UNICAMP

FACULDADE DE ENGENHARIA ELÉTRICA E COMPUTAÇÃO

EA006 - Trabalho de Fim de Curso

Previsão de Churn em Plataformas com Efeitos de Rede baseado em Árvores de Decisão

Aluna:

Fernanda Coelho de Queiroz

Orientador:

Prof. Dr. Fernando Antonio Campos Gomide

Campinas, SP, Dezembro de 2018

Agradecimentos

Agradeço ao Prof Dr. Fernando Gomide que gentilmente aceitou me orientar. Sem a sua disposição e excelentes recomendações este projeto não teria sido possível.

Resumo

Este trabalho visou aprimorar conhecimentos de Machine Learning através do desenvolvimento de um modelo preditivo de churn para empresas com modelos de negócio multi-laterais e não contratuais. A construção das features foi feito parcialmente utilizando a biblioteca tsfresh e o algoritmo de classificação escolhido foi a Árvore de Decisão. Todo o código desenvolvido foi disponibilizado publicamente no Github. Diante de uma base de dados onde aproximadamente 11% das amostras pertence a classe positiva (churn) foram obtidas as seguintes performances: AUC-ROC: 0.92, F1: 0.44, Precision: 0.63

Key-words: churn prediction, decision tree, non-contractual, two-sided market, network effects, time-series

Abstract

This work aimed to deepen knowledge of Machine Learning through the development of a predictive model of churn for companies with multi-lateral and non-contractual business models. The construction of the features was partially done using the tsfresh library and the chosen classification algorithm was the Decision Tree. All the developed code was made publicly available on Github. In a given a database where approximately 11 % of the samples belongs to the positive class (churn) the following performances were obtained: AUC-ROC: 0.92, F1: 0.44, Precision: 0.63

Key-words: churn prediction, decision tree, non-contractual, two-sided market, network effects, time-series

Sumário

Lista de Figuras

Lista de Tabelas

1. Introdução

1.1 Churn, sua relevância e os efeitos de rede

Churn de clientes é o nome do evento em que o cliente termina seu relacionamento com uma empresa. Esse tipo tipo de evento é muito prejudicial para as operações, independente do tipo de negócio, pois perder clientes também significa perder o motor que gera o faturamento.

No caso de mercados fortemente saturados, como por exemplo o de telecomunicações, adquirir novos clientes é cada vez mais difícil e custoso para essas empresas.[1] Por este motivo, existe um interesse econômico muito forte em desenvolver estratégias de retenção de clientes através de redução do churn.[2] Essa pode ser apontada como principal causa pra tantas pesquisas desenvolvidas[3] sobre churn na referida indústria.

Existem também indústrias onde a redução do churn pode representar uma vantagem economica relevante, independente do nível de maturidade do mercado.[4][5][6] Este é o caso de empresas que possuem modelos de negócio multi-laterais, que distinguem-se por atuar fornecendo serviços para ao menos dois grupos distintos de usuários.[7] O valor que este tipo de negócio oferece para determinado grupo de usuários geralmente está atrelado ao tamanho e qualidade dos demais grupos.[8][9] Isto é, há forte presença de efeitos de rede indiretos além dos efeitos diretos normalmente existente em modelos tradicionais.

O efeito de rede pode ser definido como a externalidade, ou efeito, causado pela alteração do número de usuários em determinada comunidade. Estes efeitos

podem ser positivos ou negativos e também diretos ou indiretos.[10] Classifica-se como direto o efeito de rede que gera externalidades na mesma comunidade ao qual pertence o usuário, sendo classificado como indireto quando a externalidade ocorre em uma comunidade distinta. Diversas empresas hoje apoiam-se em efeitos de rede para oferecer um serviço de maior qualidade para seus usuários.[11]

Table 1.1: Exemplos de Efeito de Rede

	Direto	Indireto
Positivo	WhatsApp: Quanto mais usuários usam o WhatsApp, mais força o WhatsApp tem para atrair novos usuários	Uber: Quanto mais motoristas utilizam a plataforma, menor o custo e o tempo de espera para um passageiro que quer ser transportado
Negativo	Celular: Quando mais usuários usando a internet numa mesma área, menos velocidade cada pessoa terá para navegar devido à congestão da rede	

Efeitos de rede podem ser consideradas um dos fatores que impactam as Forças de Porter[1] criando uma forte barreira de entrada, aumentando os custos de substituição e a proposta de valor[13]. Isso é especialmente verdadeiro hoje quando outros fatores que poderiam dificultar o surgimento de competidores estão ameaçados.[14]

1.2 Objetivo

A proposta desse trabalho é realizar um estudo de algoritmos de aprendizado de máquina através da aplicação de árvores de decisão para endereçar um problema real de classificação. O problema escolhido trata-se da evasão de clientes em organizações com modelos de negócios não contratuais e cujo crescimento se apoie majoritariamente em efeitos de rede indiretos.

[1] "The Five Forces is a framework for understanding the competitive forces at work in an industry and which drive the way economic value is divided among industry actors"[12]

1.3 Motivação

A principal motivação deste trabalho foi aprofundar e validar conhecimentos na área de Machine Learning aplicando técnicas e processos para endereçar um problema real de negócios. Além do aprendizado naturalmente obtido durante este projeto, pretende-se fornecer à empresa concedente dos dados um modelo capaz de prever quais clientes são mais propensos à abandonarem o serviço, tornando possível que esta tome medidas para reduzir a ocorrência de churn.

2. Metodologia

2.1 Ferramentas e materiais utilizados

- Docker
- Git
- Python
    - Pandas
    - Sklearn
    - TSfresh
- PyCharm
- Makefile

2.2 Preparação dos dados

A preparação dos dados é um conceito amplo que inclui diversas tarefas.[15] Para este problema em específico, nos concentraremos nas seguintes etapas:

- Coleta e limpeza dos dados

- Engenharia de features, transformando os dados em um formato mais adequado para a modelagem.

Coleta e limpeza dos dados

A base de dados foi obtida a partir de uma parceria com uma empresa com as características mencionadas (*i.e.* modelo de negócio não contratual com forte presença de efeitos de rede). Devido ao acordo confidencialidade, os dados não poderão ser compartilhados com terceiros, assim como algumas das decisões, tomadas com base em informações estratégicas, para tratamento desses e modelagem do problema.

A limpeza dos dados é considerada uma das partes mais demoradas da ciência de dados. A preparação dos dados pode tomar 60% do tempo do projeto, porém representa somente 15% da importância para o sucesso do mesmo. (PYLE[16], 1999) Para manter o sigilo de informações sensíveis, os métodos específicos aplicados para normalização da base de dados ficarão de fora do escopo de apresentação deste trabalho. Contudo, é esperado que essa omissão não impacte na relevância do mesmo.

Todo o código fonte necessário para execução deste projeto encontra-se disponível publicamente no github[1], juntamente com instruções para facilitar a reprodução desta analise.

Engenharia de features

O algoritmo a ser aplicado, indução de árvore de decisão, pertence à uma classe de métodos de aprendizado de máquina dito 'supervisionado'. Esse tipo de algoritmo é aplicado quando queremos encontrar uma função $f(X)$ que mapeie as variáveis de entrada, X, nas variáveis de saída, Y. Isto é, $Y = f(X)$

Para o problema que queremos resolver, prever se dado cliente irá abandonar o serviço, a forma mais direta de estruturar os dados é com um tabela onde cada linha representa um cliente e cada coluna representa uma das múltiplas informações que podemos usar para caracterizar os clientes.

[1] `https://github.com/fcqueiroz/tcc_churn`

Os dados fornecidos, porém, encontram-se no formato de séries temporais, onde cada linha representa um pedido realizado por algum dos clientes. Torna-se então necessário transformar os dados do formato original para um formato mais adequado para a modelagem, extraindo no processo características descritivas (features) dos dados originais.

Criação do target

De modo simples e geral, definimos um cliente ativo como um cliente que gera receita para uma empresa e muito provavelmente também irá gerar receita no futuro. De modo análogo, um cliente inativo é um cliente que não gera receita para a empresa e muito provavelmente não irá gerar receita no futuro. O evento de churn, ou inativação, é, portanto o evento em que o estado de um cliente ativo se altera para inativo. De modo mais prático, um cliente que foi inativado é um cliente que até um momento recente gerava receita para a empresa, mas muito provalmente não irá gerar receita no futuro.

Devido à característica não-contratual do negócio, o estado de um cliente em certo instante não pode ser determinado diretamente. É possível determinar o quanto de receita este cliente gerou no passado recente, mas a receita futura é uma variável aleatória. Ao contrário de situações de negócios contratuais, onde o evento de churn é identificavel pelo instante em que um cliente decide encerrar o contrato, o evento de churn não é diretamente observável. Sendo assim, em situações não-contratuais o churn deve ser inferido por métodos indiretos.

O problema de identificação do estado de um cliente em certo momento pode ser modelado de forma satisfatória por uma análise de probabilidade baseado na frequência de compra e recência do cliente, como no modelo Beta-Geométrico/NBD (FADER et al.[17], 2005). Porém, para fins de simplicidade, resolveu-se adotar o mesmo método que é aplicado na empresa fornecedora dos dados. Isto é, um cliente é considerado ativo se ele tiver feito ao menos um pedido nos últimos 30 dias.

O primeiro problema dessa solução é o atraso na identificação de quando um cliente torna-se inativo. Isto porque, assumindo que o cliente tomou a decisão de não utilizar mais os serviços da empresa no dia $D+0$, somente no dia $D+30$ o

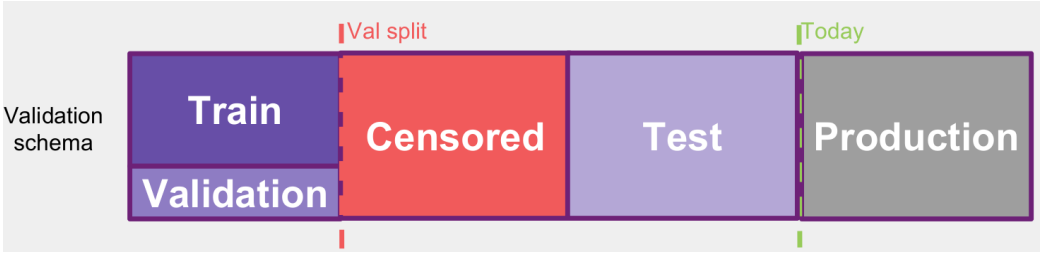


Figure 2.1: Esquema de Validação

indicador irá alertar os administradores da companhia a respeito dessa decisão. Esse tipo de atraso de identificar um churn reduz a probabilidade de um método de retenção de clientes ser efetivo.

Um sistema capaz de antever o estado desse indicador somente utilizando informações obtidas até o instante $D+0$, ou seja, um sistema capaz de prever o comportamento do cliente pelos próximos 30 dias, poderia representar uma melhora significativa no processo de retenção de clientes desta empresa.

Para estimar corretamente a capacidade preditiva do modelo é necessário limitar os dados disponíveis temporalmente, de modo a simular o ambiente de mundo real em que não temos como tomar decisões hoje com coisas que acontecerão no futuro.[18]

Por esta razão, parte dos dados para treinamento do modelo deve ser censurado para o cálculo das variáveis de entrada de modo a simular dados futuros que não podem ser conhecidos no momento de otimização do modelo e tomada de decisão.

Criação das variáveis de entrada

Como dito anteriormente, a criação das variáveis de entrada possui um desafio inerente à forma como os dados estão estruturados. Para facilitar o processo de classificação, necessitamos modificar o formato dos dados para que cada registro seja referente a um cliente específico e suas características.

Para fazer isso foram utilizados dois métodos para extração de features. O primeiro, manual, consiste na criação de features relevantes a partir do próprio conhecimento de negócios e a partir do que está disponível na literatura sobre previsão de churn.[19] A segunda forma, automatizada, consiste na extração de

Table 2.1: Features extraídas manualmente

Feature	Descrição
recency_v0	A idade do cliente no momento de seu último pedido realizado. É calculada como a diferença, em dias, entre a data do primeiro e do último pedido realizado
recency_v1	Tempo decorrido entre o último pedido de um cliente e a data presente, em dias
c_age	Idade presente do cliente, em dias
gravity	Centro de massa dos pedidos realizados. É calculada como o somatório da idade que o cliente tinha ao realizar cada um dos pedidos **recency_v0** dividido pela quantidade de pedidos realizados
gravity_norm	Centro de massa dos pedidos realizados normalizada pela idade presente do cliente. É calculada como o valor da **gravity** dividido pela idade presente do cliente, **c_age**
monetary	Valor total gasto em todos os pedidos do cliente até a data presente
frequency	Número de dias em que o cliente realizou um pedido, desconsiderando o dia em que o cliente entrou no serviço
frequency_norm	Número de dias em que o cliente realizou um pedido, desconsiderando o dia em que este entrou no serviço, normalizado pela idade presente do cliente. Calculada como **frequency** dividido por **c_age**

features a partir de bibliotecas apropriadas para a caracterização de séries temporais. Neste caso, foi utilizado o pacote python *tsfresh*[2], que propoe reduzir o tempo gasto em feature engineering através da extração automatizada de características relevantes de séries temporais. É importante mencionar que após a extração de todas as features, algumas delas foram descartas por apresentarem alta correlação entre si.

[2]Abreviação para "Time Series Feature extraction based on scalable hypothesis tests"

2.3 Modelagem

Escolha do modelo

O modelo foi estruturado em torno do algoritmo de indução de Árvore de Decisão já implementado pelo pacote python *scikit-learn*. A escolha do algoritmo deu-se por vários fatores, dentre eles:

- Ser pouco afetado pela presença de features irrelevantes no dataset. Como o próprio método de indução de árvores é realizado utilizando o ganho de informação, a seleção das features mais importantes pcorre de forma integrada ao processo.
- Por ser um algoritmo fácil de interpretar e de explicar mesmo para pessoas sem conhecimentos profundos em matemática.
- Robustez contra distruições não equilibradas e enviesadas pois, ao contrário de algoritmos como, as Árvores de Decisão não necessitam de suposições sobre a linearidade dos dados.

Métrica de otimização

Na fase de escolha da métrica de otimização foi considerado que a melhor métrica deveria estar alinhada com os interesses financeiros do negócio. Ou seja, o modelo deveria ser otimizado de forma a maximizar o retorno financeiro de sua implementação. Porém, devido à variação temporal da taxa base de churn[3] e o desconhecimento da matriz de custos e benefícios, o cálculo do valor esperado foi descartado como métrica de verificação de performance neste projeto. Pela sua capacidade de lidar com as incertezas mencionadas, a AUC-ROC[4] foi escolhida como métrica de otimização.[20]

[3] A taxa base de churn representa a quantidade de amostras positivas para *churn* no dataset fornecido.

[4] Area Under Curve Receiver Operating Characteristics

2.4 Validação

Separação de dados de treinamento e teste

Para determinar a performance do modelo de forma mais assertiva, os dados disponíveis foram inicialmente separados em dois subsets: o primeiro utilizado para treinar o modelo e o segundo para testar sua perfomance. Dessa forma, é possível identificar quão bem o modelo e os dados fornecidos contribuem para a performance de generalização, isto é, a performance em dados nunca antes vistos.

A separação foi feita em duas etapas: Na primeira, a lista de clientes disponível foi aleatoriamente dividida em duas, de modo que o mesmo cliente não apareça simultaneamente nos registros dos subsets de treino e de teste. A segunda etapa consistiu de eliminar no subset de treino um certa quantidade dos registros mais recentes para simular a passagem de tempo entre os instantes de treinamento e de aplicação do modelo. Isto é necessário pois, numa aplicação no mundo real, a passagem do tempo pode interferir em características relevantes dos dados e comportamento dos clientes. Além disso, devido ao período de determinação do target, existe um atraso de pelo menos 30 dias para que os dados gerados pelo serviço possam de fato ser utilizados para treinamento do modelo.

Performance de generalização

A performance de generalização, como dito anteriormente, representa a performance do modelo quando aplicado em dados nunca antes vistos. Ou seja, dados que o modelo não teve acesso no momento de treinamento.

Para maximizar esta performance, é necessário que o modelo se ajuste à tendência real e evite se ajustar às informações ruidosas dos dados . Em outras palavras, o modelo necessita ser complexo o suficiente para capturar todas as informações relevantes da amostra, mas precisa limitar esta complexidade de modo a não capturar também as idiosincrasias. Este desafio é conhecido como o balanço entre viés e variância.[5]

[5]O modelo ideal possui viés e variância nulos, porém a redução de uma dessas métricas natu-

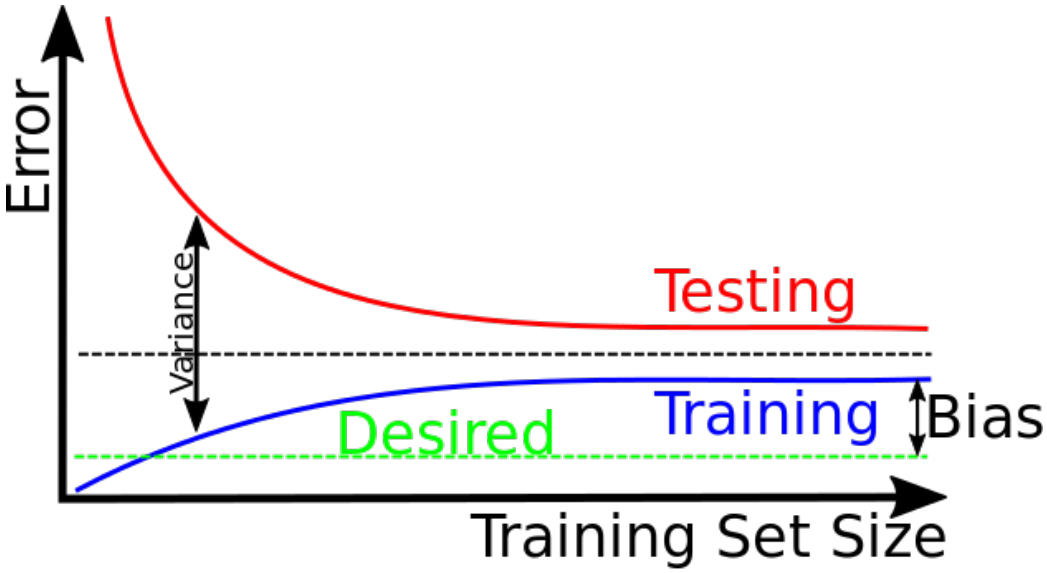


Figure 2.2: Visualizando Variância e Viés

Quando o modelo é simples demais, acaba por realizar suposições incorretas sobre a natureza dos dados. Este tipo de erro é chamado de **viés**, e provoca erros de valor elevado tanto no dataset de treino quanto no de teste.

Quando o modelo é complexo demais, acaba por se tornar muito sensível à pequenas variações (ruído) nos dados de treinamento. Este tipo de erro é conhecido por **variância** e reduz consideravelmente o erro no dataset de treino, ao custo de erros muito elevados quando aplicado em dados de teste (não vistos).

Controle de complexidade

O controle da complexidade é feito através do ajuste dos hiperparametros do modelo. No caso sendo tratado, os parametros ajustados foram a profundidade máxima da árvore, o número mínimo de amostras para um nó interno ser dividido, o valor mínimo de redução de impureza para permitir que um nó interno seja dividido.

A escolha de cada um desses hiperparametros foi feita através do módulo *GridSearchCV* presente na biblioteca scikit-learn. Este módulo implementa um

ralmente aumenta a outra, de modo que na prática deve-se encontrar um balanço que minimize o erro total.

algoritmo de busca exaustiva através de *cross-validation* com *K-folds* aplicado no subset de treinamento.

É importante notar que este tipo de validação cruzada, apesar de muito eficaz para controlar a complexidade de modo geral, não será efetiva contra as possíveis idiosincrasias temporais entre os subsets de treino e de teste.

Curva de aprendizado

Sem outras alterações, a performance de generalização normalmente melhora diretamente com a quantidade de amostras fornecidas para o aprendizado do modelo, até um certo limite.[6] O gráfico que mostra a relação entre performance e volume de dados é chamado de *Curva de Aprendizado*. Esta é uma importante ferramenta para fazer determinar o limitante da performance do modelo.

[6]Este limite não ocorre em modelos baseados em redes neurais

3. Resultados

O principal resultado deste trabalho foi uma melhora significativa da compreensão de conceitos e processos de ciências de dados e engenharia de software. Durante o desenvolvimento do projeto, muito empenho foi dedicado para estruturação do código de modo que o projeto possa crescer de forma escalável. Em relação aos algoritmos de aprendizado de máquina, conceitos importantes como viés, variância, performance de generalização e ganho de informação foram adquiridos e aplicados na construção do modelo.

O resultado secundário, mas também relevante, foi a criação de um modelo possivelmente eficaz para impactar positivamente o programa de retenção de clientes da empresa concedente.

3.1 Criação de modelo preditor de churn

Feature Relevance

Totas as features identificadas como relevantes estão listadas abaixo em ordem decrescente de importância, sendo as duas primeiras responsáveis por uma redução de impureza próximo de 91% da redução total obtida.

1. recency_v1
2. frequency_norm
3. TS1 index_mass_quantile__q_0.4

Table 3.1: Melhores hiperparametros encontrados

max_depth:	4
min_impurity_decrease:	0.001
min_samples_split:	50
demais parametros:	valor padrão

4. TS2 index_mass_quantile__q_0.9

5. gravity

6. TS3 approximate_entropy__m_2__r_0.1

TS1, TS2 e TS3 são 3 das séries temporais obtidas do histórico de pedidos de cada cliente. O último termo de cada uma dessas linhas refere-se ao tipo de característica extraída desta série temporal pela biblioteca *tsfresh*. Por tratar-se de informação estratégica e fortemente relacionada com o indústria em que a empresa está inserida, a descrição destas séries será mantida em anonimato.

Características da Árvore de Indução

A árvore de indução gerada possui 4 níveis de profundidade e 17 nós, sendo 9 'folhas' e 8 nós intermediários.

Performance da classificação

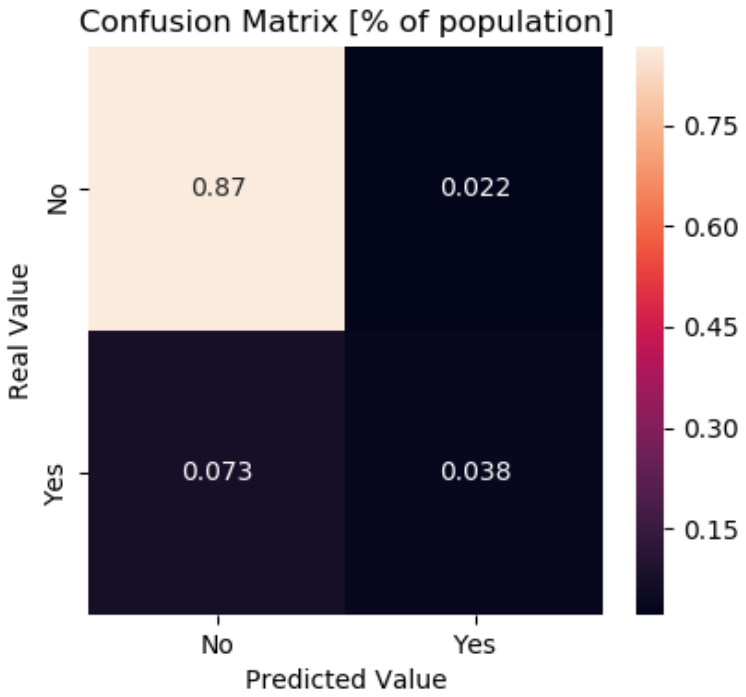
A matriz de confusão (figura 3.1) deixa evidente a taxa base de churn dos dados utilizados (próximo de 11%). Além disso, ela permite facilmente calcular a precisão e a sensibilidade do modelo.

$$\frac{TruePositives}{\sum_P redictedConditionPositives} \quad \text{(precision)}$$

$$\frac{TruePositives}{\sum_R ealConditionPositives} \quad \text{(sensitivity)}$$

$$2 * \frac{precision * sensitivity}{precision + sensitivity} \quad \text{(f1\_score)}$$

Figure 3.1: Matriz de Confusão do modelo encontrado



Podemos perceber que o modelo ajustado classifica corretamente aproximadamente 34% das amostras positivas (sensibilidade) e, dentre as amostras classificadas como churn, acerta 63% (precisão). Isso equivale a um f1 score de 44

Para fins de comparação, um classificador aleatório que sempre prediz a classe positiva obteria uma precisão próxima de 0.11 (taxa base de churn dos dados analisados) e f1_score próxima de 0.20

Ao observar a curva de aprendizado (figura 3.3) podemos dizer que a capacidade de aprendizado do modelo fica saturada acima de 1000 amostras, não sendo capaz de melhorar sua performance com mais dados a partir deste ponto. Apesar disso, a performance está bem próxima do classificador ideal (AUC = 1.0) e a variação de performance média entre treino e teste tem valor moderado, indicando que o modelo está com sua complexidade ajustada corretamente e sofre níveis aceitáveis de viés e variância.

Figure 3.2: Curva comparando o balanço entre precisão e sensibilidade

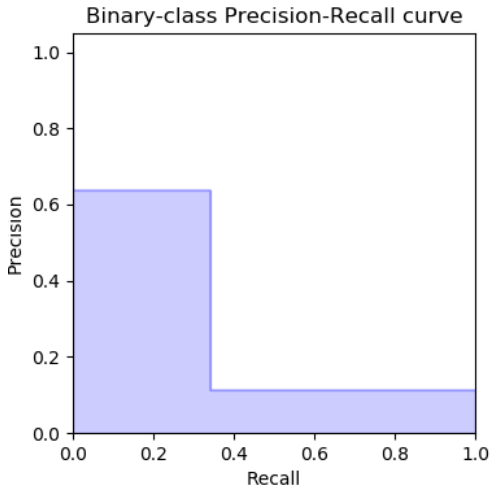


Figure 3.3: Performance vs Quantidade de amostras fornecidas

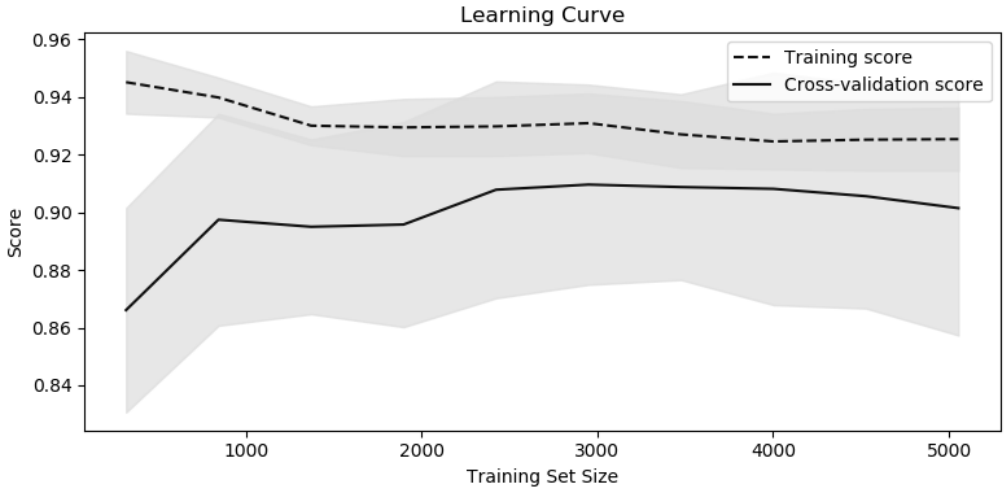
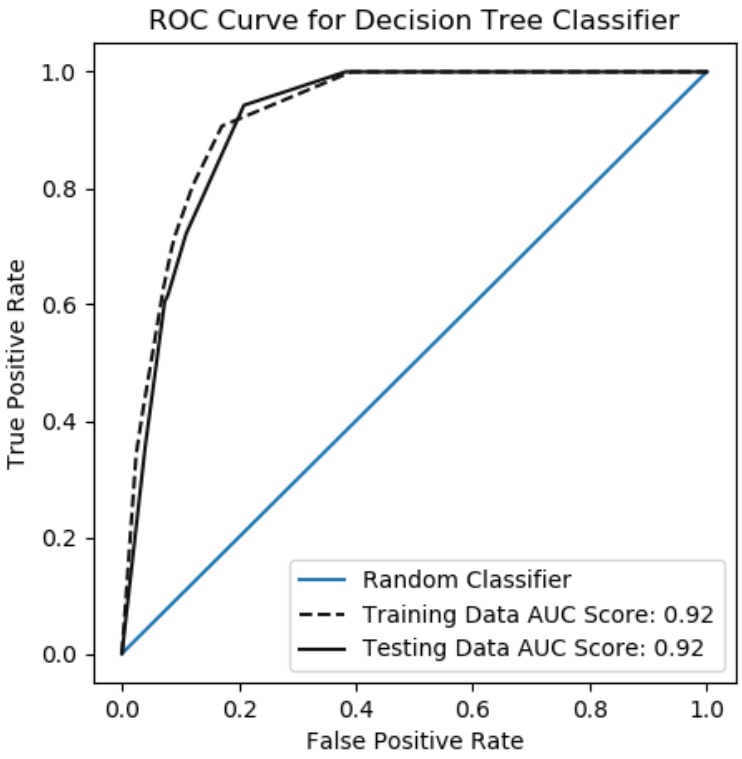


Figure 3.4: ROC: Receiver Operating Characteristics



3.2 Trabalho futuro

Engenharia de Features

- Pesquisar a relevância de features que representem os efeitos de rede para prever o churn
- Utilizar o modelo BG-NBD como estágio intermediário para criação de features para o modelo de previsão

Modelagem

- Alinhar a métrica de otimização com o resultado financeiro da empresa
- Testar a performance de outros modelos de classificação

Validação

- Implantar e validar o modelo em uma situação de mundo real
- Tornar o modelo mais robusto contra tendências temporais utilizando nested cross-validation juntamente com uma forma de penalizar registros de eventos mais antigos

4. Conclusão

O projeto se mostrou interessante desde o princípio. A oportunidade de trabalhar em um desafio de negócios real e expandir os conhecimentos adquiridos durante a graduação foi aproveitada com grande entusiasmo. Foi possível aplicar ferramentas de desenvolvimento em alta demanda no mercado e praticar de forma simultanea conhecimentos de negócios, data mining e de engenharia de software. Apesar do pouco tempo disponível para a elaboração e implementação do projeto, o resultado final foi bastante satisfatório, pois cumpriu com a proposta. O modelo, apesar de ainda não estar pronto para ser implantado em ambiente de produção de forma automatizada, já pode ser utilizado para gerar insights e direcionar o fluxo decisório da empresa. Espera-se que os métodos aplicados aqui e o software desenvolvido sejam utilizados como base para trabalhos futuros não só na área de previsão de churn, mas também em outros problemas de machine learning.

Referências

[1] SHAABAN, Essam et al. **A Proposed Churn Prediction Model**. International Journal of Engineering Research and Applications (IJERA) v.2, n.4, June-July 2012, pp.693-697, 2012

[2] UMAYAPARVATHI, V.; IYAKUTTI, K. **Automated Feature Selection and Churn Prediction using Deep Learning Models**. International Research Journal of Engineering and Technology, v.04, n.03, Mar. 2017

[3] AHMED, Ammar; LINEN, Maheswari. **A review and analysis of churn prediction methods for customer retention in telecom industries.** Rathnavel Subramaniam College of arts&science, 2017

[4] SUBRAMANYA, Karthik. **Enhanced feature mining and classifier models to predict customer churn for an e-retailer**. Graduate Theses and Dissertations, 2016

[5] WIT, L. S. **An Analysis of Non-Contractual Churn in the B2B Hotel Industry**. Master thesis: Data Science: Business and Governance 2017

[6] ZHAO, Xue. **Research on E-Commerce Customer Churning Modeling and Prediction**. International Education College, Nanyang Institute of Technology, Nanyang, 2014

[7] HAGIU, Andrei; WRIGHT, Julian. **Multi-Sided Platforms**. Working Paper - Harvard Business School, 2011

[8] EVANS, David S.; SCHAMLENSEE, Richard. **Failure to Launch: Critical Mass in Platform Businesses**. Review of Network Economics, 2010

[9] MALIK, Om. **In Silicon Valley Now, It's Almost Always Winner Takes All**. Disponível em: https://www.newyorker.com/tech/elements/in-silicon-valley-now-its-almost-always-winner-takes-all. Acesso em 03 dez. 2018

[10] EISENMANN, Thomas; et al. **Strategies for Two-sided Markets**. 2006

[11] HAGIU, Andrei; WRIGHT, Julian. **Multi-Sided Platforms**. Harvard Business School, 2015

[12] HARVARD BUSINESS SCHOOL. **The Five Forces**. Disponível em: https://www.isc.hbs.edu/strategy/business-strategy/Pages/the-five-forces.aspx. Acesso em 03 dez. 2018

[13] UENLUE, Murat. **Strategy: Porter's Five Forces explained (plus example Uber)**. Disponível em: https://www.innovationtactics.com/porter-five-forces/. Acesso em 03 dez. 2018

[14] GRIFFIN, Tren. **Two Powerful Mental Models: Network Effects and Critical Mass**. Disponível em: https://a16z.com/2016/03/07/network-effects_critical-mass/. Acesso em 03 dez. 2018

[15] MAYO, Matthew. **Data Preparation Tips, Tricks, and Tools: An Interview with the Insiders**. Disponível em: https://www.kdnuggets.com/2016/10/data-preparation-tips-tricks-tools.html. Acesso em 03 dez. 2018

[16] PYLE, Dorian. **Data Preparation for Data Mining**. San Francisco: Morgan Kaufmann Publishers, 1999

[17] FADER, Peter et al. **"Counting your Customers" the Easy Way:** An Alternative to the Pareto/NBD Model, 2005

[18] TASHMAN, Leonard. **Out-of sample tests of forecasting accuracy:** a tutorial and review. International JOurnal of Forecasting. Janeiro, 2000

[19] ZHANG, Bo; WANG, Liwei. **Application of Survival Analysis for Predicting Customer Churn with RFM**, 2017

[20] PROVOST, Foster; FAWCETT, Tom. **Data Science for Business**: What You Need to Know About Data Mining and Data-Analytics Thinking. Sebastopol: O'Reilly Media, 2013.

[21] ECONOMIDES, Nicholas et al. **Dynamic Oligopoly with Network Effects**. 2005