FUNDAÇÃO GETULIO VARGAS

RAFAEL NONATO

ANALISE PREDITIVA
PROJETO EM INDIVIDUAL

PROFESSOR: JOÃO RAFAEL DIAS PINTO

SÃO PAULO
2020

Contexto

Após ter estudado por 24h a disciplina de Análise preditiva do curso de *Business Analytics* da faculdade Getúlio Vargas, com um trabalho sob a característica de avaliação individual com a orientação do professor João Rafael Dias Pinto.

A atividade em questão é a exploração de uma base de dados da empresa UBER, onde será realizado alguns estudos para identificação de um modelo preditivo que identifica o volume de demandas da cidade de Nova York nos Estados Unidos. Com base nisso foi utilizado o software *RStudio* com codificação em R para construir os modelos e realizar as experimentações da base de dados disponibilizada no *Kaggle*.

Preparação dos Dados

Base de dados utilizada

Para o estudo em questão foi disponibilizada a base de dados denominada `uber_nyc_enriched.csv`. Ela contém o histórico de viagens realizadas na cidade de nova York e também os bairros, temperatura, dias, mês e ano assim como o número de viagens.

Descrição das variáveis

A base `uber_nyc_enriched.csv` é composta por um único *dataset* representada abaixo após o carregamento no RStudio. Abaixo é possível encontrar a lista de variáveis e suas respectivas tipagem.

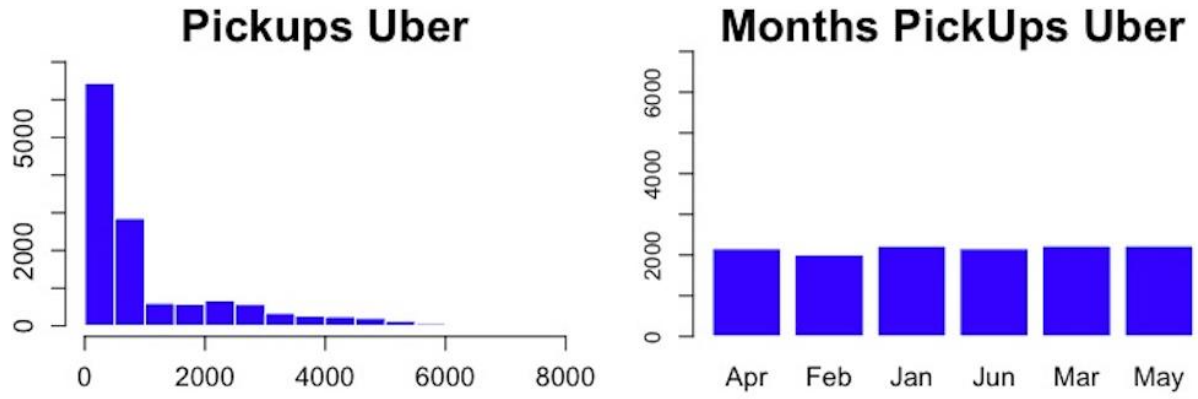
```
> str(DATA)
'data.frame':   29101 obs. of  13 variables:
 $ pickup_dt: Factor w/ 4343 levels "2015-01-01 01:00:00",..: 1 1 1 1 1 1 1 2 2 2 ...
 $ borough  : Factor w/ 6 levels "Bronx","Brooklyn",..: 1 2 3 4 5 6 NA 1 2 3 ...
 $ pickups  : int  152 1519 0 5258 405 6 4 120 1229 0 ...
 $ spd      : num  5 5 5 5 5 5 5 3 3 3 ...
 $ vsb      : num  10 10 10 10 10 10 10 10 10 10 ...
 $ temp     : num  30 30 30 30 30 30 30 30 30 30 ...
 $ dewp     : num  7 7 7 7 7 7 7 6 6 6 ...
 $ slp      : num  1024 1024 1024 1024 1024 ...
 $ pcp01    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ pcp06    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ pcp24    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ sd       : num  0 0 0 0 0 0 0 0 0 0 ...
 $ hday     : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...
```

Análises exploratória dos dados

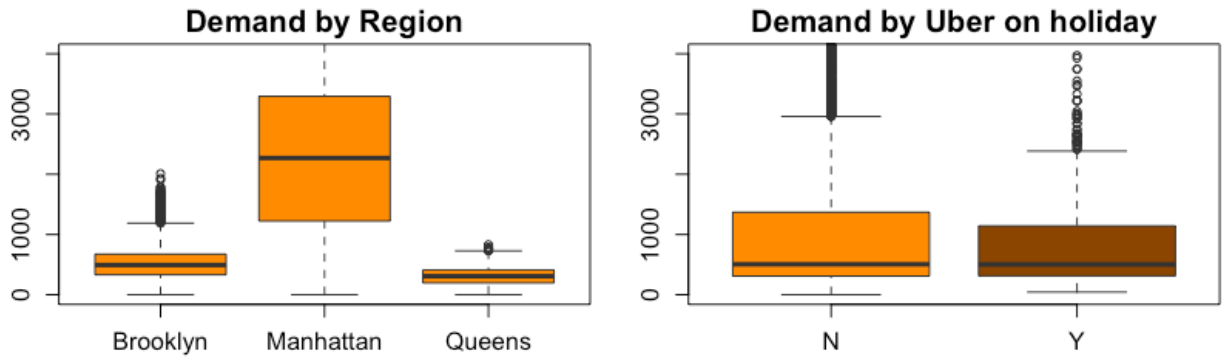
Uma análise exploratória dos dados para verificar o número de viagens por bairro em NY. Com essa dimensão podemos saber se existe algum NA no *dataset*. Outro ponto importante é que usaremos apenas 3 bairros para está análise, Manhattan, Brooklyn, Queens, representado no dado abaixo.

	borough	`Pickups Totally`
	<fct>	<int>
1	Manhattan	10367841
2	Brooklyn	2321035
3	Queens	1343528
4	Bronx	220047
5	Staten Island	6957
6	NA	6260
7	EWR	105

Após realizar o tratamento da base de dados, necessário para identificar melhor as informações para esta análise dentro dos cenários, verifiquei a distribuição das viagens mensais e por bairro. Aqui também foram realizadas as conversões de *datetime,* temperatura, velocidade e altura para que elas se comportassem como variáveis qualitativas. Essa distribuição pode ser observada no histograma abaixo.



E possível observar a distribuição entre os meses do número de viagens sendo assim uma frequência alta, e se mantendo na média entre todos os meses, mas de acordo com a figura onde mostra o total de viagens, pick-ups UBER há uma concentração entre a faixa de meses ao lado.

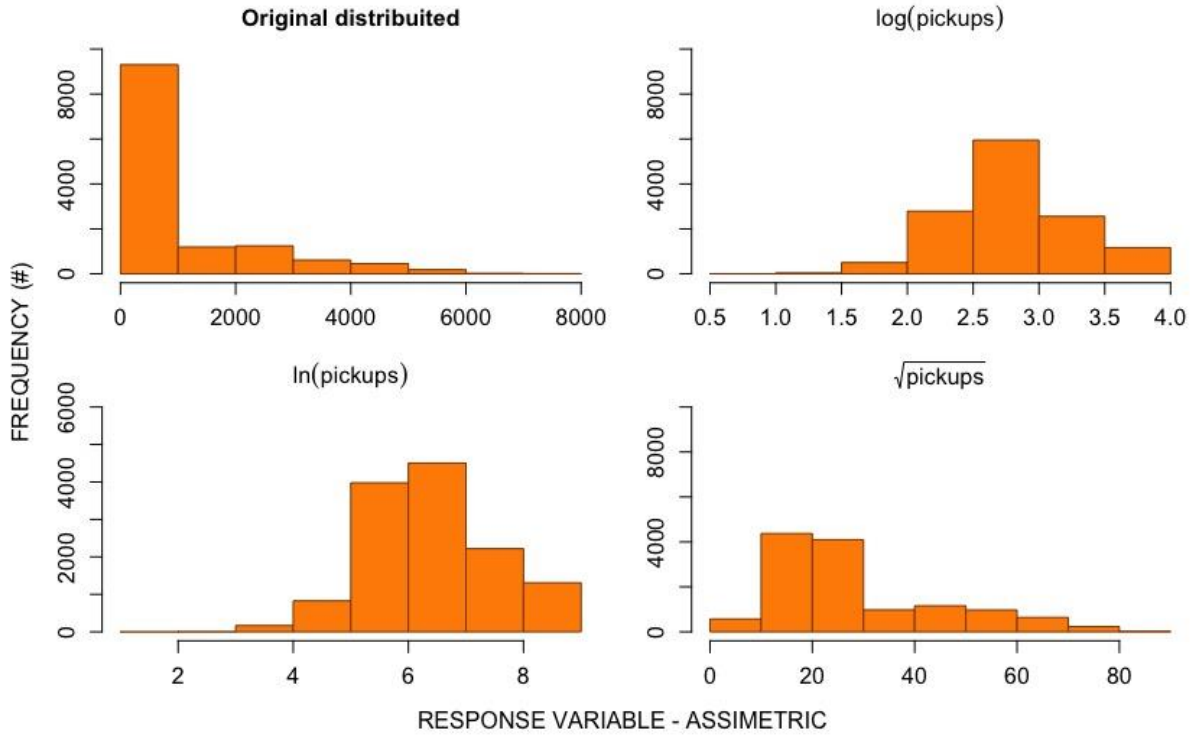


No boxplot acima, as seguintes estatísticas descritivas podem ser observadas:

- A distribuição dos dados demonstra que Manhattan está muito acima das demais regiões em número de viagens.
- No feriado as demandas aparecem na média inclusive no entre os primeiros quatis.

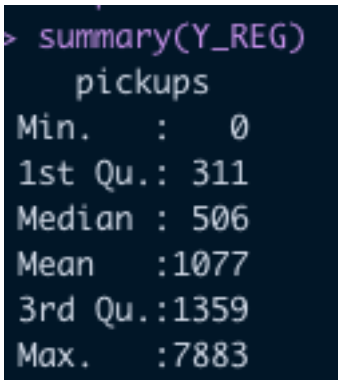
Análise da variável autoexplicativa

Abaixo a análise da variável autoexplicativa *pickups,* onde o número de viagens e suas distribuições aparecem numa frequência alta conforme o histograma abaixo.

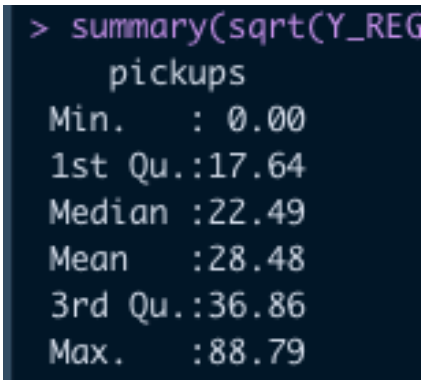


Análise das variáveis com raiz quadrada

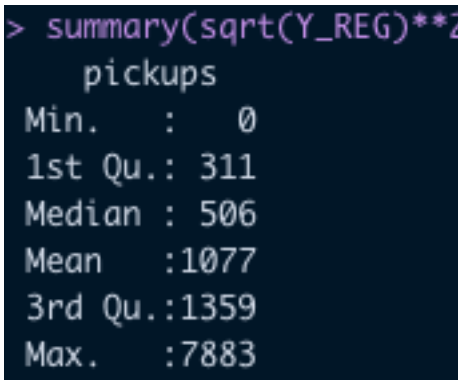
Abaixo as análises de Raiz quadrada para confirmar os valores entre a variáveis transformada e a original.



Quantidade original



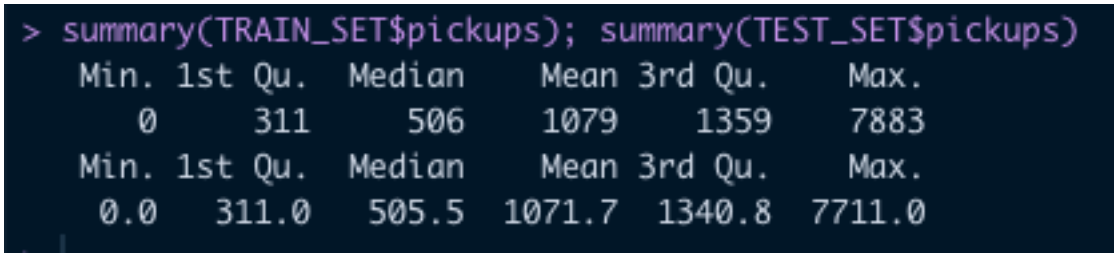
Quantidade transformada



Retornando para original

Análise de distribuição na variável resposta

Após a separação da base de treino e teste onde foi considerado 70% da base de treino e 30% para a base de teste, as amostras abaixo demonstram a distribuição igualitária da variável resposta *pickups* entre os modelos preparados.



Com isso a variável pick-up em minha análise passa a ser considerada, até aqui como uma das mais importantes até o momento pela sua representatividade na distribuição e na validação dos histogramas acima, ainda apenas como percepção. Mais a frente haverá a comprovação desta percepção.

Construção e análise das árvores

Aqui, neste momento foi um grande desafio cria-la por desconhecimento na preparação, porém os resultados abaixo demonstram a criação da árvore e seus números n, abaixo, porém existe um CP=0.0015 para uma poda inicial. Antes de fazer esse tratamento iniciei com CP= -0.1 e ficou extremamente grande, com esse valor consegui imprimir estes resultados e um número n= 9123.

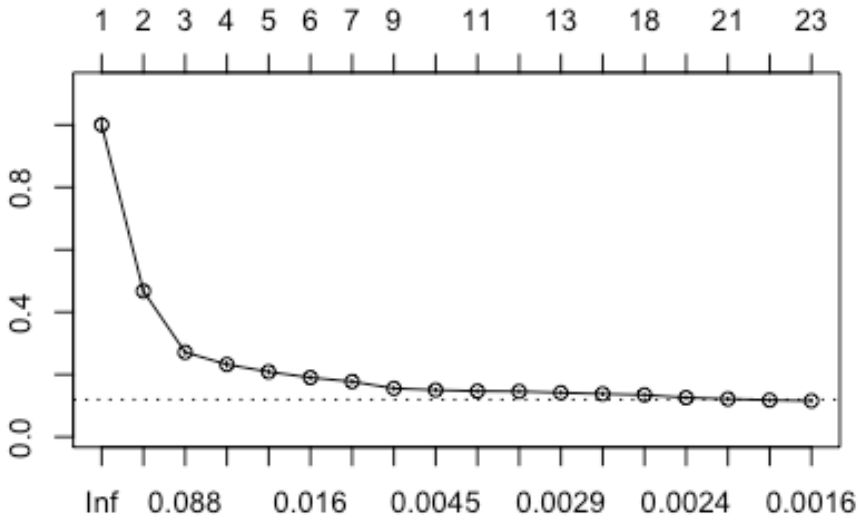
```
Call:
rpart(formula = pickups ~ ., data = TRAIN_SET, method = "anova",
    control = rpart.control(minbucket = 10, cp = 0.0015))
  n= 9123

           CP nsplit rel error    xerror        xstd
1  0.532764969      0 1.0000000 1.0001694 0.022994680
2  0.196124859      1 0.4672350 0.4673812 0.012124583
3  0.039293716      2 0.2711102 0.2713575 0.007342277
4  0.024112922      3 0.2318165 0.2322046 0.008294123
5  0.018523331      4 0.2077035 0.2086406 0.006526565
6  0.012980840      5 0.1891802 0.1901567 0.006104882
7  0.010701362      6 0.1761994 0.1772516 0.005778789
8  0.005326824      8 0.1547966 0.1554752 0.004872060
9  0.003754201      9 0.1494698 0.1501554 0.004861436
10 0.003452369     10 0.1457156 0.1468261 0.004863751
11 0.003049537     11 0.1422632 0.1457950 0.004743557
12 0.002770284     12 0.1392137 0.1416161 0.004583000
13 0.002547141     13 0.1364434 0.1379536 0.004473560
14 0.002542725     17 0.1261401 0.1346545 0.004458397
15 0.002322128     19 0.1210546 0.1258768 0.004254364
16 0.001995018     20 0.1187325 0.1215714 0.004170619
17 0.001800332     21 0.1167375 0.1179027 0.004095473
18 0.001500000     22 0.1149372 0.1155746 0.004094149

Variable importance
borough_Manhattan              hora       weekday_Sun              temp    borough_Queens       weekday_Sat       weekday_Mon
               57                30                 3                 2                 2                 2                 1
```

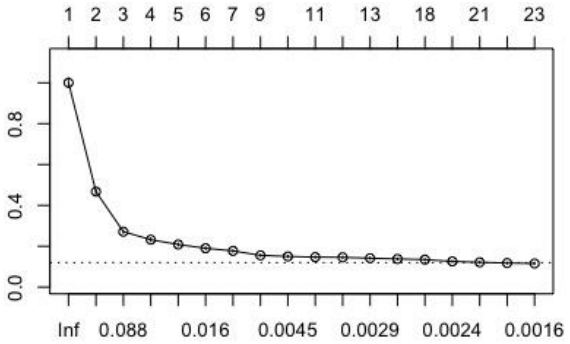
Análise e poda das árvores

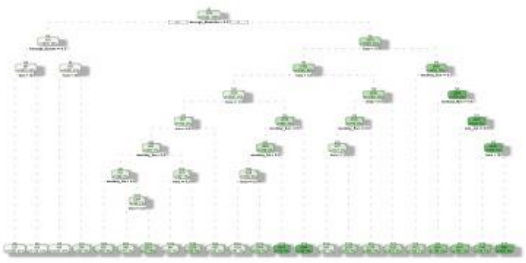
Abaixo o resultado após aplicar a poda, pois o número a quantidade de árvores passou a não fazer diferença mais após esse valor de 0.0015 que considerei como ponto de corte. Isso está demonstrado no quadro abaixo.

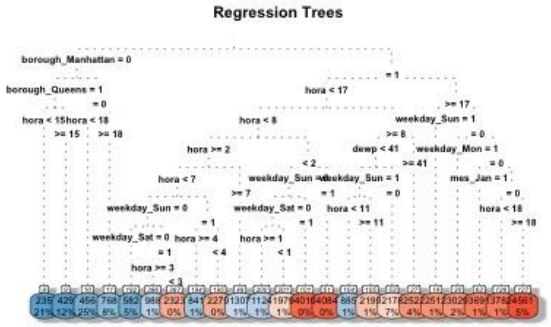


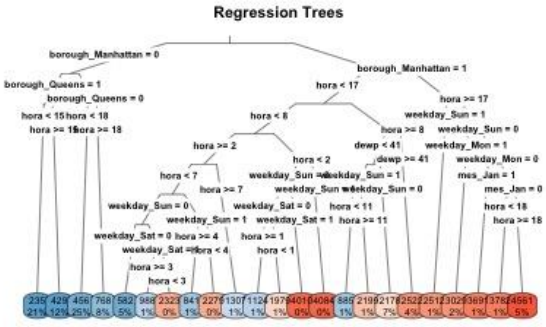
Visualizando as arvores de regressão

Esse foi um momento bem complicado da construção pois a máquina não conseguia *renderizar* todas as arvores e também eram muitas antes da poda, até que encontrei um número adequado para demonstrar tudo em um plano só. Abaixo a representação das arvores e suas quebras.









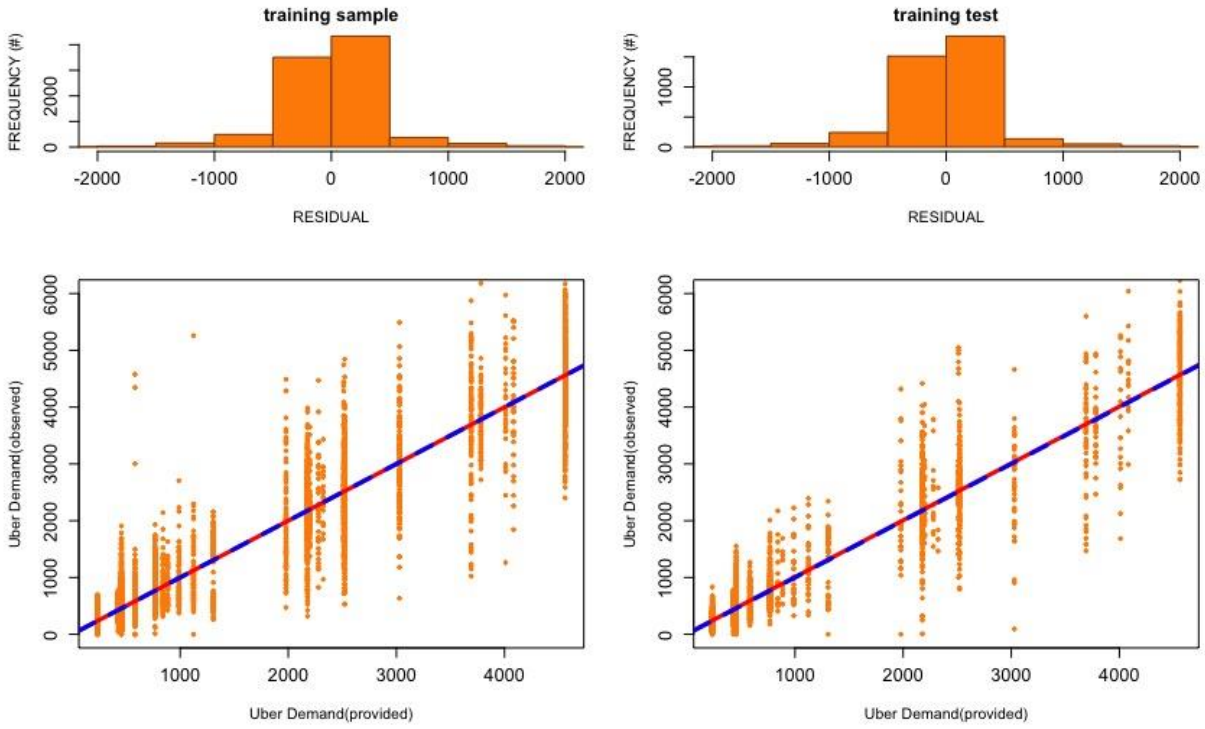
Avaliação de performance do modelo, *Overfitting*

Na comparação entre as bases de treino e teste conforme a figura abaixo representa que o valor de *Rsquared* são muito próximos significando que há uma baixa variação entre os dois modelos de treino e teste garantindo a sua reprodutibilidade.

```
> # Arvore
> postResample(pred = Y_VAL_TRAIN, obs = TRAIN_SET$pickups)
       RMSE    Rsquared         MAE
428.4084094   0.8850628 265.9517338
> postResample(pred = Y_VAL_TEST,  obs = TEST_SET$pickups)
       RMSE    Rsquared         MAE
426.1066535   0.8846188 263.3040971
```

Análise e observação das árvores entre treino e teste

Observando os dois modelos e comprovando o que a figura acima afirma é que não há muita diferença nas distribuições entre os modelos de treino e teste, de acordo com o histograma abaixo. Acredito que a comparação desta amostra foi importante para esta demonstração nas distribuições abaixo.



Análise na regressão com *Random Forest*

Outro momento desafiador foi este, onde o algoritmo precisa ser aplicado o *Randow Forest* explicado e sala e também aplicado neste modelo, onde as bases de treino e teste também foram selecionadas conforme o valor de 30%, 70% por cento. Depois o algoritmo também foi bem complicado em rodar, pesado para compilar, porém obtive os demais resultados abaixo.

Distribuição da variável resposta entre treino e teste

Aqui como foi feito anteriormente apenas para garantir que a distribuição entre treino e teste estão igualitárias, isso pode ser observado na variável *Median* e na *1st Qu*. Isso significa que temos uma divisão consistente para seguirmos.

```
> summary(TRAIN_SET$pickups);
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0     311     506    1079    1359    7883
> summary(TEST_SET$pickups)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0   311.0   505.5  1071.7  1340.8  7711.0
```

Aplicando ao modelo de treino o *Random Forest*

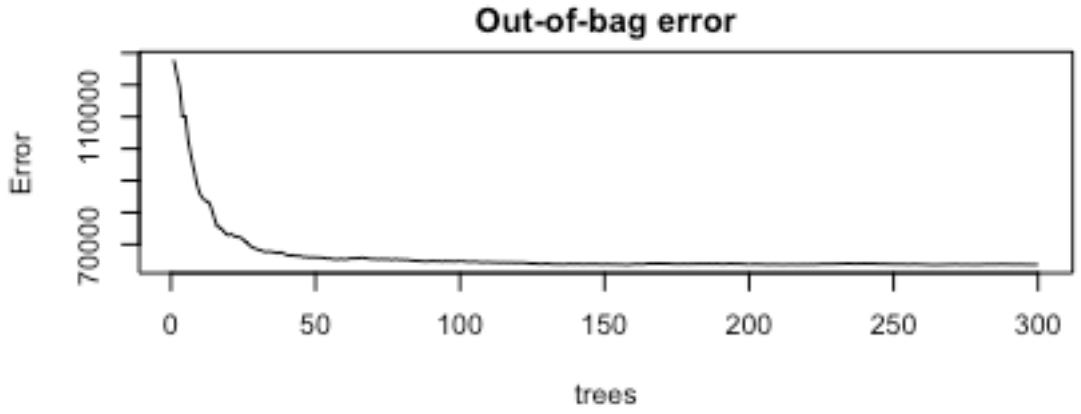
Nesta demonstração da aplicação do *randow forest* e suas divisões com números de *nodesize* e um *ntree=300.* Esses valores são importantes para definição dos resultados e qualificação das métricas para os próximos passos no treino modelo.

```
Call:
 randomForest(formula = pickups ~ ., data = TRAIN_SET, importance = T,      mtry = 24, nodesize = 3, ntree = 300)
               Type of random forest: regression
                     Number of trees: 300
No. of variables tried at each split: 24

          Mean of squared residuals: 63688.29
                    % Var explained: 96.01
```

Taxa de erro das árvores

Aqui como nos modelos mais acima foi levado em conta a taxa de erro de acordo com o número de arvores, isso significa que a partir do valor 100 aproximadamente as arvores não apresentam mais pouca variação na taxa de erro.



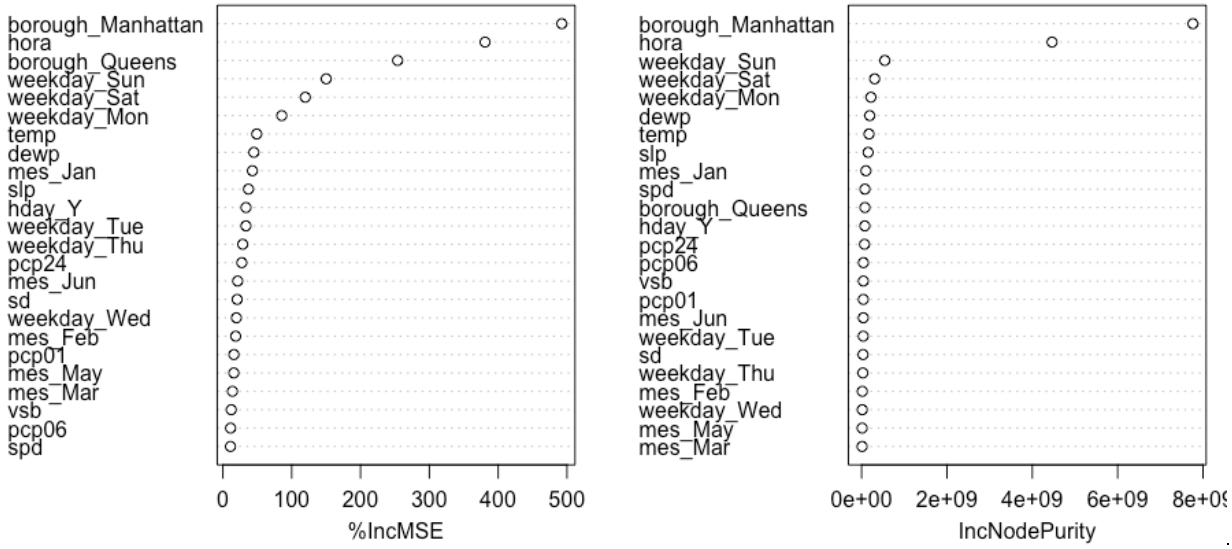
Performance do modelo entre treino e teste com *Random Forest*

Abaixo a performance do modelo de treino e teste para validar se há alguma diferença entre a base de treino e base de teste aplicado ao algoritmo do *Randow Forest*. De acordo com a figura abaixo o indicador *Rsquared* não apresenta muita variação por este motivo entendo que é possível a replicação e ele tem um alta poder de abrangência.

```
> postResample(pred = Y_VAL_TRAIN, obs = TRAIN_SET$pickups)
       RMSE    Rsquared         MAE
428.4084094   0.8850628 265.9517338
> postResample(pred = Y_VAL_TEST,  obs = TEST_SET$pickups)
       RMSE    Rsquared         MAE
426.1066535   0.8846188 263.3040971
```

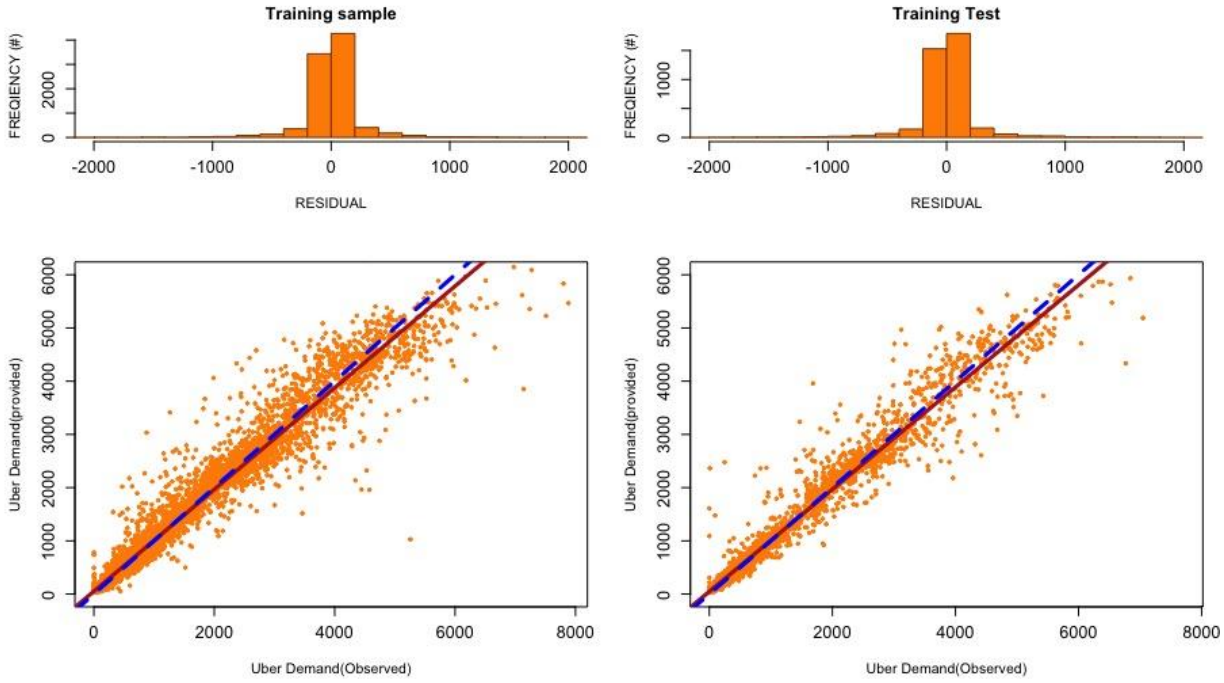
Importância das variáveis com *Random Forest*

Demonstração das variáveis mais importantes após execução do algoritmo do *randow forest,* com isso fica diferente da percepção indicadas seção de análise dos dados mais acima onde apresentei a variável *pickups* porém aqui a variável mais importante é o *borough_manhattan* como visto na figura abaixo.



Visualizando os dados usando algoritmo *Random Forest*

Aqui apenas para detalhar os dados acima demonstrados onde não há muita diferença ou expressiva diferença entre a base de treino e teste aplicando o algoritmo de *randow forest,* significando que há reprodutibilidade neste modelo, abaixo um histograma e a regressão linear comprovando que a distribuição é bem próxima de 0 indicando com a linha pontilhada por cima da linha contínua laranja, muito próximas.



Variabilidade dos resultados por bairro entre treino e teste

Como a variável mais importante dentro do modelo de *randow forest* foi o bairro de Manhattan abaixo apliquei a comparação com os outros 3 bairros para verificar a variabilidade dos modelos entre treino e teste. Onde indicou que por bairro a variabilidade ou reprodutibilidade deste modelo se aplica para identificação de demandas por viagens, demonstradas nos 3 casos na sequencia abaixo.

Manhattan

```
> postResample(pred = Manhattan_train$pred, obs = Manhattan_train$pickups)
      RMSE   Rsquared        MAE
423.455727   0.913892 269.448278
> postResample(pred = Manhattan_test$pred,  obs = Manhattan_test$pickups)
       RMSE    Rsquared         MAE
428.3910527   0.9091081 276.1816141
```

Queens

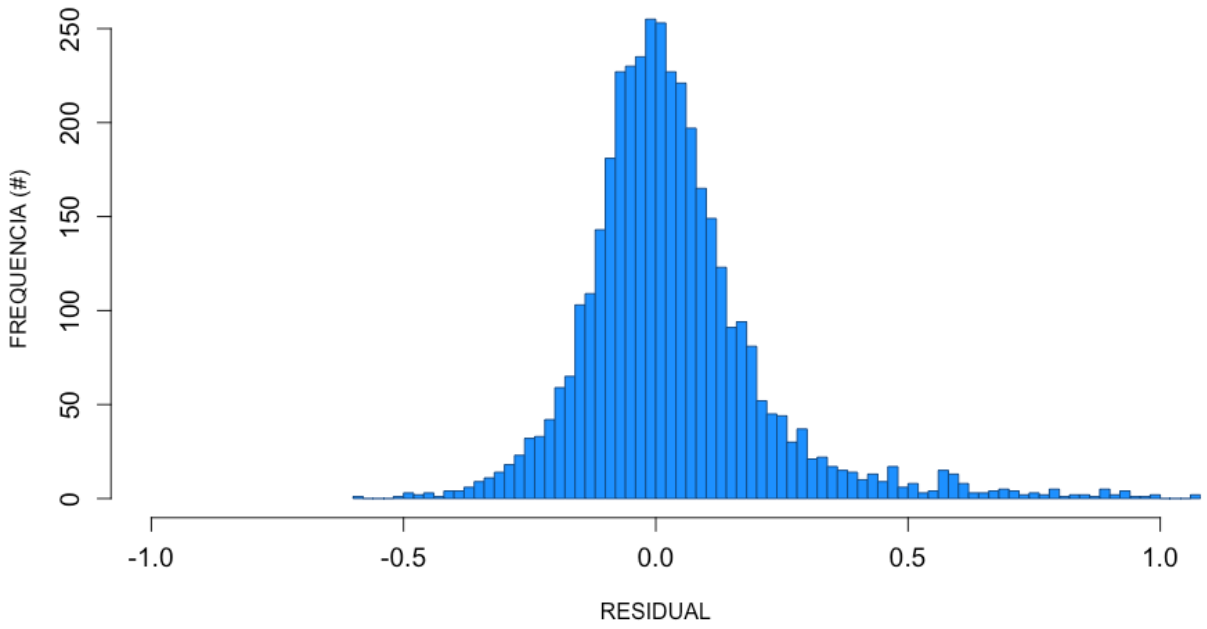
```
> postResample(pred = Queens_train$pred, obs = Queens_train$pickups)
      RMSE   Rsquared        MAE
52.2306729  0.8878047 37.7370784
> postResample(pred = Queens_test$pred,  obs = Queens_test$pickups)
      RMSE   Rsquared        MAE
51.9705292  0.8833646 38.1653438
```

Brooklin

```
> postResample(pred = Brooklyn_train$pred, obs = Brooklyn_train$pickups)
      RMSE   Rsquared        MAE
80.5795825  0.9245494 52.0120899
> postResample(pred = Brooklyn_teste$pred,  obs = Brooklyn_teste$pickups)
      RMSE   Rsquared        MAE
75.9333558  0.9368743 50.3772726
```

Sendo assim, é importante destacar que nesta parte ainda fica pendente se houve ou não uma superestimação ou subestimação do modelo que independente dos casos as figuras acima aplicam apenas reprodutibilidade, até onde entendo.

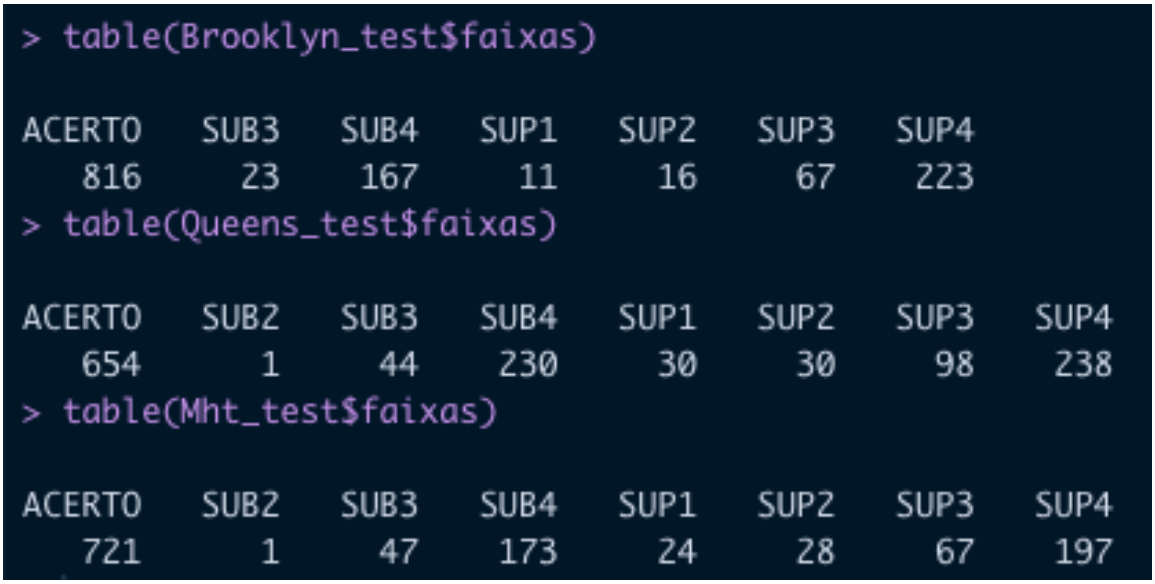
Já na figura abaixo é onde fiz a aplicação da sub e superestimação do modelo que acertou, sendo assim aplicável para realização da escolha dentro da faixa que a área de negócio determinou como acima de 60%. Houve pouquíssima subestimação e houve um número maior de superestimação.



A representação geral da superestimação e subestimação representado na figura abaixo.

ACERTO	SUB2	SUB3	SUB4	SUP1	SUP2	SUP3	SUP4
2191	2	114	570	65	74	232	658

A distribuição dos acertos por bairro, onde houve mais superestimação do que sub conforme a figura abaixo.



Considerações Finais

A partir dos resultados encontrados entendo que o modelo conseguiu acertar de maneira geral de acordo com todas as informações demonstradas acima. Sob meu entendimento das aulas considero alguns pontos abaixo:

- Que sim, com as informações apresentadas acima é possível construir um modelo com uma forte abrangência e muita reprodutibilidade de acordo com os experientes realizados entre os *dataset* de treino e teste.
- Que as variáveis borough_manhattan e pick-ups são as variáveis importantes para construção destes modelos.
- De acordo com os modelos apresentados, não seria necessário um modelo por região pois há muita reprodutibilidade entre os modelos de treino e teste.
- De acordo com o que estudei e apliquei o maior insight que tiro aplicando os dois modelos é que o de *random forest* é mais amplo e tem mais índices para avaliar se um modelo que tenha reprodutibilidade dentro dos seus parâmetros, isso porque as representações dos dados são mais estruturados e também há mais parâmetros que puder perceber para treinar o modelo.
- Entendo que o modelo de *random forest* é o mais adequado, no meu entendimento e observação dos dados,

Obrigado Prof. João me tirou uma nuvem dos olhos para trabalhar com dados.