Aplicando Aprendizado de Máquina em uma competição do Kaggle

Abstract

A proposta deste artigo é demonstrar técnicas de aprendizado de máquina em um problema do mundo real. Para isso analisaremos a mais recente competição promovida pela Expedia no site Kaggle. Demonstramos o uso de técnicas de regressão logística para classificação multi-classe e analisamos a viabilidade da mesma para o problema proposto.

1. Introdução

Planejamento de férias sempre demanda muito cuidado e na maioria das vezes gera dor de cabeça para o consumidor na hora de escolher o Hotel no qual vai se hospedar. Visando melhorar a experiência do consumidor, diversos tipos de serviços surgiram com o objetivo de facilitar e esclarecer a incógnita que é escolher um hotel em uma cidade na qual não conhece.

Um destes serviços oferecidos é o site Expedia, onde são ofertados preços de hospedagem, aluguel de carros, pacotes de viagens e voos baratos. Recentemente a Expedia lançou no Kaggle um desafio: a partir dos dados de comportamento do usuário no site prever em qual cluster de hotéis está o hotel que o usuário fará sua reserva. Kaggle é uma plataforma fundada em 2010 para competições de modelagem preditiva e analítica onde empresas e pesquisadores postam seus dados para que data miners do mundo inteiro possam competir sobre quem gera os melhores modelos.

Neste desafio eles disponibilizaram dados da navegação do usuário no site, incluindo o que o foi pesquisado, como eles interagiram com o resultado da pesquisa, se a pesquisa resultou em um pacote de viagem, entre outros. Eles também já disponibilizaram os clusters dos hotéis, onde hotéis similares estão agrupados.

Foram disponibilizados 4 arquivos para serem utilizados, $train.csv$, $test.csv$, $destination.csv$ e$sample_submission.cs$v. Onde o primeiro e o terceiro deverão ser utilizados para modelar o problema, o segundo deve ser utilizado para se gerar uma submissão para avaliação no Kaggle, e o último é um template de como deve ser feita essa submissão. Na próxima seção discutiremos como foi feito a modelagem do problema. Para isso iremos analisar o conjunto de dados disponível, os algoritmos de aprendizado de máquina experimentados, a métrica de validação e os resultados alcançados. Por fim iremos discutir os resultados, avaliar a que conclusão podemos chegar e apresentar os próximos passos para a continuação do estudo desse problema.

2. Modelagem do Problema

2.1. Dataset

A tabela 1 mostra as $features$ disponiveis, além da resposta que deve ser prevista, no caso o cluster do hotel definido pela coluna hotel_cluster.O conjunto de treino inicial dispõe de 36 milhões de linhas que representam os dados de navegação do usuário dentro do site da Expedia. Essa navegação pode resultar em duas ações, a reserva de um hotel ou o clique para ver mais informações referente ao hotel. Essa ação é definida pela coluna is_booking. Para is_booking = 1 a ação foi uma reserva, para is_booking = 0 a ação foi um clique. Para o conjunto de teste, que será avaliado para geração de uma pontuação, todas as ações são ações de reserva. Logo, para diminuir o conjunto de treino e deixar o problema mais tratável, foi gerado um subconjunto utilizando apenas os dados no qual is_booking = 1, com isso o conjunto de treino passou a ter 3 milhões de linhas.

Outros tratamentos no conjunto de dados foram realizados para facilitar a modelagem do problema. Inicialmente foi criado uma nova coluna que mostra o número de dias para os quais o hotel foi reservado. Além disso os dados que representavam datas foram divididos em ano, mês, dia e trimestre, já que os algoritmos de aprendizado não saberiam lidar com o tipo de dado que representam datas.

Table 1. Descrição dos dados

Nome da Coluna	Descrição
date_time	Timestamp
site_name	ID do site da Expedia onde o evento ocorreu (i.e. Expedia.com, Expedia.co.uk, Expedia.co.jp, ...)
posa_continent	ID do continente associado ao site_name
user_location_country	ID do país onde o cliente está localizado
user_location_region	ID da região onde o cliente está localizado
user_location_city	ID da cidade onde o cliente está localizado
orig_destination_distance	Distância física entre o cliente e o hotel no momento da busca
user_id	ID do usuário
is_mobile	1 se o usuário está conectado utilizando um dispositivo móvel, 0 do contrário
is_package	1 se o evento (clique/reserva) foi feito como parte de um pacote, 0 do contrário
channel	ID do canal de marketing
srch_ci	Data de check-in
srch_co	Data de check-out
srch_adults_cnt	O número de adultos especificados para o quarto do hotel
srch_children_cnt	O número de crianças especificdas para o quarto do hotel
srch_rm_cnt	O número de quartos especificados na pesquisa
srch_destination_id	ID do destino onde o hotel se localiza
srch_destination_type_id	Tipo de destino
hotel_continent	Continente do hotel
hotel_country	Hotel country
hotel_market	Mercado do hotel
is_booking	1 se reserva, 0 se clique
cnt	Número de eventos similares dentro da mesma sessão do mesmo número
hotel_cluster	ID do cluster do hotel

Antes de iniciarmos com a proposta para tratamento do problema, vamos analisar o balanceamento das classes. A figura 1 mostra a frequência com que cada cluster aparece no conjunto de dados. Podemos observar que embora alguns clusters apareçam mais que os outros, existe, em média, um relativo balanceamento entre eles. Discutiremos agora a metodologia adotada nesse trabalho.

2.2. Metodologia

A metodologia desenvolvida nesse trabalho envolve 5 passos, são eles:

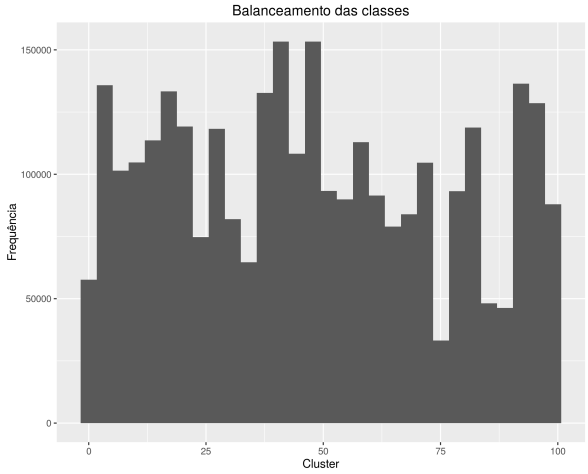

Figure 1. Balanceamento das classes

- Pré processamento dos dados e divisão em treino e validação.
- Escolha do algoritmo de aprendizado de máquina a ser utilizado.
- Definição dos melhores parâmetros para o algoritmo escolhido.
- Geração do modelo a partir do algoritmo escolhido.
- Validação do modelo gerado e avaliação crítica.

O pré processamento dos dados foi discutido na seção anterior. Devido ao grande número de dados disponíveis para treino foi realizado uma amostragem para a utilização na geração do modelo. Assim foram selecionados 2% do conjunto, em torno de 60 mil registros, para treino e outros 2% para validação. Devido ao grande número de dados foi decidido não usar k-fold cross validation para realizar validação, e sim um conjunto de validação.

Sabemos que não existe uma abordagem melhor que a outra quando tratamos de aprendizado de máquina, por isso, inicialmente decidimos ver como o problema se comporta quando modelado utilizando uma regressão logística, uma das mais simples abordagens de aprendizado de máquina, para classificação, que existem. Dessa forma, foram realizados experimentos utilizando regressão logística e três técnicas de regularização: ridge regression, LASSO e elastic net. Foram então variados os valores do parâmetro de penalidade da regularização (lambda) e os resultados foram validados através do conjunto de validação.

$$MAP@5 = \frac{1}{|U|}\sum_{u=1}^{|U|}\sum_{k=1}^{min(5,n)} P(k)$$

Figure 2. Formula da métrica utilizada.

2.3. Experimentos

Para a execução desse trabalho foi utilizado a linguagem de programação R, mais especificamente o pacote h2o. Esse pacote é uma API para o R de uma engine open-source escrita em Java que implementa algoritmos de aprendizado de máquina e permite o processamento paralelo e distribuído para grandes volumes de dados. A escolha desse pacote se sustenta pelo fato de estarmos tratando de um conjunto de dados muito grande, e a forma de processamento utilizada auxilia no melhor tratamento do problema.

Inicialmente foram gerados modelos utilizando a regressão logística para cada uma das técnicas de regularização, vale ressaltar que os dados foram normalizados antes de serem utilizados para a geração do modelo. Além disso, os modelos foram utilizados para prever o cluster em duas situações diferentes, na primeira foram considerados todos os 100 clusters possíveis, na segunda os clusters foram limitados pelo srch_destination_id. Sabemos que o problema de previsão multi-classe para 100 classes diferentes é extremamente sujeito a erros, por isso, definimos uma técnica para diminuir o número de classes possíveis de serem previstas. Para isso utilizamos todo o conjunto de treino e agrupamos os clusters pela coluna srch_destination_id. Essa coluna representa o destino que o usuário está pesquisando, logo, é valido assumir que para um determinado destino apenas certos clusters fazem sentido. Dessa forma, quando realizando uma nova previsão os clusters possíveis passam a ser aqueles que já surgiram como resultado para o srch_destination_id do novo registro. Por fim o melhor modelo gerado foi utilizado para gerar uma submissão no Kaggle. Para esse último caso foi utilizado 10% do conjunto inicial (com is_booking = 1) para o treinamento.

2.4. Métrica

A métrica de validação é a mesma utilizada pelo Kaggle para avaliação dos resultados. Nesse caso é a mean average precision @ 5 (MAP@5). Essa métrica é representada pela fórmula na figura 2.

Nessa fórmula U representa o número de eventos de usuários (número de registros avaliados), n é o número de clusters previstos para um evento e P(k) a precisão no corte k. Suponha U = 2, no qual temos as seguintes previsões e os seguintes valores reais conforme na tabela 2. Para esse

Table 2. Valores de possível saída

Id	Cluster Previsto	Cluster Real
1	1 5 9 10 11	5
2	2 3 15 7 99	99

caso teríamos $MAP@5 = (4/5 + 1/5)/2 = 0.5$.

2.5. Resultados

Como explicado anteriormente foram gerados vários modelos, utilizando diferentes técnicas de regularização, variando o valor da penalidade (lambda), além disso as previsões foram feitas em duas situações diferentes. Os três gráficos abaixo mostram o resultado para os dados de treino e validação, para cada técnica de regularização quando se varia o valor da penalidade (lambda), e para aos dois contextos definidos.

O gráfico da figura 3 mostra os resultados quando utilizamos $ridgeregression$, o gráfico da figura 4 quando utilizamos o $LASSO$ e o gráfico da figura 5 quando utilizamos a $elasticnet$. No eixo y temos a precisão medida utilizando $MAP@5$ e no eixo x a variação do valor da penalidade (lambda). No título dos gráficos temos a pontuação que o melhor modelo, para aquela técnica de regularização, gerou quando avaliado no conjunto de validação. Para o caso do $elasticnet$, que gerou os melhores resultados em validação, observamos também o resultado da avaliação no Kaggle. Podemos observar resultados parecidos para as três técnicas de regularização, o valor ótimo de lambda, dentre os testados, foi 0.0005, e temos uma performance ligeiramente melhor quando utilizando $elasticnet$.

3. Discussão

O valor de referência para MAP@5 em uma escolha aleatória de clusters é em torno de 2.3%. Podemos observar pelos resultados que quando a escolha é feita para os 100 clusters possíveis o resultado é semelhante a escolha aleatória, o que não representa nenhum ganho do modelo gerado. No entanto, quando a escolha é feita limitando os clusters pelo destino, o resultado sobe para uma precisão em torno de 9%.

Ainda que a precisão aumente o resultado é relativamente ruim. Outra característica que percebemos, analisando os resultados, é que os valores da métrica utilizada para o conjunto de treino, validação e o resultado da avaliação no Kaggle é aproximadamente o mesmo, isso indica um alto $bias$. Além disso o treinamento utilizado para gerar a submissão para o Kaggle utilizou mais dados e o resultado se manteve, o que também é indicativo de alto $bias$

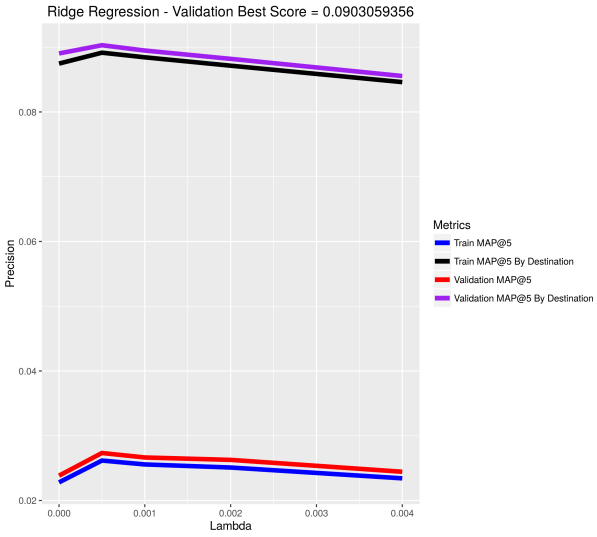

Figure 3. Resultados para ridge regression

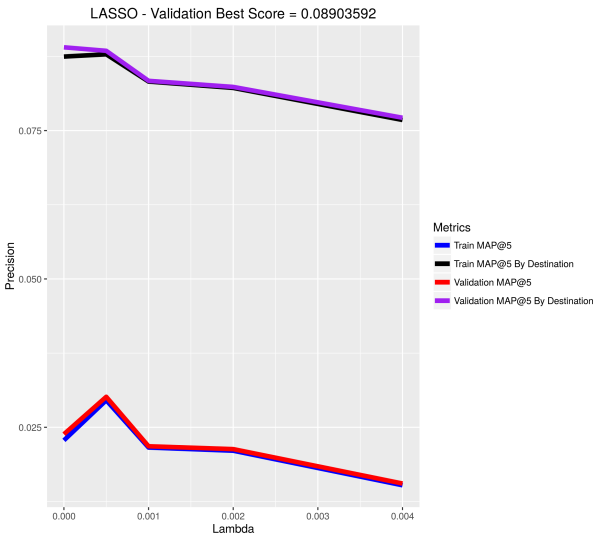

Figure 4. Resultados para lasso

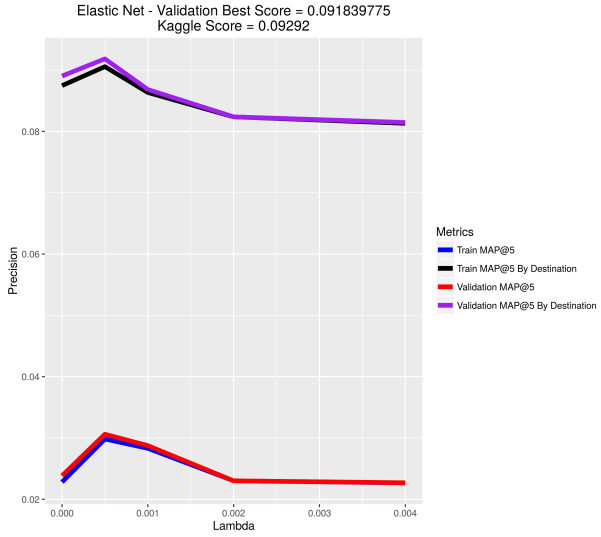

Figure 5. Resultados para elastic net.

4. Conclusão

Observamos que, embora a precisão aumente quando se limita a escolha pelo destino, os resultados gerados ainda possuem uma baixa precisão. Isso nos leva a crer que a resposta não é linear em relação as $features$ utilizadas. Podemos confirmar essa suspeita observando os valores dos coeficientes de correlação de Pearson entre as $features$ e a resposta. Esse coeficiente mostra se duas variáveis são linearmente dependentes entre si, valores próximos de 0 indicam que não existe relação linear, no entanto pode existir uma relação que não seja linear. O gráfico da figura 6 mostra o valor do coeficiente de correlação de Pearson no eixo y e um índice que representa cada $feature$ no eixo x. Podemos observar que todas apresentam valores próximos a 0.

Concluímos anteriormente que a relação entre as features e a resposta não é linear, portanto os próximos passos envolvem utilizar métodos de aprendizado de máquina que são sensíveis a não linearidade. Além disso não utilizamos o arquivo destinations.csv, esse conjunto de dados representa características latentes aos destinos que foram normalizadas e tornadas anonimas. No entanto, esse arquivo possui uma alta dimensionalidade, portanto técnicas de redução de dimensionalidade podem ser necessárias para que sejam adicionados ao modelo.

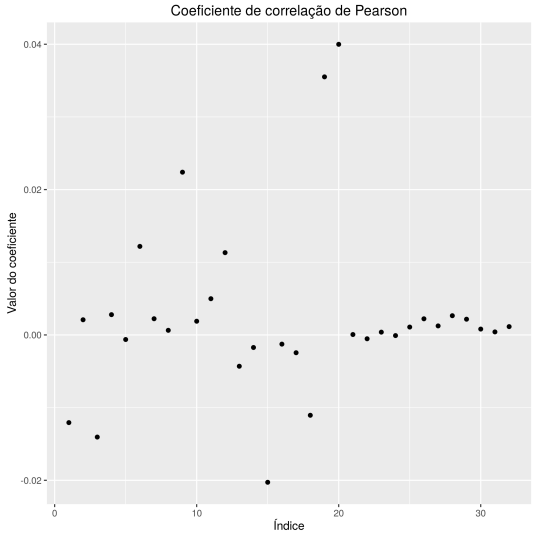

Figure 6. Coeficiente de correlação de Pearson

Referências

Gareth James, Daniela Witten, Trevor Hastie and Tibshirani, Robert. An introduction to statistical learning. Srpinger, 2009.

Spencer Aiello, Tom Kraljevic and Maj, Petr. h2o package, 2016. URL https://cran.r-project.org/web/packages/h2o/h2o.pdf.