INF-0619

Projeto Final

Instituto de Computação - UNICAMP

Trabalho Final: NYC Taxi

Prof. Zanoni Dias

Data Troopers (grupo 11):

- Eduardo Nogueira Pavan
- Gustavo Leite Machado
- Guilherme Ramos Gouveia
- Caio Augusto Cunha Volpato

Campinas, 23 de agosto de 2020

Introdução

O presente trabalho discorre sobre uma possível solução para o desafio do kaggle New York City Taxi Trip Duration, que tem como objetivo prever o tempo da duração de uma viagem de táxi a partir de dados históricos de viagens de 2016 com informações variadas e com a possibilidade de incluir dados externos na análise.

Conjunto de dados

A competição disponibilizou um conjunto de dados de viagens de taxi (e limusines) realizadas entre Janeiro de 2016 até Julho de 2016.

O conjunto disponibilizado já estava separado entre treino (1.4 milhão de registros) e teste (0.6 milhão de registros) e tem os seguintes campos e suas respectivas descrições:

1. id: identificador da viagem
2. vendor_id: identificador do fornecedor da viagem
3. pickup_datetime: data e hora do início da viagem
4. dropoff_datetime: data e hora do fim da viagem
5. passenger_count: número de passageiros
6. pickup_longitude: local do início da viagem (longitude)
7. pickup_latitude: local do início da viagem (latitude)
8. dropoff_longitude: local do fim da viagem (longitude)
9. dropoff_latitude: local do fim da viagem (latitude)
10. store_and_fwd_flag: se o taxi tinha conexao quando enviou as infos, ou foi armazenada na memória e enviada posteriormente.
11. trip_duration: duração da viagem (s) **(target do modelo)**

Feature Engineering

A Fim de melhorar o desempenho do modelo preditivo, vamos criar features adicionais, são elas:

1. Tempo:
    a. Dia da Semana (início da viagem)
    b. Hora do dia (início da viagem)
    c. Feriados (início da viagem)
2. Distância e rota:
    a. Distância Linear
    b. Distância Manhattan
    c. Distância por avenidas
    d. Identificação das principais Rotas
    e. Direção da Rota
    f. Origem e destino dentro ou fora de manhattan
3. Clima
    a. Precipitação
    b. Neve
    c. Profundidade de neve
4. Trânsito
    a. Média de velocidade

Algumas dessas features necessitam de uma explicação mais aprofundada:

2.c) Distância por avenidas:

1) Criação do shapefile com avenidas de manhattan:

Utilizamos dois shapefiles, o primeiro com todas as vias de NY extraído do seguinte link: https://data.cityofnewyork.us/City-Government/NYC-Street-Centerline-CSCL-/exjm-f27b

O segundo com a divisão por áreas (Boroughs) de NY: https://data.cityofnewyork.us/City-Government/Borough-Boundaries/tqmj-j8zm

Processamos os dados no Alteryx (workflow a_01_Criar_shapefile_Avenues.yxmd) para selecionar e agrupar apenas as principais avenidas de manhattan

2) Cálculo da distância por avenidas:

Desenvolvemos um racional dentro do Alteryx para encontrar a avenida mais próxima do ponto de partida e calcular uma distância aproximada para os três seguimentos de reta, as distâncias para a avenida e a distância percorrida na avenida. Testamos o calculo com a avenida mais próxima do ponto de partida e com a mais próxima do ponto de destino.

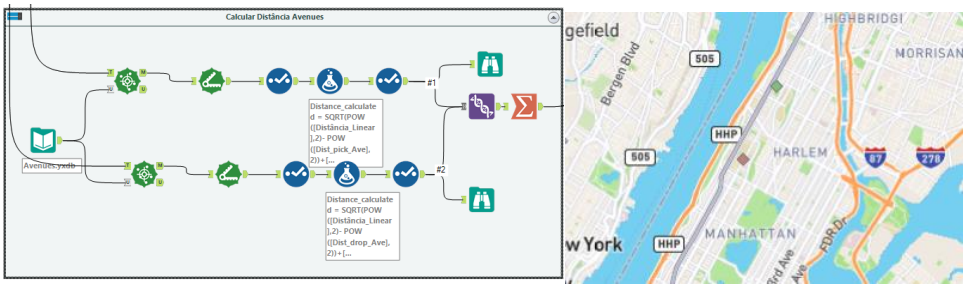

Figura 1 – Cálculo distância Avenidas

Para checar a acurácia da distância calculada por avenidas utilizamos a API do google maps para calcular a distância de menor rota para uma amostra de 1900 rotas e segmentamos as rotas pelas origem e destino dentro ou fora de manhattan. O resultado do MAPE de cada medida de distância é apresentada na tabela 1.

Origem	Destino	Viagens treino	%	Viagens Teste	%	MAPE Linear	MAPE Manhattan	MAPE Avenidas
Manhattan	Manhattan	1232293	84%	528114	84%	24,3%	16,6%	10,4%
Manhattan	Not Manhattan	111499	8%	48055	8%	31,9%	18,5%	17,7%
Not Manhattan	Manhattan	56648	4%	24150	4%	34,0%	17,9%	16,3%
Not Manhattan	Not Manhattan	58204	4%	24815	4%	27,7%	17,0%	27,5%

Tabela 1 – Erro distâncias

Dessa forma usamos a distância para avenidas para as viagens em manhattan e a distância de manhattan para os outros tipos.

2.d) Identificação das principais Rotas:

Construímos uma malha de 1km de granularidade e construímos as rotas conectando o ponto de partida e de destino. Ordenamos as rotas por duração da viagem e denominamos um número para cada rota. Primeiramente tentamos filtrar somente as principais mas ao final entendemos que fazia sentido nomear todas as rotas.

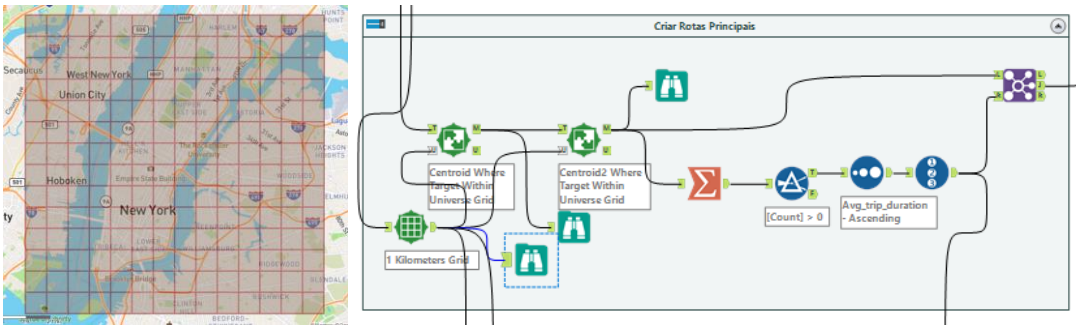

Figura 2 – Definição das principais rotas.

2.e) Direção da rota:

Calculamos a direção em graus das rotas e agrupamos por intervalos de 5 graus.

2.f) Origem e destino dentro ou fora de manhattan

Usamos o shapefile de boroughs para se os pontos de partida e destino se davam dentro ou fora de manhattan. Como apresentado na tabela 1, 84% das viagens começam e terminam na ilha de manhattan. Definimos um número para cada combinação de origem e destino.

3) Clima:

Buscamos no site da NOOA (National Centers for Environmental Information):
https://www.ncdc.noaa.gov/cdo-web/search

Conseguimos dados diários para o período das viagens e isolamos os dados da estação do central park. As três informações que utilizamos foram a precipitação diária, a queda de neve e a profundidade de neve no solo. Como há uma dispersão muito grande nos valores principalmente de precipitação e como acreditamos que o efeito da chuva em uma cidade grande seria parecido a partir de um certo valor de volume decidimos por transformar as variáveis em binárias, teve ou não precipitação no dia.

4)Trânsito:

Conseguimos os dados fornecidos pelo betaNYC (entitade cívica de NY):
https://data.beta.nyc/dataset/nyc-real-time-traffic-speed-data-feed-archived

Isolamos os dados relativos à sensores de manhattan e calculamos a média por hora/dia.

Análises exploratórias

A fim de entender melhor o desafio vamos analisar os dados fornecidos e construídos:

Pelo histograma da figura 3 podemos concluir que as viagens são em sua maioria mais curtas de 3 à 20 minutos, mas uma amostra relevante de viagens mais longas.

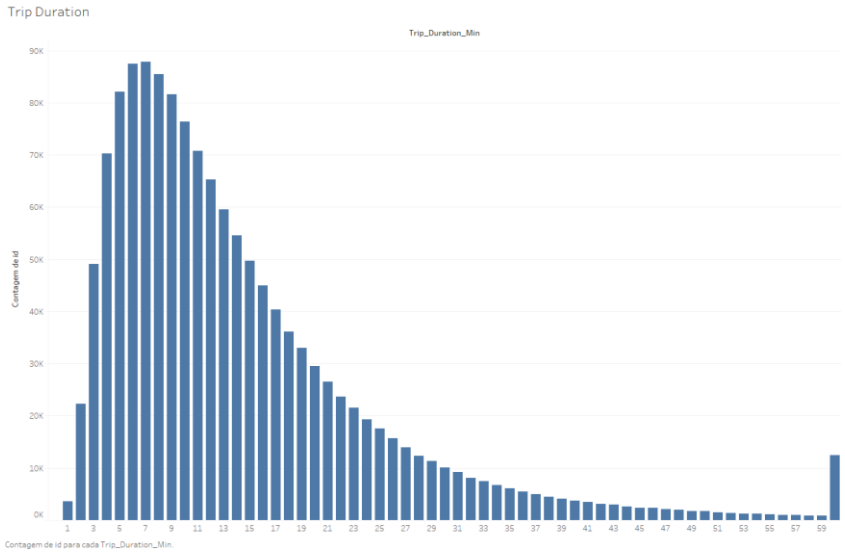

Figura 3: Histograma do tempo de viagem.

E pelos gráficos da figura 4 podemos ver as diferenças na duração das viagens por dia da semana e hora do início da viagem.

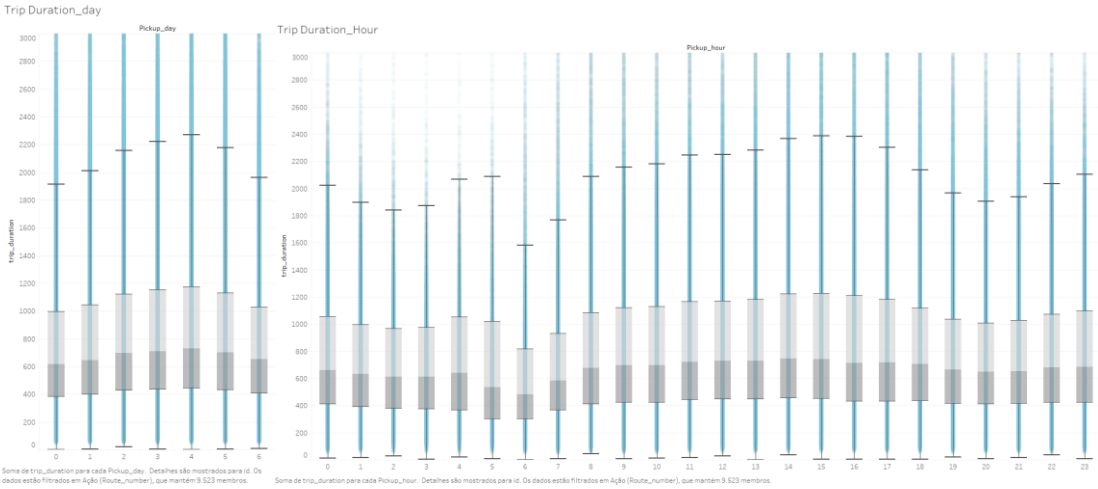

Figura 4 – Boxplots da duração das viagens por dia da semana e hora do dia

Sobre a distribuição geográfica dos pontos de partida plotamos um mapa com uma malha e o número de partidas (esquerda) e chegadas (direita) na figura 5.

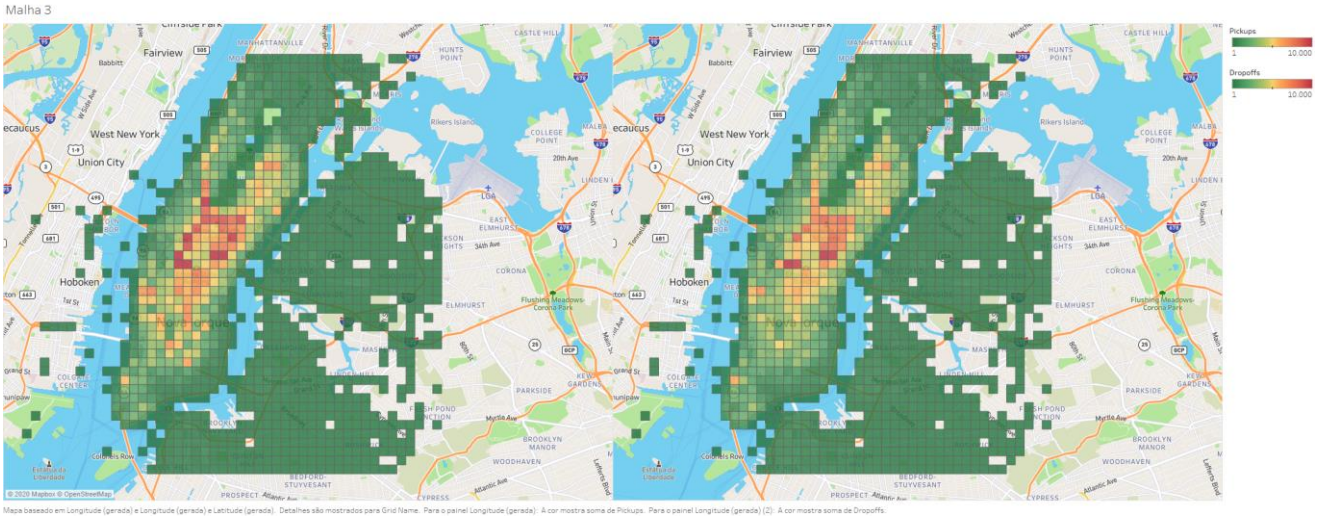

Figura 5 – Concentração das partidas e destinos por célula de uma grade de 300m centrada em manhattan.

Podemos perceber que há uma grande concentração em manhattan, tanto de partidas como de destinos, como já também concluimos na tabela 1.

Utilizando uma malha um pouco menos granular (1km) definimos as rotas com a combinação das células de partida e destino. Em seguida construímos um dashboard para entender o comportamento das principais features por rota, o que ao nosso ver faria mais sentido do que tentar explorar os dados de forma genérica. Na figura 6 podemos ver o dashboard sem nenhum filtro, é possível identificar alguns padrões, mas tudo fica mais claro quando isolamos algumas rotas (figura 7) e exploramos os efeitos nas variáveis para caminhos similares.

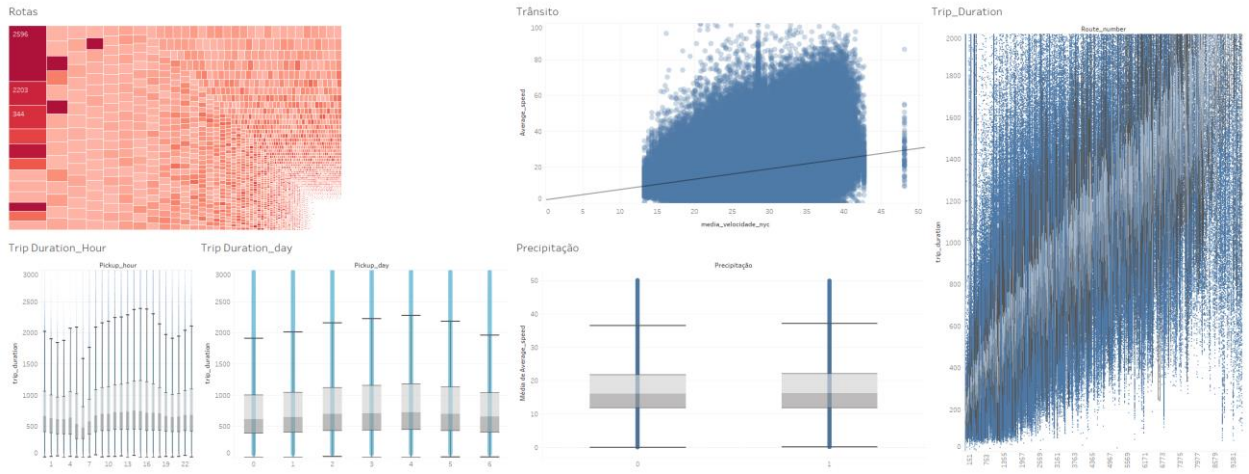

Figura 6: Dashboard das variáveis sem filtro de rotas

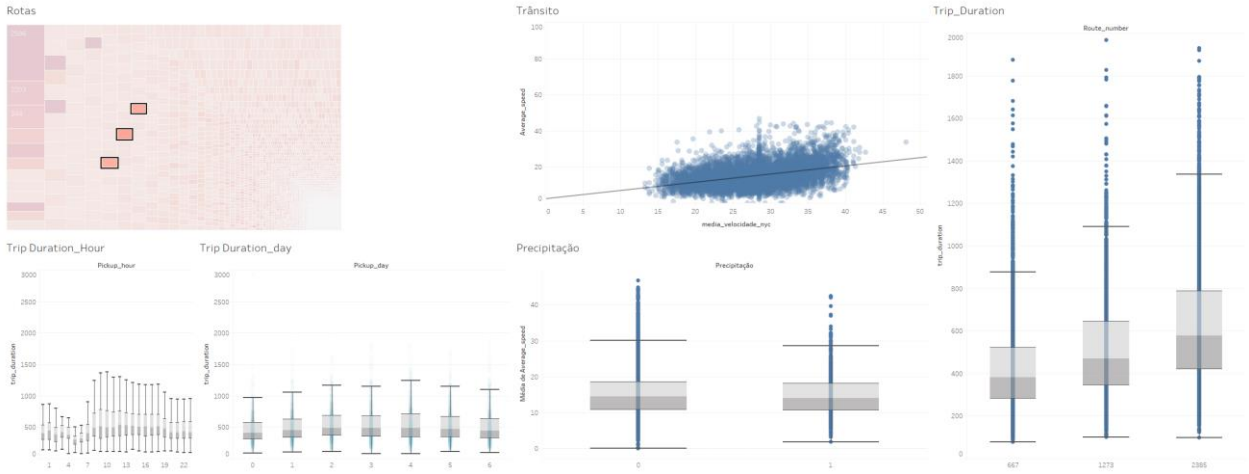

Figura 7: Dashboard das variáveis com algumas rotas selecionadas.

Da análise do dashboard podemos já realizar uma seleção prévia das variáveis e definir alguns limites para realizar os filtros dos dados de treino. Diferentemente do que esperávamos a variável de precipitação não mostra uma diferenciação na velocidade de deslocamento médio das rotas.

Tratamento dos dados

A partir da exploração dos dados realizamos três filtros de outliers na base:

1. Número de passageiros como 0
2. Velocidades médias de viagem menores que 5km/h e maiores que 120 km/h
3. Distâncias iguais a 0 ou maiores que 25km.

Em seguida decidimos testar uma clusterização das viagens para facilitar o trabalho dos modelos. Para a clusterização selecionamos as features, primeiramente removendo variáveis muito correlacionadas e depois utilizando a função de selectKbest para escolher as 5 principais features. Definimos o melhor número de clusters utilizando a função de silhueta (figura 8) e analisamos os cluster gerados através de um PCA na figura 9.

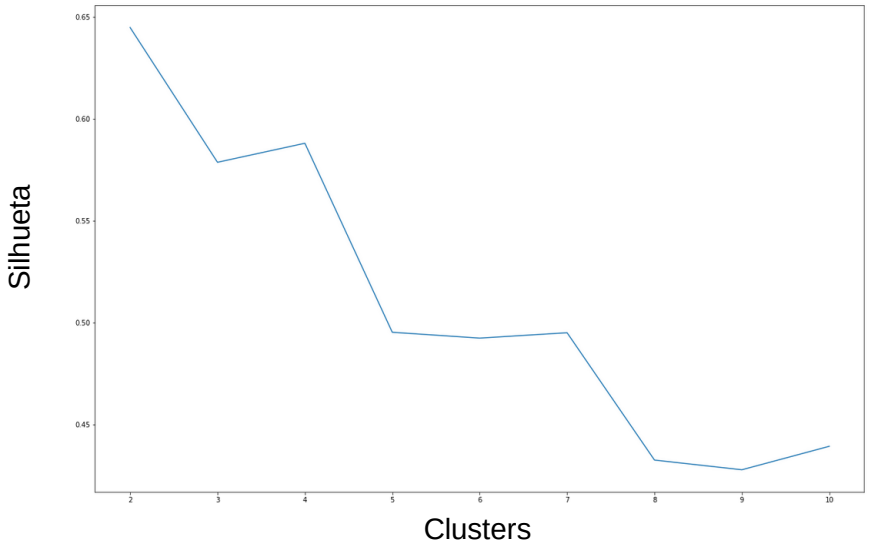

Figura 8 – Variação do indicador de silhueta por número de clusters

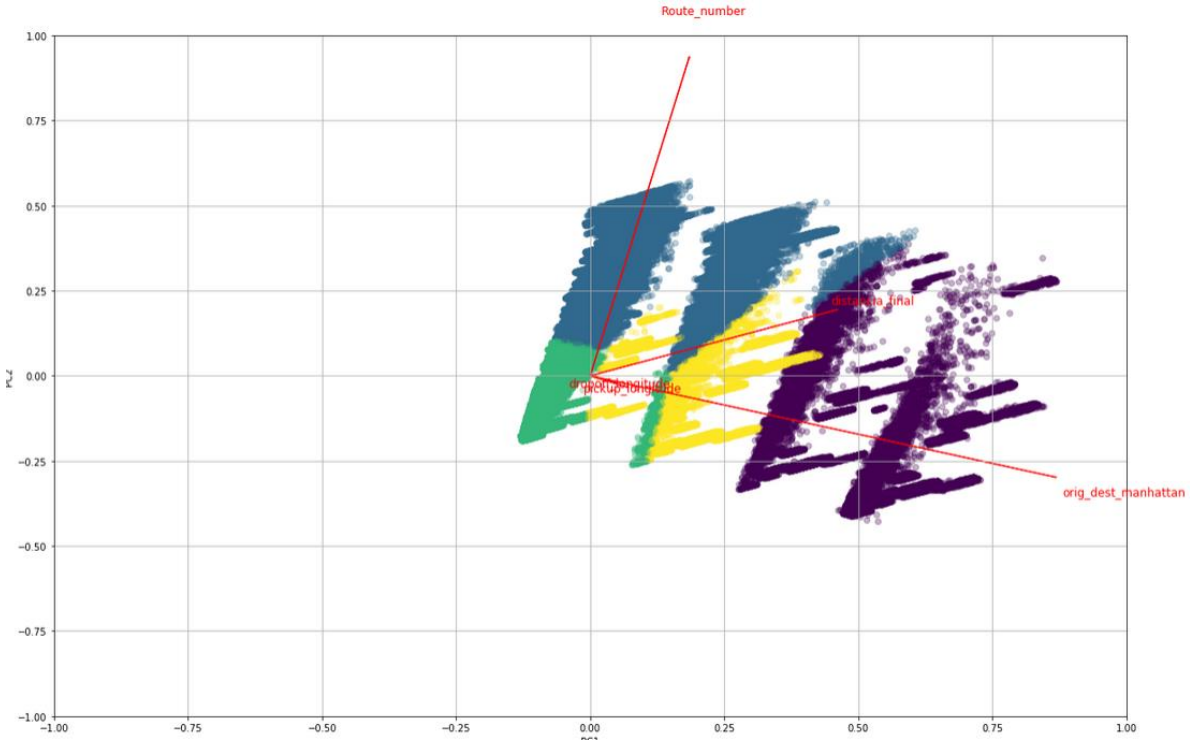

Figura 9 – PCA dos 4 clusters gerados com os vetores das variáveis

Apesar dos grupos não estarem claramente separados entendemos que os clusters poderiam ajudar o modelo dado que as viagens do mesmo grupo possuem características parecidas.

Para gerar a base para alimentar os modelos também realizamos uma seleção de features utilizando a mesma função de selectKbest para selecionar as 15 melhores features.

Modelos Preditivos

Baseline:

No modelo baseline treinamos um modelo de random forest com hiperparâmetros padrões e com poucas features adicionais (somente distância manhattan) e obtivemos o resultado de RMSLE de 0,56 no teste no Kaggle.

Modelos finais:

Após processar todas as features no alteryx e no python construímos um modelo de random forest para cada cluster de viagem definido no passo anterior. Testamos várias combinações de hiperparâmetros até chegarmos no resultado final do modelo que obteve o resultado apresentado na figura 10 para treino e validação e o resultado de teste no Kaggle apresentado na figura 11.

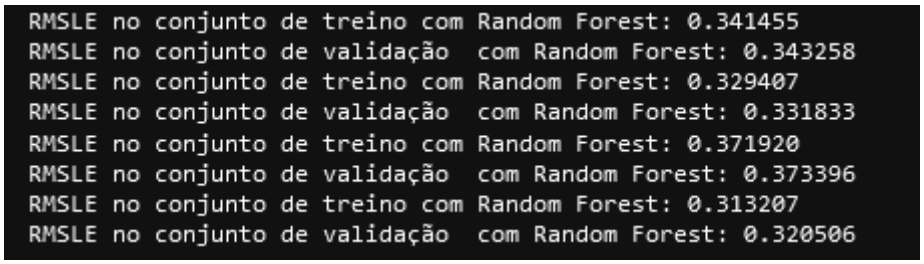

Figura 10 – Resultados de treino e validação para o modelo Random Forest

Figura 11 – Resultado do Teste no Kaggle do modelo Random Forrest.

Testamos também um modelo XGboost mas mesmo variando os hiperparâmetros não conseguimos controlar o overfitting do modelo como apresentado na figura 12, mesmo assim o resultado não foi muito pior que o Random forest no treino como apresentado na figura 13.

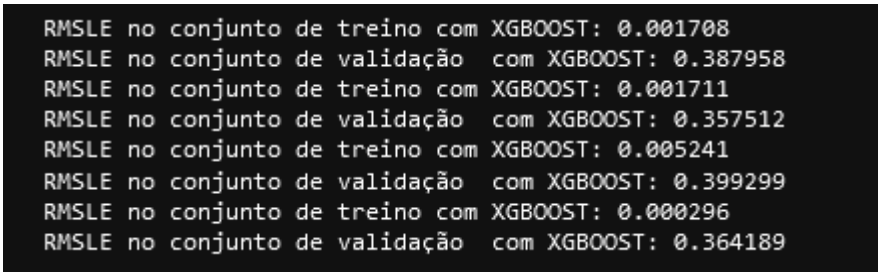

Figura 12 – Resultado no treino e validação do modelo XGboost

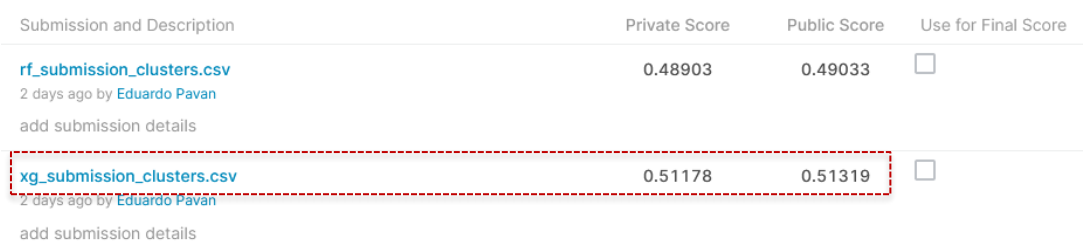

Figura 13 – Resultado no teste do Kaggle para o modelo XGboost

Conclusões

Obtivemos uma melhora considerável em relação ao baseline e as novas features se mostraram em sua maioria relevantes para o treinamento do modelo. Apesar do nosso resultado estar longe do recorde de 0.28 estaríamos próximos do meio da tabela (posição 770 de 1254) o que mostra que as features desenvolvidas foram suficientes para que um modelo simples pudesse ter uma acuracidade razoável. Acreditamos que a aplicação de modelos mais complexos e um treinamento mais extenso de um modelo como o XGboost ou uma rede neural poderia trazer resultados ainda mais acurados com as features já desenvolvidas. Por outro lado outras features externas que não pensamos poderiam ser usadas para também aumentar a acuracidade do modelo. Mesmo assim acreditamos que a solução desenvolvida nesse trabalho traz além de um pacote de features criativas um resultado satisfatório de modelo de predição para o desfio proposto.