MO444 - 2018s1 - Atividade Final

Carlos Henrique Baia e Silva
Paulo Fernando Ozório Ferraz
Anderson Rocha

1. Introdução

Este trabalho visa explorar soluções para o problema da competição *Data Science Game 2018*. O desafio consiste na criação de um modelo capaz de predizer o interesse de um cliente na compra ou venda de títulos corporativos em uma determinada semana.

Os dados disponíveis são do banco BNP Paribas o qual realiza o serviço de transações dos títulos entre os clientes e que através de modelos de aprendizado de máquina faz sugestões de qual título será interessante em comprar ou vender para seus clientes.

2. Soluções propostas

Analisar os dados disponibilizados, realizar diferentes formas de extração das features, utilizar diferentes tipos de modelos, utilizar redução de dimensionalidade, criar uma base de validação que represente o mais próximo possível a base de teste.

3. Experimentos e discussões

- Descrição da base

A base de dados disponibilizada está dividida em 6 arquivos: *Trade*, *Challenge*, *Isin*, *Customer*, *Market* e *Macro Market*. Que serão descritas a seguir.

1. *Trade*

Dados base para o treino. Este arquivo possui o histórico de transações de títulos dos clientes no período de 1 de janeiro de 2016 à 22 de abril de 2018. Consiste em 6.762.021 *samples* informando o *status* da transação e qual foi o interesse de compra ou venda de 3.439 clientes em 27.305 títulos diferentes.

O interesse de compra e/ou venda de um cliente em um título em específico é dado como 1 quando houve o interesse e 0 quando não houve. O interesse é a classe a ser prevista na base de teste. A distribuição do interesse pela direção da transação (compra ou venda) pode ser visto no gráfico a seguir.

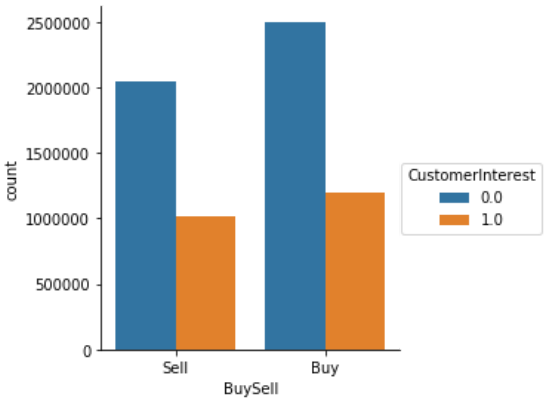

Gráfico 1: Distribuição do interesse

Pode-se notar que existem mais interações de compra do que de venda e que também na maior parte da vezes não existe o interesse na transação, o que gera um desbalanceamento nas classes.

As interações estão distribuídas dia a dia e como dito anteriormente são 1 de janeiro de 2016 até 22 de abril de 2018, conforme gráfico:

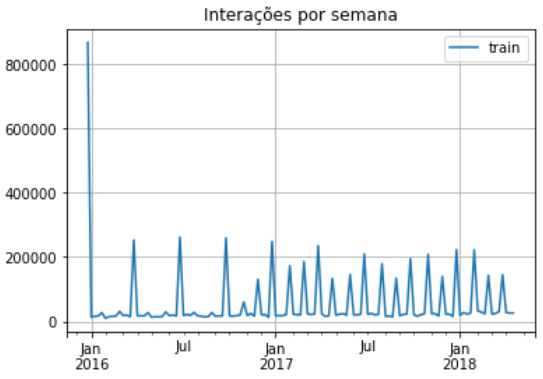

Gráfico 2: Interações por semana

Existe um pico desproporcional de transações na primeira semana composta somente por interesses zeros. Os demais picos menores são no último dia dos mês, que segundo informações da competição, mensalmente alguns clientes tem obrigação legal de prestar contas e isso gerar uma interação que não se converte em compra ou venda, sendo sempre de interesse zero.

A chave única dessa tabela é a junção do *id* do cliente com o *id* do título o qual juntos serão usados para a combinação com as demais tabelas.

2. *Challenge*

Dados da base de teste o qual deve-se prever o interesse do cliente de comprar ou vender um título.

Diferentemente da base de treino, a base de teste não é composta por entradas dia a dia, mas sim agrupada em uma única semana, que é a semana seguinte da última entrada na base de treino, no caso, a semana que se inicia 23 de abril de 2018.

Um ponto importante da base de teste é que diferentemente da base de treino que possui por volta de 15 a 20 mil transações por semana podendo ter ou não transações de compra e venda, a base de teste possui um total de 484.748 entradas referentes a semana do dia 23 de abril, sendo que para cada chave cliente/título há sempre uma entrada para a predição de compra e sempre uma para a predição de venda.

As 484.758 entradas da base de testes são formadas por diferentes combinações de 2.495 clientes e 18.265 títulos diferentes sempre em ambas as direções compra e venda como mostrado no gráfico a seguir.

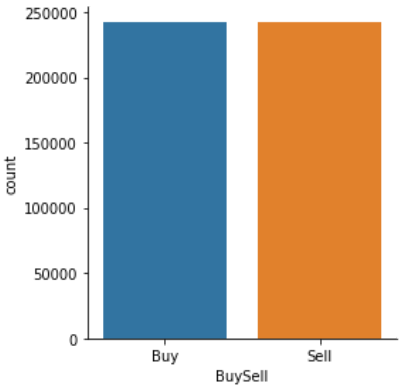

Gráfico 3: Distribuição da base de teste

A distribuição exatamente igual de 242.379 para cada direção e a discrepância quanto a base de treino indicou uma possível manipulação da base de teste o que posteriormente foi confirmado no fórum da competição pelos organizadores, o que traz o desafio de criar uma base de validação capaz de simular um comportamento próximo da base manipulada do teste. A criação da base de validação será explicada mais adiante.

Conforme dito anteriormente, o gráfico abaixo demonstra a diferença da quantidade de transações por semana entre a base de treino e a base de teste.

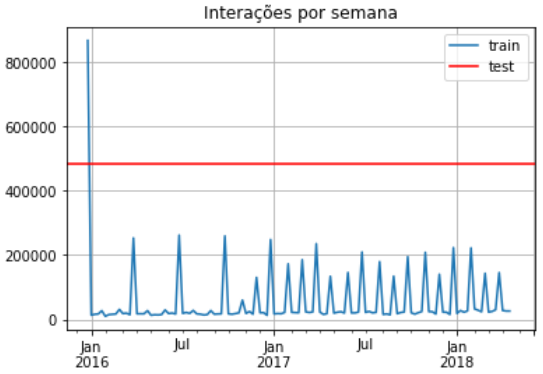

Gráfico 4: Interações por semana (treino x teste)

3. *Isin*

Dados referentes aos títulos. Essa base é composta por 16 características relacionadas a 27.411 diferentes títulos existentes, sendo 12 destas características do tipo categóricas, como *Currency* (a moeda relacionada ao título: *USD*, *EUR*, ...), *MarketIssue* (classificação padrão do

título no mercado: *Domestic*, *Global*, …) entre outras.

4. *Customer*

Dados referentes aos clientes. Essa base é composta somente por 4 características de 3.471 clientes existentes, sendo elas setor, subsetor de atuação, região e país de origem, todas também do tipo categóricas.

5. *Market*

Dados referentes aos valores dos títulos durante o período do treino. Essa base contém 3 informações numéricas relacionadas aos valores de cada um dos títulos dia a dia em todo período analisado.

6. *Macro Market*

Dados com informações financeiras diárias relacionadas ao mercado no geral, como índices de bolsas de valores e valores de ações. Alguns desses índices são representações do mesmo valor em diferentes câmbios.

No total são 111 índices numéricos que representam o mercado com um todo dia a dia. De forma a não aumentar exageradamente o número de features do modelo, foram utilizadas algumas técnicas para agrupar e/ou escolher alguns desses índices. A opção com melhor resultado foi o uso do PCA, assim reduzindo as 111 dimensões para 5, retendo 99,89% da informação, como apresentado no gráfico abaixo.

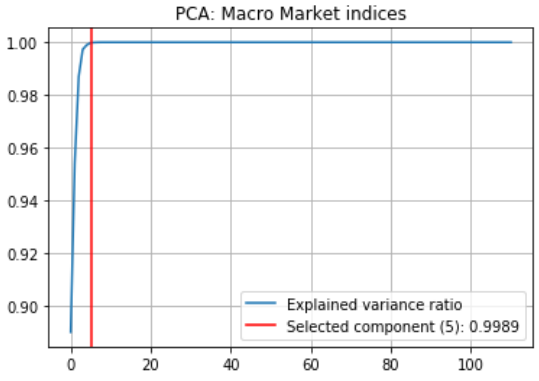

Gráfico 5: Resultado do PCA

- Base de treino

Seguinte o mesmo padrão existente na base de teste, os dados de treinamento foram agrupados por semanas ao invés de manter em dias. O agrupamento foi feito baseado no trio cliente, título e compra/venda. Caso um mesmo trio apareça com interesse 0 e 1 em uma mesma semana, prevalece o interesse 1, uma vez que o importante no teste é ter ocorrido o interesse no decorrer da semana e para evitar duas entradas classificadas com *labels* diferentes.

- Base de validação

Um dos principais problemas encontrados no início foi a geração de uma base de validação consistente que fosse uma boa base de comparação com as submissões feitas na plataforma do *Kaggle*.

Por se tratar da previsão da semana seguinte, tornou-se mais coerente utilizar a última semana da base de treino como validação. Porém, conforme visto no gráfico 4 o volume de entrada na base de teste é muito maior que todas as semanas da base de treino, refletindo diretamente na métrica utilizada para medir o modelo, assim, o resultado obtido na base de validação sempre se apresentava muito distante do resultado obtido nas submissões no *Kaggle*.

Dessa forma, optou-se por manipular a última semana da base de treino no intuito de deixá-la mais parecida com a base de teste. A abordagem utilizada foi copiar a base de teste, preencher com interesse 1 todas as chaves cliente-título existentes na última semana da base de treino e preencher todo os restante faltante com zeros. Esta abordagem foi utilizada devido vários testes, discussões e análises dos dados, levando a concluir que a base de teste é quase toda formada por interesses zeros, sendo o principal desafio acertar os poucos uns.

Após a geração da validação foi possível conseguir um resultado do *score*, não ideal, mas um pouco mais próximo ao da

submissão no *Kaggle*, com uma variação próxima de 0.06 acima da submissão. A métrica utilizada na competição é *Area Under the Receiver Operating Characteristic Curve*.

- Tratamento dos dados

1. Nulos

As bases são compostas por dados categóricos e numéricos, dados quais podem possuir valores nulos em diferentes colunas.

Dentre os dados categóricos, foi testado tratar o nulo como vazio e com o valor mais frequente (moda), tendo a utilização do vazio como melhor resultado. Os numéricos foram testados com a mediana e com valor zero. O melhor resultado foi obtido utilizando o valor zero, além de ser coerente com o tipo de informação das colunas.

A tabela *Market Data* foi a única a ser tratada de forma diferente. Ela é composta por índices como os da bolsa de valores, porém em diversas datas não existiam as medições.

Nesse caso o uso de zeros poderia indicar que aconteceu uma variação brusca naquele índice, o que não seria verdade. Como os dados apresentados são uma sequência temporal, foi tomada a decisão de repetir o valor do dia anterior no dia seguinte caso esse seja nulo, evitando uma flutuação irreal nos índices. A abordagem utilizada pode ser vista na tabela abaixo.

Vale ressaltar que os dados nulos foram tratados somente em modelos que não fazem uso de árvore.

Data	Índice 1 (antes)	Índice 1 (depois)
01-01-17	12,25	12,25
02-01-17	NaN	12,25
03-01-17	12,55	12,55
04-01-17	12,58	12,58

Tabela 1: Tratamento de valores numéricos

2. Dados categóricos

Algumas diferentes abordagens foram tomadas de acordo com o modelo que foi utilizado. As técnicas aplicadas foram a *One Hot Encoding* que separa a característica em colunas binárias para cada valor, a *Label Encoding* que cria *alias* numéricos para as categorias e a utilização dos valores reais em modelos que usam árvores e que suportam valores do tipo texto como entrada para o modelo, não sendo necessário o tratamento dos mesmos.

- Modelos

Após os tratamentos e discussões descritos anteriormentes sobre as bases de treino, validação e teste, diferentes modelos e abordagens foram aplicados sobre as bases construídas com o intuito de fundir estes resultados de modelos completamente distintos a fim de alcançar melhores resultados.

A métrica da validação definida pela competição foi a Area Under the ROC Curve (AUC).

As abordagens utilizadas serão apresentadas a seguir.

1. *CatBoost*

CatBoost é um algoritmo baseado em árvores de decisão e que faz um tratamento diferenciado em dados categóricos.

- Extração de features

Neste modelo foi feita a fusão entre a base de treino e as tabelas de clientes e títulos. Foram utilizadas somente as features categóricas, gerando um total de 18 features.

- Modelo

Dentre os testes realizados o melhor resultado obtido foi utilizando parte dos dados de treinamento (últimos 6 meses), 200 iterações, taxa de aprendizado de 0.15, profundidade 6 e 5 modelos trocando a semente.

- Resultados

Com a média normal das 5 predições foi alcançado o *score* de 0.746 na validação e 0.673 no *Kaggle*. Outro método utilizado foi o de aplicar a potência de 0.5 nas predições e em seguida fazer a média, essa abordagem elevou o score no *Kaggle* para 0.695.

2. *XGBoost*

- Extração de features

Conforme mencionado anteriormente a tabela *Macro Market* teve os 111 índices reduzidos para 5 usando PCA, porém essa tabela é diária e os dados de treinamento foram agrupados por semana, consequentemente os 5 componentes foram agrupados por semana também. As seguinte funções foram aplicadas no agrupamento: média, mínimo, máximo, desvio padrão e a diferença entre o primeiro e último dia da semana. Essas 5 operações aplicadas nos 5 componentes gerou um total de 25 novas *features*.

Outras *features* foram geradas olhando o histórico do par cliente-título desde a data inicial até o momento da transação. Foi feita a soma de quantas vezes o cliente comprou ou vendeu o título, quantas vezes teve interesse, quantas não teve interesse, etc. Se mostraram *features* interessantes, pois no geral para vender um título o cliente deve ter comprado ele antes em algum momento. Também foram geradas *features* olhando a histórico de compra e venda do cliente em questão com todos os títulos.

Após diversos refinamentos e testes com diversas *features*, ficou um total de 88 features, sendo 18 categóricas (utilizando Label Encoding) e 70 numéricas.

- Modelo

Devido a limitações de recursos computacionais e longo tempo no treinamento, foi utilizado apenas o último 1 ano no treinamento. Para achar os melhores parâmetros para o modelo, foi executado uma otimização bayesiana, sendo esta uma espécie de *grid search*, porém ao invés de testar diversas combinações por força bruta, ela faz um processo de otimização para buscar os melhores valores dentro de um intervalo no menor número de interações possível.

O melhor modelo encontrado foi com 100 estimadores, coeficiente de aprendizado 0.1 e profundidade 3.

- Resultados

Esse modelo apresentou um resultado de 0.711 na validação e 0.670 no *Kaggle*.

Outros modelos foram implementados e testados e os resultados podem ser vistos na tabela abaixo.

Modelo	**AUC**
CatBoost	0.6955
XGBoost	0.6706
Light GBM	0.5955
Extra Trees	0.5924
Random Forest	0.5662
Rede Neurais	0.5439

Tabela 2: Resultado dos modelos

3. *Fusão dos modelos*

Cada algoritmo tem suas particulares de aprendizado e podem se tornar mais assertivos para diferentes tipos de entradas, logo a combinação de diferentes modelos pode gerar um ganho significativo na assertividade.

Como parte dos modelos não tiveram um bom desempenho, foram selecionados o *CatBoost* e *XGboost* e aplicado diferentes tipos de fusão. O melhor resultado foi aplicando a fusão de ranqueamento, técnica que transforma as predições em ranques, faz a média desses valores e por final os normaliza entre 0 e 1. Essa fusão gerou um ganho relativamente significativo como visto na tabela abaixo.

Modelo	AUC
CatBoost	0.6955
XGBoost	0.6706
Ensemble	0.7101

Tabela 3: Fusão de ranqueamento

4. Trabalhos futuros

Os próximos passos serão testar duas novas abordagem para esse problema:

1. Long short-term memory (LSTM)

Tipo de rede neural recorrente com bons resultados em problemas que possuem séries temporais. Devido ao fato que todo cliente tem um histórico de vendas e compras, talvez seja possível extrair um comportamento dele no decorrer do tempo, a fim de validar essa ideia será testado o uso de uma LSTM na sequência de *features* que ele possui no tempo.

2. Sistema de recomendação

Os 3.439 clientes e 27.305 títulos diferentes podem ser abordados como um sistema de recomendação, uma vez que os títulos são produtos com suas características específicas. A recomendação para cada entrada da semana de teste, pode indicar uma forte probabilidade de compra/venda.

5. Conclusão

O desafio se mostrou bastante complexo devido a natureza do problema, pouca informação sobre o negócio e dificuldade em construir uma validação consistente.

O *baseline* da competição é de 0.7955, depois de diversos modelos, abordagens na geração da base de treinamento, extração de *features* e técnicas de fusão, o melhor valor alcançado foi de 0.7101, bem distante do *baseline*. Visto isso, algo ainda se passa despercebido, *features* importantes que não estão sendo encontradas ou talvez abordagens e visão errada do problema, como por exemplo aplicar soluções como um sistema de recomendação como dito anteriormente.

Baseado nos experimentos citados dentre os modelos abordados, o *CatBoost* se mostrou o mais promissor, acredita-se ser pelo fato de o problema possuir muitos dados categóricos e esse modelo ser especialista nisso.

O uso de *ensemble* de ranqueamento também deu um ganho no resultado mesmo combinando o melhor modelo com um modelo inferior, o que mostra que diferentes abordagens combinadas podem gerar ganho significativo no resultado.

Concluímos também que achar um validação adequada faz toda a diferença na solução de um problema e esse foi um dos grandes desafios enfrentados devido ao fato de base de teste ter sido manipulada.

Com o objetivo de melhorar o resultado planeja-se testar as duas novas abordagens anteriormentes citadas e continuar no processo de *feature engineering* gerando *features* mais significativas, assim como eliminar *features* não tão importantes a fim de diminuir a complexidade do modelo, facilitar o aprendizado e aumentar a assertividade.