PlayKids

Churn Predictor

Gabriel Ghellere
Robson Kitano


ifood
movilepay
Pague com ifood
zoop
PlayKids
movile
wavy

PlayKids


Desenhos
Livros


CRAFTY
COLORS


Crie o que quiser!


Leiturinha


PlayKids
EXPLORER


PLAY AND CHAT WITH FRIENDS!!

Métricas e Indicadores

Como bolamos estratégias, gerenciamos, e operamos nosso negócio no dia a dia usando dados a nosso favor?

Modelo de Negócio: Assinatura

- Base e Vendas
- Ticket Médio
- LTV/CAC
- **Churn**

Por que o Churn é tão importante

- Impacto direto em Receita;
- Custo de aquisição de novos usuários é mais alto do que o de retenção;
- Impacto direto na relação LTV/CAC.

Projeto Preditor de Churn

Como poderíamos analisar os comportamentos e prever quem vai cancelar a assinatura?

Objetivos do Projeto:

- Identificar quais usuários são mais propensos a churnar e tomar medidas preventivas;
- Identificar quais os fatores que influenciam o churn (causa raiz).

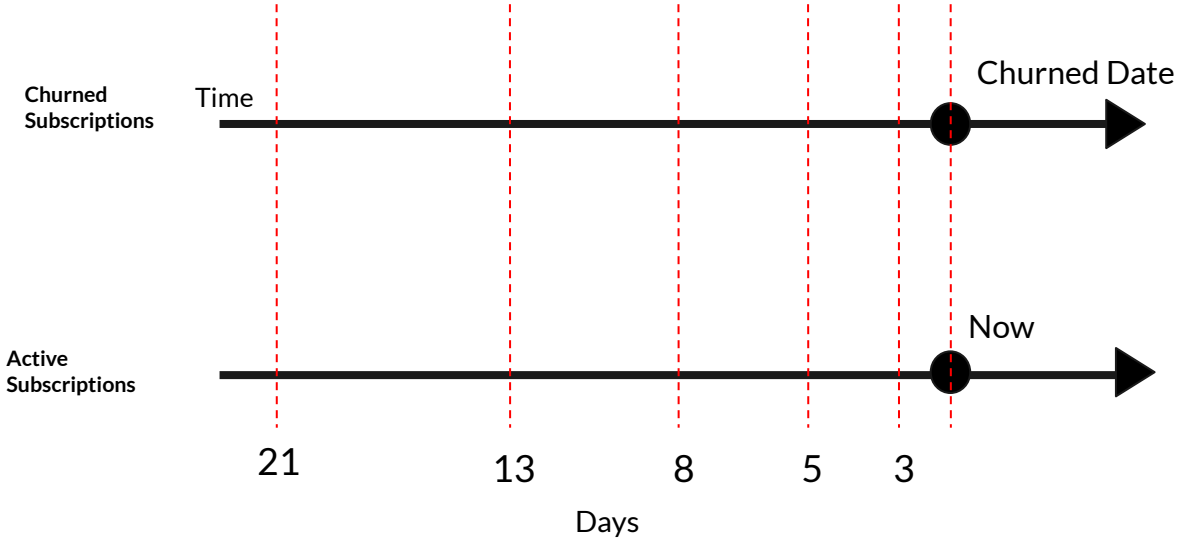
Criando a primeira versão do Preditor de Churn

- **Modelo baseado em estatística tradicional (sem aprendizado de máquina)**
- **Regressão Logística**
- **Por quê não fomos bem sucedidos?**
  - Mudanças constantes na App faziam que o preditor perdesse poder de predição rapidamente;
  - A atualização do modelo era lenta e custosa;
  - Cometemos erros no processo de **feature engineering**;
  - Custo para atualização do modelo inviabilizou iterações de melhoria no modelo gerado;

Features

Dados de entrada:

- Dados de assinatura: Período de renovação, idade da assinatura, plataforma de pagamento, dispositivo, País, recência, etc.
- Eventos dos usuários: Open, Play Video, Play Game, Play Book (count, tempo, dias distintos)



Feature Engineering

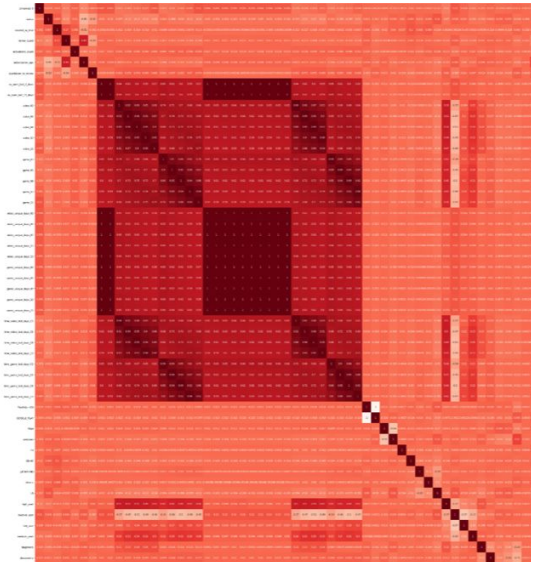
Técnicas utilizadas

- Criação de features temporais (exemplo idade da assinatura)
- Pensar além do One Hot encode
  - Cuidado com K-1 dummy variables
  - Tratamento de labels raras
  - Feature Hashing
  - Count e Frequency encoding
- Balanço entre robustez e precisão
  - Binarização, Bucketização
- Transformações numéricas
  - Quantile transformation

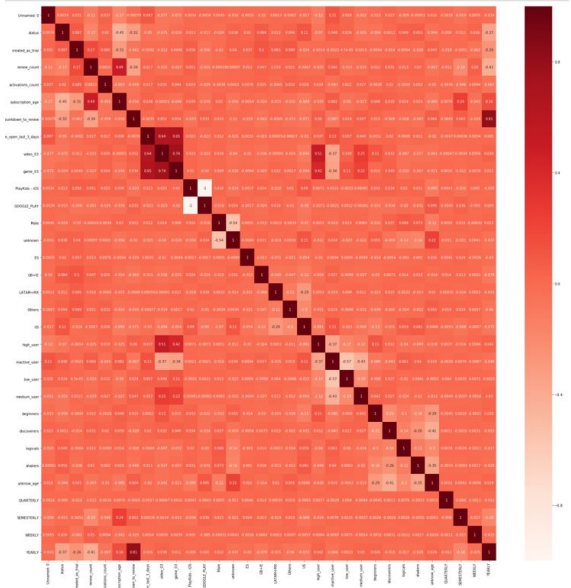
Feature Selection

Removendo features correlacionadas entre si

Antes



Depois

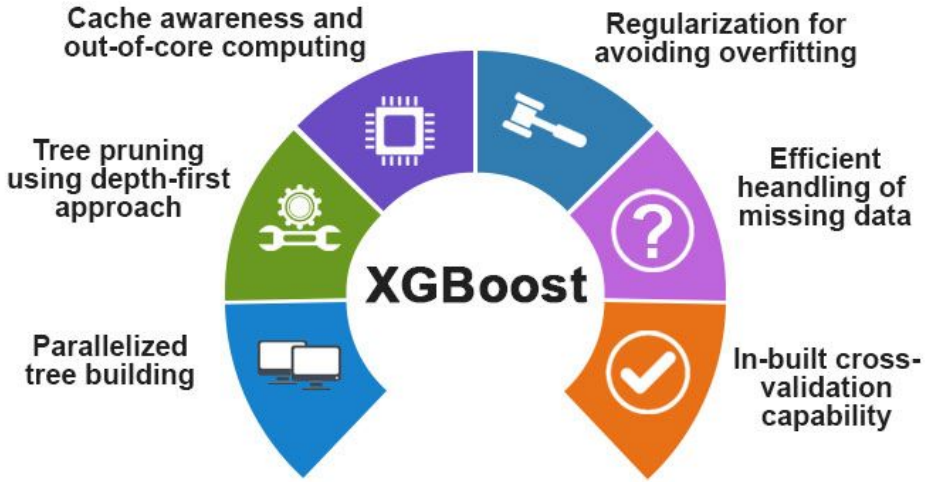


Algoritmos

- **Aprendizado de máquina supervisionado. Algoritmos Testados:**
  - Logistic Regression
  - Random Forest
  - Redes Neurais Keras
  - **Boosting Models**
- **Tecnologias:** Python, Scikit-Learn, Tensorflow with Keras
- **Vantagens:**
  - Maior precisão
  - Maior velocidade na geração, evolução e manutenção dos modelos

XGBoost


eXtreme Gradient Boosting
custom tree building algorithm
XGBoost
by Tianqi Chen
Used for:
• classification
• regression
• ranking
with custom loss functions
Interfaces for Python and R, can be executed on YARN


Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient heandling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Dificuldades - Bias Variance Tradeoff

Lista de items para reduzir overfitting XGBoosting (Reduce Variance):

- Subsampling
- Bootstrap Aggregation
- Aumentar o número de iterações e diminuir learning rate.
- Max depth of trees (max 4)
- Early Stop
- Medir bias-variance trade-off usando k-fold cross validation e aplicar GridSearch nos hiperparâmetros.

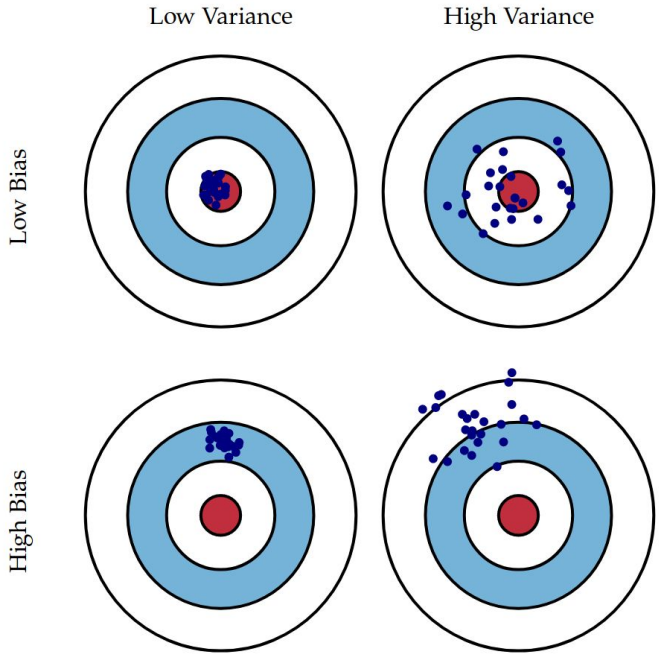


Fig. 1 Graphical illustration of bias and variance.

Hyperparameters

Parâmetros fixos:

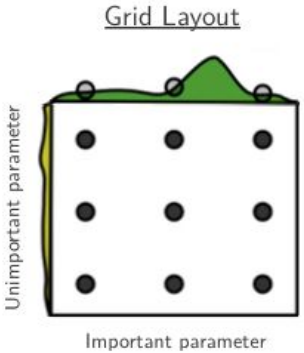
- Max depth of trees
- Early Stop
- Learning rate

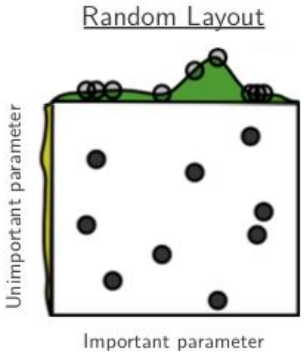
Parâmetros otimizados:

Eta, colsample_bytree_vals, gamma,

learning_rate, n_estimator, subsample…

Método: RandomizedSearchCV com 10 CV





Importância da interpretabilidade do modelo

Relevância / Peso

Tempo de Assinatura

[...]

Engajamento Vídeo

[...]

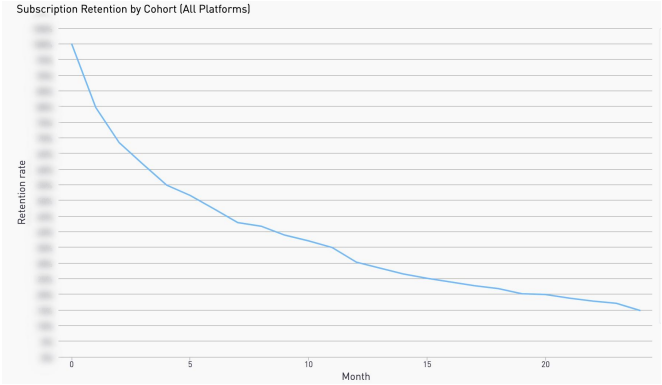
Engajamento Games

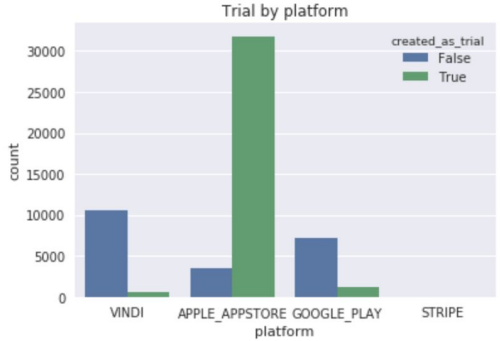
[...]

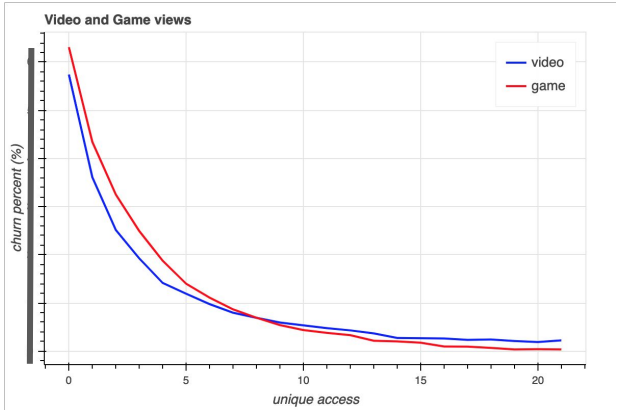
[...]

[...]

Trial







Importância da interpretabilidade do modelo

Uso de árvore de decisão para obter insights:

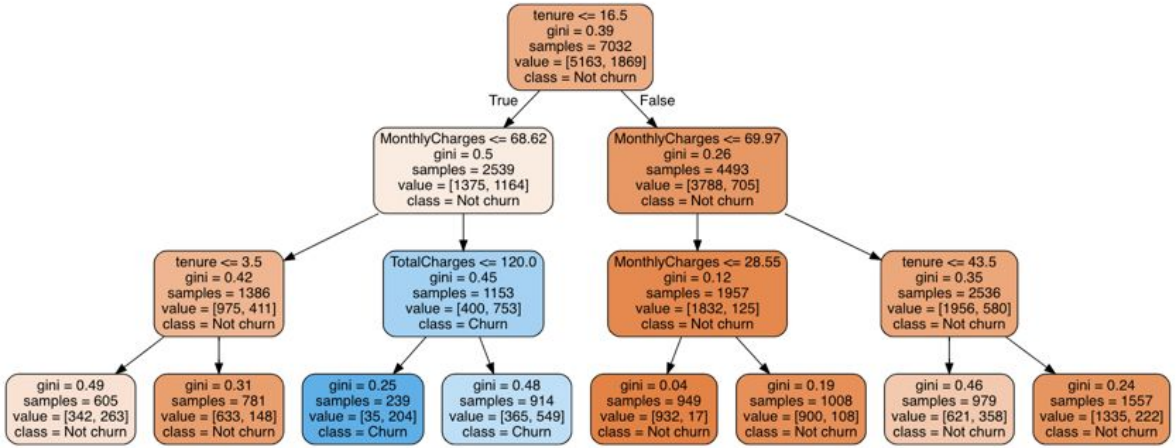
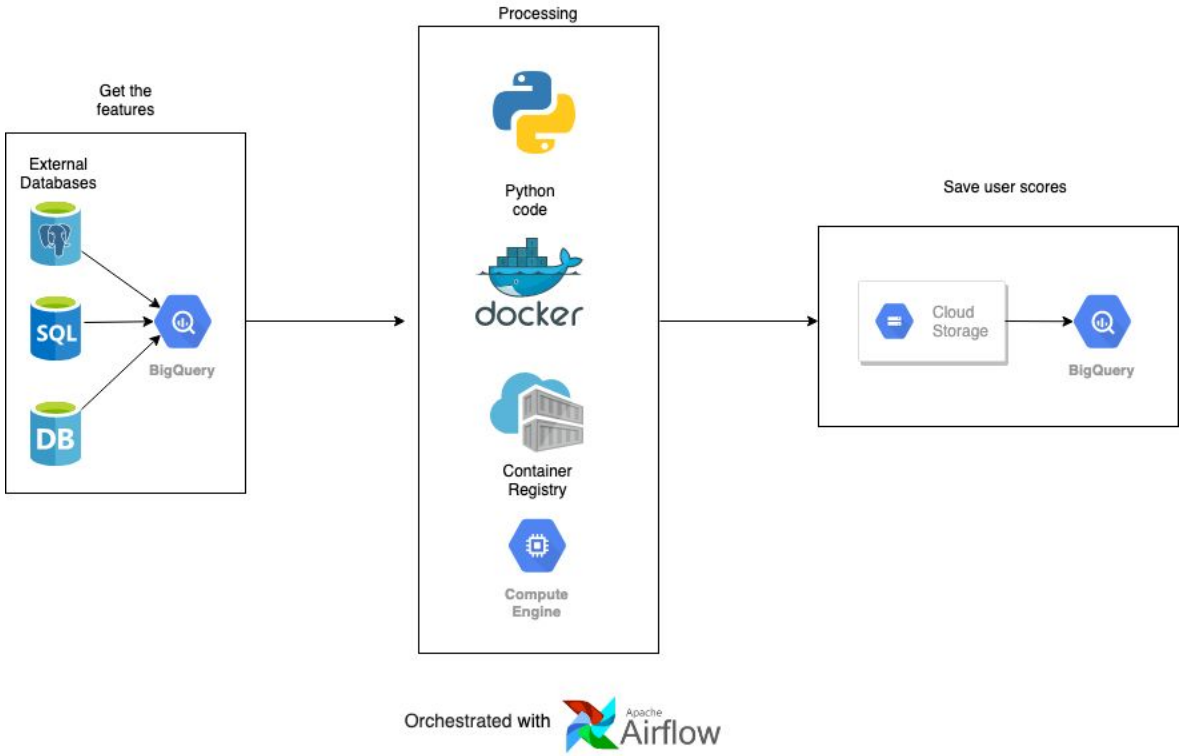


Image from Kaggle

https://www.kaggle.com/pavanraj159/telecom-customer-churn-prediction

Deploy - Batch



Lições Aprendidas

- É muito importante conhecer bem o seu modelo (premissas, funcionamento, fragilidades, fit com o seu problema, etc);
- É necessário trabalhar bem o seu dataset para tirar o melhor proveito de cada modelo ou então usar o melhor modelo para o seu dataset/problema;
- Não se apegar apenas a precisão do seu modelo. O seu business vai determinar se tolerará mais falsos positivos ou falsos negativos;
- A maior parte do tempo é gasto trabalhando o dataset. O trabalho de feature engineering é uma das (ou a etapa) mais importante do desenvolvimento do seu modelo, e foi responsável pelos maiores saltos de melhora de acurácia no processo de desenvolvimento;
- O processo de análise exploratória dos dados pode gerar tanto valor quando o modelo em si.

Obrigado!

PlayKids