriler, regresörle doğrulamalar, hatırlamayı destekleyen metrikler, olasılık lehine optimizasyonlar ve müşteri yorumları ile güçlendirilmiş sonuçlar olarak özetlenebilir. Kazanan model, hatırlamaya odaklanmasına rağmen, sınıflandırma performans üçlüsünün her birinde, doğruluk, kesinlik ve hatırlamada çift haneli iyileştirme sağlamıştır.

**Anahtar Kelimeler:** Kayıp tahminleri, NLP optimizasyonları, Hatırlama odaklı metrikler, Sınıflandırıcı olarak regresyon

[*] Corresponding Author: emre.ozmen@nisantasi.edu.tr

# 1. Introduction

The foundations of centuries-old business operations lie in extensive military learnings and medical needs, which were instrumental in launching data science. Both fields have widespread applications  regarding data today, and continue to reflect the nuances of their predecessors, which may be favorable or unfavorable; these nuances might aid the application of data by providing leading factors, or hinder the process by introducing a lagging factor. The objective of this paper is to introduce a lagging factor (Fayyad et al., 1996; Kabasakal, 2020; Peng et al., 2008).

Regulatory bodies, financial institutions, and mobile operators have three characteristics in common: first, all three industries have entry barriers, and therefore relatively few players. Second, they run mass businesses, cater to both consumers and corporations, and deal in millions. Third and most importantly, they cannot resell their services, and therefore have to manage their customers directly (Ozmen et al., 2018; Karahoca et al., 2007).

From a prediction perspective, these characteristics require players in these industries to have access to a fair size of big data with real-time notions. These data are able to expose the full extent of estimation, from regression to classification and from natural language processing (NLP) to recommenders. In practice, nearly all aspects of the mobile operating industry can be represented by these data, including revenue forecasts (average revenue per user (ARPU), acquisition/churn projection, customer satisfaction analysis, and product/service recommendations (Huang et al., 2012; KDD, 2018).

The problems with working in churn prediction are twofold. First, unlike pure science, the practical applications of data in the business world limit the probability of finding real data—that is, more data is subject to big data with regulative liabilities in the real business world. This results in data collection becoming more challenging despite the increased practicality of the findings. (Xiao et al., 2016).

With respect to mobile operators, this discrepancy results in a coupling, wherein a group that works with mobile operators with the best real data on real problems is unable to publish this work owing to company asset restrictions, whereas another group that works with special occasion data can publish the information, but the limited applicability diminishes its value. From a scientific developmental perspective, this duality hinders findings production and shares. However, this does not stop stakeholders from employing this dichotomy (Au et al., 2003).

# 2. Dataset

The Association for Computing Machinery (ACM) has been serving the machine learning community since the late-1990s as pioneers of the dataset world, characterized by the initiation of high practicality samples that the ACM calls Knowledge Discovery in Databases (KDD). KDD refers to the broad process of finding knowledge in data and emphasizes the “high-level” application of certain data mining methods. KDD is of particular interest to machine learning, pattern recognition, databases, statistics, artificial intelligence, knowledge acquisition for expert systems, and data visualization.

The KDD Cup 2009 is one of the most renowned KDD datasets out of over 20. It offers the opportunity to work on large marketing databases of the French telecoms company Orange to predict the propensity of customers to switch providers (churn), buy new products or services (appetency), or buy proposed upgrades and add-ons to make more profitable sales (up-selling). The dataset consisted of 100,000 instances split randomly into equally sized training and test sets. Of these, 15,000 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 inaccuracy.

# 3. Proposals

This study scrutinizes the less discussed aspects of the dataset, particularly the churn aspect, through the following four proposals (P):

1. Scrutinizing an implicit bias: Can linear regression be used in a classification problem?

2. AUC (Area Under Curve) optimization: Can decreasing precision or recall help us? Churn’s Type-I receptive nature (unlike that of spam email detection) involves blaring a “false alarm” by trading off with a false negative.

3. Fuzzy logic vs. binary classification: With regard to KDD "emphasizing the ‘high-level’ application” of datasets, promoting the utilization of probabilities instead of traditional binary classification.

4. Feature engineering with NLP: Make customer feedback part of features.

One of the primary limitations is that the original data is no longer being published; therefore, the demonstration for this study was conducted on a smaller version, containing 7,044 records and 21 labels, of which the vast majority are self-explanatory: customerID, gender, SeniorCitizen, PartnerDependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod MonthlyCharges, TotalCharges, and Churn.

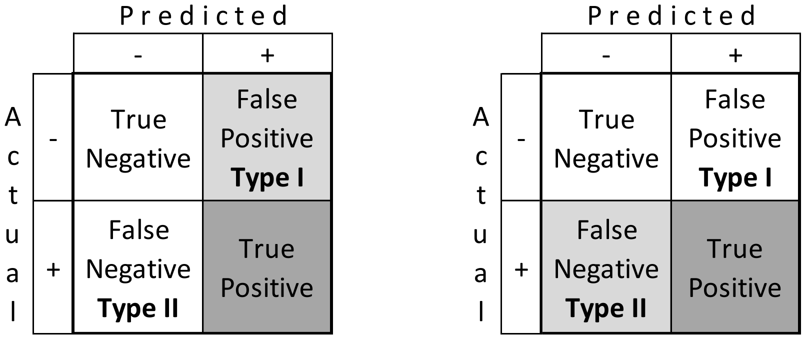
## 3.1. Scrutinizing an Implicit Bias

Linear regression has not been previously used for classification owing to the implicit bias it generates. The binary decisions of all classification models are based on probabilities, and hence, there are no technical limitations that keep linear regression from being applied to probabilities. As long as the regression predictions are between 0 and 1, the results can be utilized to make binary decisions. The primary reason that linear regression was not used was not only because it was considered a non-classifier but also because it was tacitly perceived as the least trivial predictor model (Chen et al, 2017). On the other hand, although it is—that is, binary decision points are also a possibility for regressors—they may not be as useful as perceived. (Rajbahadur et al., 2017) This introduced an incremental discussion point, which was addressed in P3.

## 3.2. AUC Optimization

Lift scores might help with segmentation and cause a boost in confidence levels; however, they may not always be helpful. For mass mailing operations, companies may want to choose 10% with the best accuracy from among their over 10 million customers, which would allow them to decrease their required budget while increasing efficiency. However, unlike in segmentation for mass emails, churn job predictions are unable to do this. Additionally, doing so would jeopardize royalty management, which would need to ignore the vast majority of data.

For optimization, manipulating the AUC might yield some useful combinations. However, the traditional approach, which utilizes medical applications, does not have the luxury of a second decision point; a healthy individual cannot be termed otherwise, and vice-versa. One of the differentiators of classification in the domain of a regression model is that different problems may require polarized decisions. AUC is known for four typologies; while favoring true positive is the ultimate goal, decisions (except medical derivatives that favor F1) about second preferences are also important. As shown in Figure 1, churn may accept false positives as the secondary choice, whereas spam would do the opposite and opt for the false negatives as the secondary choice.



| | | Predicted | |
|---|---|---|---|
| | | - | + |
| Actual | - | True Negative | False Positive **Type I** |
| | + | False Negative **Type II** | True Positive |

| | | Predicted | |
|---|---|---|---|
| | | - | + |
| Actual | - | True Negative | False Positive **Type I** |
| | + | False Negative **Type II** | True Positive |

*Figure 1. Churn scenario (left) vs. Spam scenario (right)*

The churn scenario would accept false alarms while decreasing false negatives, since treating no-churn customers as churn customers would not have an adverse impact on the company. The only risk they took is spending a bit more on royalty than required; however, by doing this, they mitigate the risk of treating a churn customer as a no-churn customer, thus decreasing precision but increasing the information they need (Bell et al., 2019; Sedgwick, 2012)

The spam scenario would accept false negatives while decreasing false alarms, since treating spam customers as no-spam customers would not adversely impact the company. The only risk they took is having a few more emails than usual; however, they mitigate the risk of treating a no-spam email as a spam email, thus decreasing recall but increasing the information they need.

In summary, churn applications prefer false alarms compared to false negatives and would rather pick Type I compared to Type II, thus decreasing precision but increasing the useful knowledge, as favored by KDD.

## 3.3. Fuzzy Logic vs. Binary Classification

Traditional practices with medical applications need to make decisions about whether or not to initiate treatment, since it is usually impossible to apply a partial treatment. However, this is not the case in most business applications, where the graduality is well-received on most occasions, since multiple options are offered. From this perspective, even if the classification is applied, unlike the norm, the binary is not necessarily the most useful, and probabilities might be a better fit (Amaral et al., 2019; Vannucci et al., 2011). In this approach, the necessity of making decisions about secondary options through AUC manipulation would be redundant, since the value of categorization is diminished. In other words, a 51% probability does not have to yield 1 nor does it have to be treated with 99%. When this is a possibility, the churn problem can even be a part of the regressors' world and increase the modeling selections.

## 3.4. Feature Engineering with NLP

Traditionally, customer records are not graded and do not constitute mobile operators' prediction labels. Despite having over a thousand labels, this is also true for The Orange Lab dataset distributed by KDD. Since predicting churn is indicative of customer satisfaction and may already exist within the organization, it is notable that the juxtaposition has not been addressed adequately, and therefore, churn prediction can be claimed to have an implicit bias towards less successful estimations.

Today, customer records arrive as calls and emails, with the former easily able to be transformed to the latter through common audio-to-text libraries. Even with primitive text mining features, a compound Vader score can be generated for each customer that can contribute to the labels, which might lead to greater accuracy. Probabilities may contribute to customer management screens, since providing call center agents access to the churn probability instead of manual flags or binary classification will allow them to offer the designated products to customers.

Besides the four types of revenue derivatives, the vast majority of labels were categorical, followed by six binary typologies, such as the target "Churn" column. Owing to categorical dominance, the Phik correlation was applied through a powerful pandas profiling library. As shown in Figure 2, significant centric interdependencies of revenue labels as well as tenure and MonthlyCharges were noted.

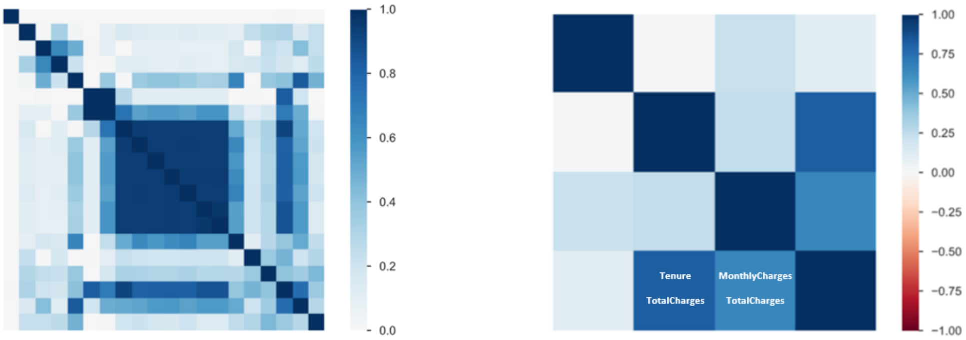


*Figure 2. Density Phik (left) vs. Summary Phik (right)*

Early descriptive findings showed that correlated labels had polarized densities over "Churn." As shown in Figure 3, higher-paying customers were more sensitive to churn, where it implicitly referred to higher-maintenance customers with high expectations. On the other hand, the longer a customer stayed with the company, the less likely were the churns. This coupling raised the dilemma of a high-tenure customer spending less, i.e., owing to the cumulative nature of the latter, tenure and total charges were significantly correlated. However, this may not be the case in the churn specifics.

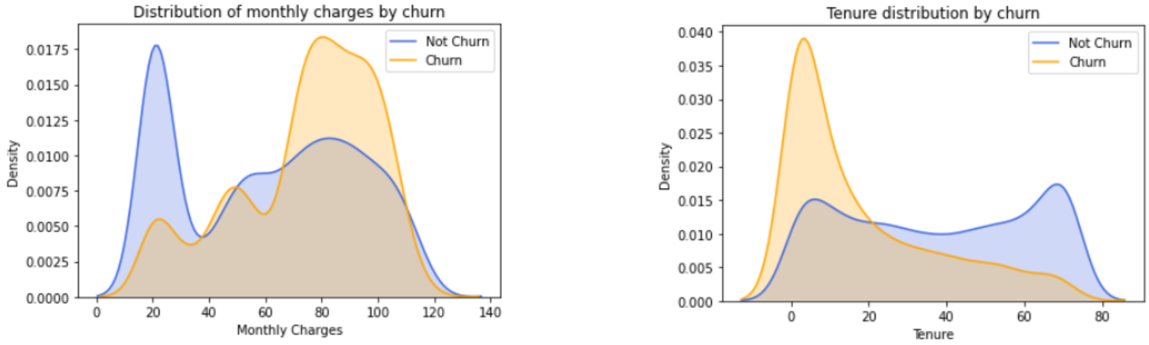


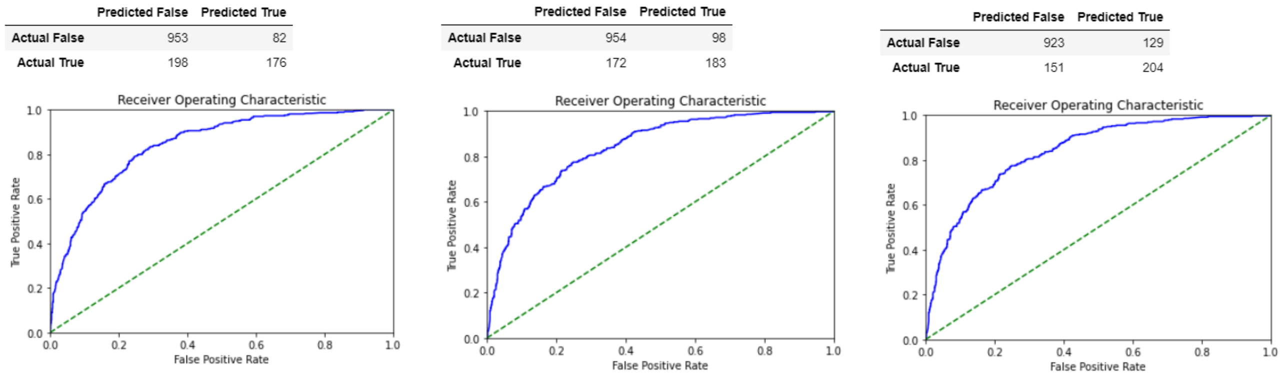
*Figure 3. Churn by Monthly Charges (left) vs. Churn by Tenure (right)*

From the 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 80% and 20% for training and test partitions respectively. Cross-validation efforts were parked for the final optimization stage.

As shown in the first bullet below, LogisticRegressor, RandomForestClassifier, XGBClassifier, and GradientBoostingClassifier were pinned for the fasttrack, where Gradient Boosting led with 0.80 accuracy, similar to the champion's score. The second bullet refers to an underdog, LinearRegression, which produced the best result, with 0.81 accuracy. More importantly, it produced a better recall result, 0.52, which is 10% more than the Gradient Boost. Notably, none of the competition participants practiced LinearRegression due to the classification dichotomy of the problem discussed earlier. The third bullet honors the best model, LinearRegression, with AUC optimization favoring better recall by trading off the precision. As shown in Figure 4, this kept the number of incorrect predictions (280) from being exploited. To be precise, 198/98 was considered 172/98 with 0.45 tolerance, i.e., accuracy was not jeopardized by smaller tolerances, although smaller tolerances will yield better recall figures. Efforts yielded 10% more compared to its predecessor, with a 0.58 recall score.

Notably, once regressors produce between 0 and 1, both RMSE and Accuracy are mentionable simultaneously. To summarize the findings:

- GB Classifier, 0.50 Tolerance | Accuracy = 0.80 | Precision = 0.68 | Recall = 0.47
- Lin Regressor, 0.50 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



| | Predicted False | Predicted True |
|---|---|---|
| Actual False | 953 | 82 |
| Actual True | 198 | 176 |

| | Predicted False | Predicted True |
|---|---|---|
| Actual False | 954 | 98 |
| Actual True | 172 | 183 |

| | Predicted False | Predicted True |
|---|---|---|
| Actual False | 923 | 129 |
| Actual True | 151 | 204 |

*Figure 4. 1st Winning Model (left) vs. 2nd Winning Model (center) vs. AUC (right)*

From a practicality perspective, around 50 people switched from missed churn numbers to a false alarm, where the latter is preferable. Additionally, around 30 people transferred to churn numbers so they may be over-treated, i.e., the model will have around 25% fewer missed churn numbers and around 60% more false alarms (bearable rounding error) with around 20% penalized (safer from a royalty/churn management standpoint), resulting in greater churn numbers.

# 4. Feature engineering with text mining

Text mining has a rich history, with roots can be traced to the library efforts of the first established universities. Over the course of several years, its capabilities expanded from summarization to information extraction/discovery, clustering, context/topic meaning, and deep discovery, such as identifying sentiments, idioms, and innuendo. Such software can also identify entities and emotions in a sentence and use these to determine if the entity is being viewed positively or negatively (Lu et al., 2012).

Often, a sentiment cannot be understood by merely studying words, which complicates the process of text mining. Sentiment analysis is generally a starting point in analyzing a text and is then coupled with other techniques such as topic analysis. Sentiment analysis is usually done using a corpus of positive and negative words; some sources compile lists of positive and negative words, whereas others include the polarity—the degree of positivity or negativity—of each word (Niculescu-Mizil et al., 2009).

Miner’s algorithm distinguishes sentiment analysis into two parts: understanding the sentence and understanding the word. Understanding that they are not contradictory and instead, support each other, results in avenues where both paths are explored. From a document perspective, the study falls into the clustering area, whereas from words perspective, it is more associated with NLP (Siddiqui et al., 2019, Szarvas, 2008).

## 4.1. Vader Sentiment

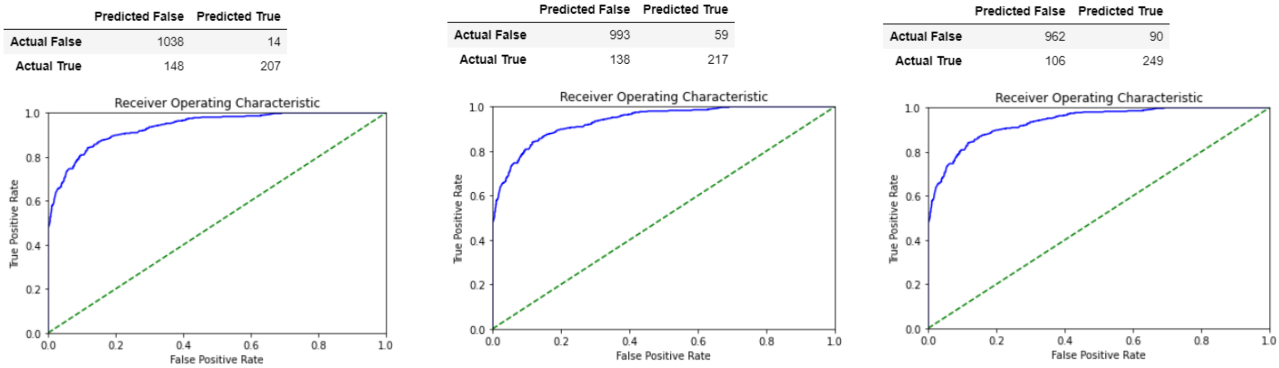
A SentimentAnalyzer is a tool that is used to implement and facilitate sentiment analysis tasks using NLTK (Natural Language Toolkit) features and classifiers, especially for teaching and demonstrative purposes, i.e., it is a weighted word analysis that uses Vader. Vader contains a list of 7,500 features weighted by how positive or negative they are. The software then uses these features to calculate how positive, negative, or neutral a passage is. It then combines these results to give a compound sentiment for the passage (higher = more positive). Ten individuals trained on Twitter data and generally considered good at informal communication 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 receive 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
- Within this study, three social media listening were made for the sake of customer satisfaction. The average was added as a new feature under column name “Compound”
- Twitter.com: 2,500 latest tweets with a compound score varying between -0.9 and 0.6
- Trustpilot.com: 1,500 records with a compound score varying between -0.7 and 0.5
- ConsumerAffairs.com: 1,000 records with a compound score varying between -0.8 and 0.5

## 4.2. Repeating Process

To make it comparable, RandomForestClassifier, XGBClassifier, and GradientBoostingClassifier were pinned for the first run. This time, the XGBClassifier led with 0.89 accuracy and 0.58 recall scores. The second bullet refers to LinearRegression, which produced the best result, with 0.86 accuracy, and 0.61 recall. The third bullet honors the best model, LinearRegression, with AUC optimization favoring better recall by trading off the precision. As shown in Figure 5, this ensured that the number of wrong predictions was not exploited by smaller tolerances for further recall improvements. Efforts yielded almost 15% more compared to its predecessor. In summary:

- XGBClassifier, 0.50 Tolerance | Accuracy = 0.89 | Precision = 0.94 | Recall = 0.58
- Lin Regressor, 0.50 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



| | Predicted False | Predicted True |
|---|---|---|
| Actual False | 1038 | 14 |
| Actual True | 148 | 207 |

| | Predicted False | Predicted True |
|---|---|---|
| Actual False | 993 | 59 |
| Actual True | 138 | 217 |

| | Predicted False | Predicted True |
|---|---|---|
| Actual False | 962 | 90 |
| Actual True | 106 | 249 |

*Figure 5. 1st Winning Model (left) vs. 2nd Winning Model (center) vs. AUC (right) w/ Compound score (Vader sentiment)*

From a practicality perspective, around 75 people switched from missed churn numbers to the false alarm, where the latter is preferable. Additionally, around 40 people transferred to churn numbers so they may be over-treated, i.e., the model used will have approximately 30% fewer missed churn numbers and around five times as many false alarms, with approximately 20% penalized (safer from a royalty/churn management standpoint), thus yielding more churn numbers.

# 5. Optimization

The traditional winning path approach ignores both linear regression and recall performance, resulting in the worst results per churn estimation as well as missing churn numbers. Adding the incremental predictive power of customers' NLP yields two significant figures, as follows:

- Missed churn numbers decreased around 50%, from 196 to 106 customers
- Re-estimated churn numbers increased around 50%, from 176 to 249
- From a technicality standpoint, 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)

The important factors were drafted in accordance with the latest best performing model, as shown in Figure 6, where the "Compound" score dominated, followed by Contract_Month_to_month with 0.17 scaled importance.

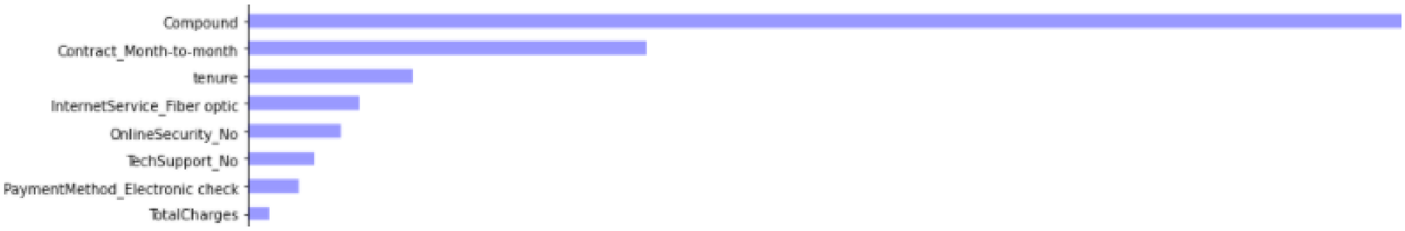


*Figure 6. Important Factors*

## 5.1 AML (Auto Machine Learning) Optimizations

The automated machine learning (AML) concept is noted for its optimized Omni-model environment as well as validation (of what was gathered) standpoint (Olson et al., 2016; Yao et al., 2018; Chen et al., 2021). AML works with several 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.ai, 2017; Drozdal et al., 2020). It further adjusts default parameters in a way to find the best split through cross-validation, finds the best algorithms, and optimizes the entire workflow except the decision points about how to trade-off. There have been a number of open-source attempts at AML in both the open source and commercial arena, including the following in the former:

- auto-Weka is a Java library built on Weka
- auto-sklearn is a Python library that optimizes per Bayesian
- TPOT works with Python
- auto-keras is a Python library with powerful classification/regression models for structured data as well as images and texts
- H2O AutoML is developed using Java and works with Python, R, and Scala

Of these, H2O AutoML possesses three distinctive features, namely, it is explicit in terms of model names and flexible in terms of inclusions and exclusions, it provides a confusion matrix if applicable, and it proposes important factors (LeDell, 2018; Lee et al., 2019; Gursakal et al., 2021). More specifically, H2O AutoML requires only two data and two stopping parameters, whereas it handles a total of 27 parameters to burst user control (Miner et al., 2012; Wang et al., 2019; Blohm et al., 2020).

From a comparability perspective, the same training and test split were applied with 80% and 20% respectively. The combined, classifiers as well as regressors, results is tabulated as below.

*Table 1. Combined results with AML*

| model_id | auc | logloss | aucpr | mean_per_class_error | rmse | mse |
|---:|---:|---:|---:|---:|---:|---:|
| *GBM_grid__1_AutoML_20210124_192217_model_5* | 0.937224 | 0.264585 | 0.879735 | 0.15721 | 0.28703 | 0.082386 |
| *GLM_5_AutoML_20210124_192217* | | | | | 0.287724 | 0.0827849 |
| *StackedEnsemble_AllModels_AutoML_20210124_192217* | | | | | 0.290865 | 0.0846027 |
| *StackedEnsemble_BestOfFamily_AutoML_20210124_192217* | 0.93569 | 0.279556 | 0.879069 | 0.149809 | 0.291073 | 0.0847233 |
| *GBM_1_AutoML_20210124_192217* | 0.934744 | 0.27194 | 0.877003 | 0.157487 | 0.288882 | 0.083453 |
| *GBM_2_AutoML_20210124_192217* | 0.933587 | 0.273209 | 0.875652 | 0.15773 | 0.289969 | 0.0840819 |
| *GBM_3_AutoML_20210124_192217* | 0.932083 | 0.276879 | 0.872136 | 0.157011 | 0.292782 | 0.0857214 |
| *GBM_grid__1_AutoML_20210124_192217_model_3* | 0.930889 | 0.283341 | 0.869795 | 0.172086 | 0.296791 | 0.0880849 |
| *GBM_4_AutoML_20210124_192217* | 0.929928 | 0.28163 | 0.871707 | 0.158404 | 0.29284 | 0.0857552 |
| *GBM_grid__1_AutoML_20210124_192217_model_2* | 0.929639 | 0.287839 | 0.868668 | 0.158851 | 0.297383 | 0.0884364 |

Although the last winning model is classifier version of GBM, it is notable that it is challenged by regressors, not only by  pur regressors as GLM, but also the regressor version of the StackedEnsemble. With cross-validation added to the process, K-fold 5 generated accuracies varying between 0.86 and 0.90. To mitigate the overfitting odds, its mode (0.89) was accepted, where the confusion matrix-based results follow as in Table 2.

*Table 2. Confusion Matrix with AML Optimization*

| | Predicted False | Predicted True |
|---|---|---|
| *Actual False* | 966 | 82 |
| *Actual True* | 85 | 274 |
| *Total* | 1051 | 356 |

Overall comparison to the previous model can be summarized as:

- GB Classifier, 0.43 Tolerance (RMSE = 0.29) | Accuracy = 0.89 | Precision = 0.77 | Recall = 0.78
- Missed churn numbers decreased around 20%, from 106 to 86 customers
- Re-estimated churn numbers increased around 10%, from 249 to 274
- From a technicality standpoint, the whole tripartite was augmented up to 50%
    - Accuracy = 0.86 | Precision = 0.73 | Recall = 0.70
    - Accuracy = 0.89 | Precision = 0.77 | Recall = 0.78

### 5.2. Lift Optimizations

As examined earlier, gains/lift table do not produce useful knowledge for churn prediction. However, mega mobile operators with over 100 million subscribers will statistically have over 1 million data subjects to churn and therefore, may want to work with deciles (10 quantiles) data, as shown in Figure 7. Scores generated for testing stated the average response rate as 25.36% and average score as 26.48%. Fifteen partitions were led by 99% AUC, which is 7% more than the 93% AUC average.

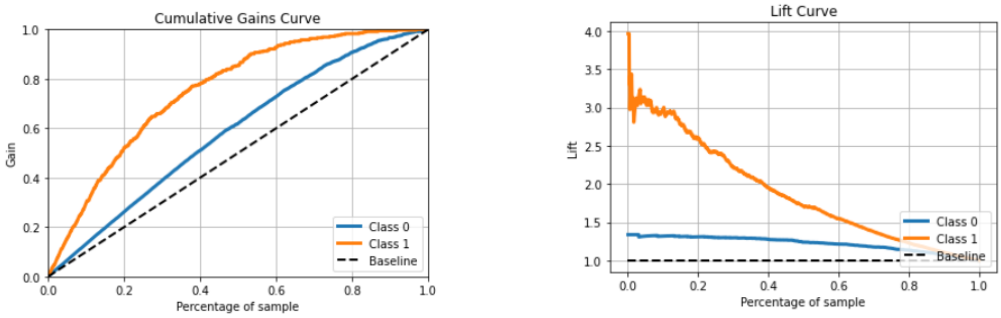


*Figure 7. Curves for Gain and Lift*

## 6. Discussions

This study introduces frontier ideas by placing inceptions within typical stages of churn prediction. As part of the study, four proposals were generated and applied, and the winning model was challenged with double-digit improvement in each aspect of the classification performance tripartite, namely accuracy, precision, and recall. The proposals and outcomes of the study were as follows:

- As per the “high-level” application of datasets, as per KDD, utilization of probabilities instead of traditional binary classification may help call center agents ensure customer satisfaction through real-time decision-making over the phone.

- With probability-like figures, the implicit bias was scrutinized and classification problems in churn prediction were challenged using regressors. Consequently, it was found that regressors could be an option as well as a leader model, with 11% more recall performance and 1 point more inaccuracy.

- AUC optimization was revisited. As per churn’s Type-I receptive (unlike spam email detection) nature, bursting the “false alarm” by trading off with the false negative yielded 45% tolerance, which resulted from 12% more recall performance without compromising accuracy.

- Although feature engineering with NLP is not unknown, it is rare from a practitioner’s perspective. Thus, Vader sentiment dynamics were used to attempt making customer feedback a part of features. This is different and more useful than knowing who will churn in binary, since 0.51 cannot yield 0.99. Having a large spectrum of products and services can address a variety of compartments for probabilities between 0 and 1, allowing five to seven categories to be created, allowing agents to help customers. This is also vital from a royalty management perspective. Summary of scores can be found below:

*Table 3. Cumulative comparison*

| Occasion | Winning Model | Accuracy | Precision | Recall | F1 | RMSE |
|---|---|---|---|---|---|---|
| *Base [Classifiers]* | Gradient Boosting | 0.8 | 0.68 | 0.47 | 0.56 | |
| *Challenge [Regressors]* | Linear Regression | 0.81 | 0.65 | 0.52 | 0.58 | 0.44 |
| | Linear Regression* | 0.8 | 0.61 | 0.58 | 0.59 | 0.45 |
| *Compound-Vader [All]* | Linear Regression | 0.86 | 0.79 | 0.61 | 0.69 | 0.38 |
| | Linear Regression* | 0.86 | 0.73 | 0.7 | 0.71 | 0.37 |
| *AML Optimization [All]* | Gradient Boosting* | 0.89 | 0.77 | 0.78 | 0.77 | 0.29 |
| *Performance* | | 11% | 13% | **66%** | 39% | 52% |

*Recall-favoring version of winning models

Churn prediction does not indicate survival. All the degrees in this business question are important. In other words, churn prediction requires more than binary decisions (Data Science, 2020; Fan et al., 2006; Lee at al., 2010) The findings of this churn prediction journey can be summarized in the following four hypotheses:

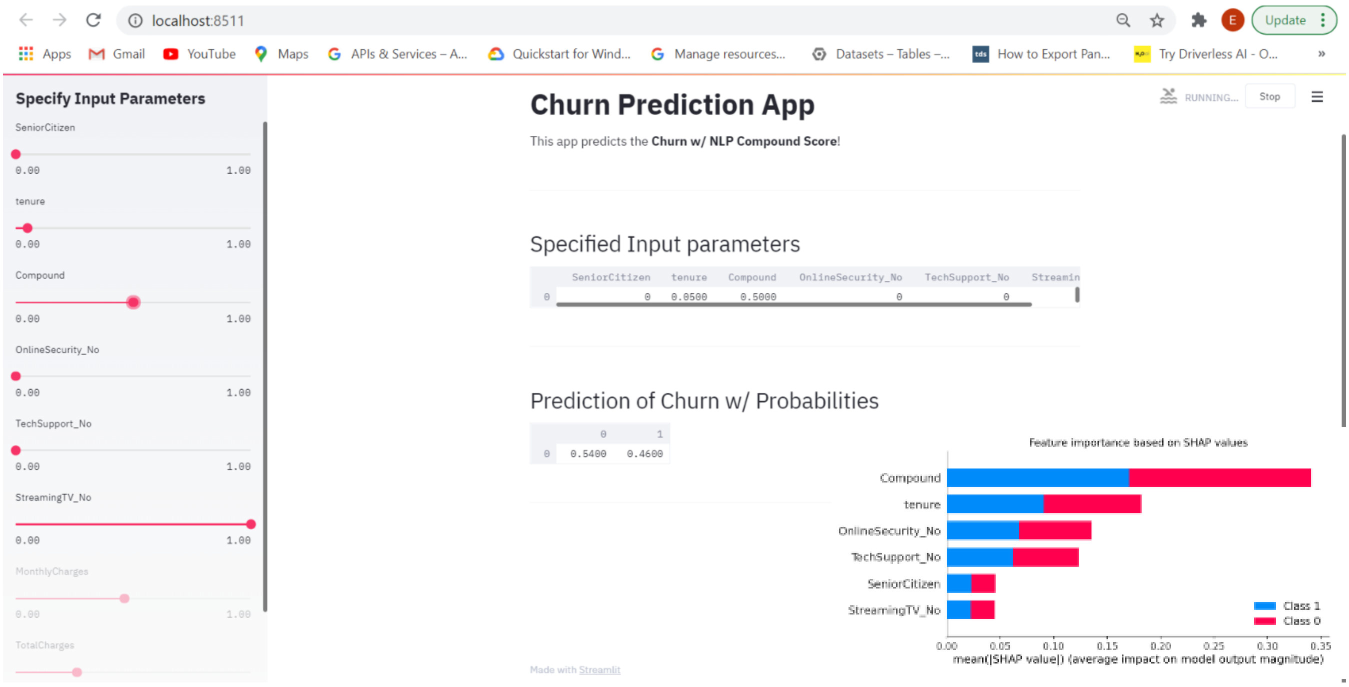
1. Respecting graduality (probabilities rather than 0 and 1) in classifiers improves the churn model

2. Being more receptive in model bias, including regressors, validates the churn model

3. Favoring recall performance improves the churn model

4. Working with customer feedback to predict customer churn improves the churn model

Figure 8 demonstrates the important factors from a practicality perspective in an app as presented through the Streamlit library. Future studies can examine a larger number of applications and intrusions in linear regressions for churn prediction.



# Churn Prediction App

This app predicts the **Churn w/ NLP Compound Score**!

## Specified Input parameters

|   | SeniorCitizen | tenure | Compound | OnlineSecurity_No | TechSupport_No | Streamin |
|---|---|---|---|---|---|---|
| 0 | 0 | 0.0500 | 0.5000 | 0 | 0 |  |

## Prediction of Churn w/ Probabilities

|   | 0 | 1 |
|---|---|---|
| 0 | 0.5400 | 0.4600 |

*Figure 8. Probability-based Churn Predictor in Production with Streamlit*

## Future Directions

Unlike the dichotomy in literature, regressors can make a winning model (or significantly challenge it) with churn predictions. However, traditionally, especially the linear regressors are being criticized by favoring multicollinearity. That could be one of the reasons that competitors do not explicitly mention about any inclusion of it, since, unlike the version we practiced here with 21, the original data has 230 labels.

In our case, after eliminating labels with non-significant t-values, only 6 factors were identified and combinations favoring an acceptable condition number (<100) was not a challenge.

On the other hand, the nature of churn predictions might prevent us to revisit the labels anyways, since, as shown on Figure 6, the vast majority factors are being blatantly dominated by the "Compound" label, a.k.a. text sentiment analysis for customer complaints. In other words, it seems that customers are not tacit with what they think, listening to them and grading what they say might be enough to make decisions.

To sum, more research for possible effects of multicollinearity on churn predictions with big data is highly encouraged.

## References

Amaral, R. P. F., Ribeiro, M. V., & de Aguiar, E. P. (2019). Type-1 and singleton fuzzy logic system trained by a fast scaled conjugate gradient methods for dealing with binary classification problems. Neurocomputing, 355, 57-70.

Au, W. H., Chan, K. C., & Yao, X. (2003). A novel evolutionary data mining algorithm with applications to churn prediction. IEEE transactions on evolutionary computation, 7(6), 532–545.

Bell, A., Ward, P., Tamal, M. E. H., & Killilea, M. (2019). Assessing recall bias and measurement error in high-frequency social data collection for human-environment research. Population and Environment, 40(3), 325-345.

Blohm, M., Hanussek, M., & Kintz, M. (2020). Leveraging Automated Machine Learning for Text Classification: Evaluation of AutoML Tools and Comparison with Human Performance. arXiv preprint arXiv:2012.03575.

Chen, Y. W., Song, Q., & Hu, X. (2021). Techniques for automated machine learning. ACM SIGKDD Explorations Newsletter, 22(2), 35–50.

Chen, Y., Shao, Y., Yan, J., Yuan, T. F., Qu, Y., Lee, E., & Wang, S. (2017). A feature-free 30-disease pathological brain detection system by linear regression classifier. CNS & Neurological Disorders-Drug Targets (Formerly Current Drug Targets-CNS & Neurological Disorders), 16(1), 5-10.

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

Drozdal, J., Weisz, J., Wang, D., Dass, G., Yao, B., Zhao, C., ... & Su, H. (2020, March). Trust in automl: Exploring information needs for establishing trust in automated machine learning systems. In Proceedings of the 25th International Conference on Intelligent User Interfaces. 297–307.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining. Communications of the ACM, 49(9), 76–82.

Fayyad, Piatetsky-Shapiro, & Smyth. (1996). From Data Mining to Knowledge Discovery: An Overview, in Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, Advances in Knowledge Discovery and Data Mining. AAAI Press / MIT Press. 1–34.

Gürsakal, N. , Gürsakal, S. & Çelik, S. (2021). Big Data Companies and Open Source Movement . Avrupa Bilim ve Teknoloji Dergisi , (21) , 680-689 . Retrieved from https://dergipark.org.tr/en/pub/ejosat/issue/59648/822219

H2O.ai, H2O AutoML. (2017). http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html. H2O version 3.30.0.1.

Huang, B., Kechadi, M. T., & Buckley, B. (2012). Customer churn prediction in telecommunications. Expert Systems with Applications, 39(1), 1414–1425.

Kabasakal, İ . (2020). Customer Segmentation Based On Recency Frequency Monetary Model: A Case Study in E-Retailing . Bilişim Teknolojileri Dergisi , 13 (1) , 47-56 . DOI: 10.17671/gazibtd.570866

Karahoca, A., Karahoca, D., & Aydin, N. (2007). GSM Churn Management Using an Adaptive Neuro-Fuzzy Inference System. The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007), 323-326.

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

Lango,M.(2019).Tackling the Problem of Class Imbalance in Multi-class Sentiment Classification: An Experimental Study. Foundations of Computing and Decision Sciences,44(2) 151-178. https://doi.org/10.2478/fcds-2019-0009

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

Lee, D. J. L., Macke, S., Xin, D., Lee, A., Huang, S., & Parameswaran, A. G. (2019). A Human-in-the-loop Perspective on AutoML: Milestones and the Road Ahead. IEEE Data Eng. Bull., 42(2), 59–70.

Lee, S., Song, J., & Kim, Y. (2010). An empirical comparison of four text mining methods. Journal of Computer Information Systems, 51(1), 1–10.

Lu, N., Lin, H., Lu, J., & Zhang, G. (2012). A customer churn prediction model in telecom industry using boosting. IEEE Transactions on Industrial Informatics, 10(2), 1659–1665.

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. 29–41.

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. (2009). Winning the KDD cup orange challenge with ensemble selection. The 2009 Knowledge Discovery in Data Competition. 23–34.

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. 485–492.

Özmen, M , Delice, Y , Kızılkaya Aydoğan, E . (2018). Telekomünikasyon Sektöründe PSO ile Müşteri Bölümlemesi . Bilişim Teknolojileri Dergisi , 11 (2) , 163-173 . DOI: 10.17671/gazibtd.368460

Peng, Y., Kou, G., Shi, Y., & Chen, Z. (2008). A descriptive framework for the field of data mining and knowledge discovery. International Journal of Information Technology & Decision Making, 7(04), 639–682.

Rajbahadur, G. K., Wang, S., Kamei, Y., & Hassan, A. E. (2017, May). The impact of using regression models to build defect classifiers. In 2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR) (pp. 135-145). IEEE.

Sedgwick, P. (2012). What is recall bias?. BMJ, 344.

Siddiqui, S., Rehman, M. A., Doudpota, S. M., & Waqas, A. (2019). Ontology driven feature engineering for opinion mining. IEEE Access, 7, 67392-67401.

Szarvas, G. (2008). Feature engineering for domain independent named entity recognition and biomedical text mining applications. University of Szeged, Szeged.

Vannucci, M., & Colla, V. (2011). Novel classification method for sensitive problems and uneven datasets based on neural networks and fuzzy logic. Applied Soft Computing, 11(2), 2383-2390.

Wang, C., & Wu, Q. (2019). Flo: Fast and lightweight hyperparameter optimization for automl. arXiv preprint arXiv:1911.04706.

Xiao, J., Jiang, X., He, C., & Teng, G. (2016). Churn prediction in customer relationship management via GMDH-based multiple classifiers ensemble. IEEE Intelligent Systems, 31(2), 37–44.

Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y. F., Tu, W. W., ... & Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. arXiv preprint arXiv:1810.13306.