Fazendo predições sobre os valores de aluguéis residênciais.

O obejetivo é fazer predições a cerca do valor médio do aluguel residêncial. O conjunto de dados possue um total de 10962 casas. Os dados foram obtidos através do Kaggle: https://www.kaggle.com/rubenssjr/brasilian-houses-to-rent

Importação dos dados

```
setwd("D:\\07 04 2020\\Documents\\Edgar\\Projetos\\Data_sciece\\DataScience_projects\\Aluguel_casa")
library("tidyverse")
library("rpart")
library("randomForest")
```

```
## Warning: package 'randomForest' was built under R version 4.0.4
```

```
dados<- read.csv("houses_to_rent_v2.csv", header = T)
View(dados)
```

Limpeza e Preparação dos dados

Organização do cabeçalho para que seja possível manusear as variáveis com mais facilidade. E retirando linhas em que não consta a indicação do andar.

```
# Exlcuindo os caracteres ..R.. que estão depois do nome da variavel.
str(dados)
```

```
## 'data.frame':    10692 obs. of  13 variables:
##  $ city               : chr  "SÃ£o Paulo" "SÃ£o Paulo" "Porto Alegre" "Porto Alegre" ...
##  $ area               : int  70 320 80 51 25 376 72 213 152 35 ...
##  $ rooms              : int  2 4 1 2 1 3 2 4 2 1 ...
##  $ bathroom           : int  1 4 1 1 1 3 1 4 2 1 ...
##  $ parking.spaces     : int  1 0 1 0 0 7 0 4 1 0 ...
##  $ floor              : chr  "7" "20" "6" "2" ...
##  $ animal             : chr  "acept" "acept" "acept" "acept" ...
##  $ furniture          : chr  "furnished" "not furnished" "not furnished" "not furnished" ...
##  $ hoa..R..           : int  2065 1200 1000 270 0 0 740 2254 1000 590 ...
##  $ rent.amount..R..   : int  3300 4960 2800 1112 800 8000 1900 3223 15000 2300 ...
##  $ property.tax..R..  : int  211 1750 0 22 25 834 85 1735 250 35 ...
##  $ fire.insurance..R..: int  42 63 41 17 11 121 25 41 191 30 ...
##  $ total..R..         : int  5618 7973 3841 1421 836 8955 2750 7253 16440 2955 ...
```

```
cab<-names(dados)
cab<-cab%>% str_remove_all("..R..")
colnames(dados)<- cab
excl<- dados$floor%>%str_which("-")
dados<- dados[-excl,]
ndados<- nrow(dados)
```

Os dados são separados em treino e teste, sendo 80% para treinar e 20% para testar o modelo.

```r
n_treino<- (70/100)*ndados
set.seed(100)
ua<- sample(n_treino)
treino<- dados[ua,]; View(treino)
teste<- dados[-ua,]; View(teste)
```

Treinamento do modelo

Modelo construído utilizando regressão linear múltipla.

```r
model<- lm(total~.,data= treino)
```

Avaliação da performance do modelo

Para a avaliação da performance do modelo foi utilizado três métricas, sendo elas erro médio absoluto, percentual médio do erro absoluto e coeficiente de determinação (R^2). Também foi avaliado a distribuição dos erros dentro de quartis.

```r
predito<-predict(model,teste)
per<- teste %>% select("city", "total")%>%
  mutate(predito) %>% mutate(erro= total-predito )%>%
  mutate(erro_abs= abs(erro))%>%mutate(erro_perc= erro/total)%>%
  mutate(erro_percabs= abs(erro_perc))
per[,c(4:7)]<-round(per[,c(4:7)], 5)

# Calculando o erro medio absoluto e percentual medio
erro_medio<- mean(per$erro_abs)
erro_percmed<- mean(per$erro_percabs)
summary(per$erro_percabs)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0000300 0.0000700 0.0002436 0.0001700 0.2589200
```

Cálculo do coeficiente de determinação (R^2), ele varia entre 0 e 1 e indica o quão bem ajustado o modelo está, quanto mais préximo de 1 melhor a performance do modelo.

```r
# Calculando o coeficiente de determinacao
resumo<- summary(model)
r2<- resumo$adj.r.squared
per_mod<- data.frame(erro_medio, erro_percmed,r2)
per_mod
```

```
##   erro_medio erro_percmed        r2
## 1  0.7663743 0.0002435668 0.9999999
```

Treinamento do modelo

Treinamento do modelo utilizando árvore de regresão.

```r
model2<- rpart(total~., data = treino)
```

Avaliação da performance do modelo

Para a avaliação da performance do modelo foi utilizado duas métricas, sendo elas erro médio absoluto, percentual médio do erro absoluto. Também foi avaliado a distribuição dos erros dentro de quartis.

```
predito<-predict(model2,teste)
per2<- teste %>% select("city", "total")%>%
  mutate(predito) %>% mutate(erro= total-predito )%>%
  mutate(erro_abs= abs(erro))%>%mutate(erro_perc= erro/total)%>%
  mutate(erro_percabs= abs(erro_perc))
per[,c(4:7)]<-round(per[,c(4:7)], 5)

# Calculando o erro medio absoluto e percentual medio
erro_medio2<- mean(per2$erro_abs)
erro_percmed2<- mean(per2$erro_percabs)
per_mod2<- data.frame(erro_medio2, erro_percmed2)
per_mod2
```

```
##   erro_medio2 erro_percmed2
## 1    1911.981     0.5758513
```

```
summary(per2$erro_percabs)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000784 0.186709 0.380779 0.575851 0.698671 5.846700
```

Treinamento do modelo

Treinamento do modelo utilizando Random Forest.

```
model3<- randomForest(total~., data = treino,
                      ntree= 100, proximity= T)
```

Avaliação da performance do modelo

Para a avaliação da performance do modelo foi utilizado duas métricas, sendo elas erro médio absoluto, percentual médio do erro absoluto. Também foi avaliado a distribuição dos erros dentro de quartis.

```
predito<-predict(model3,teste)
per3<- teste %>% select("city", "total")%>%
  mutate(predito) %>% mutate(erro= total-predito )%>%
  mutate(erro_abs= abs(erro))%>%mutate(erro_perc= erro/total)%>%
  mutate(erro_percabs= abs(erro_perc))
per3[,c(4:7)]<-round(per3[,c(4:7)], 5)

# Calculando o erro medio absoluto e percentual medio
erro_medio3<- mean(per3$erro_abs)
erro_percmed3<- mean(per3$erro_percabs)
per_mod3<- data.frame(erro_medio3, erro_percmed3)

summary(per3$erro_percabs)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00000  0.00540  0.01230  0.03504  0.02474 23.05960
```

Comparação entre os três modelos

```
per_mod1<- per_mod[,-3]
Modelo<- c("Regressão Linear", "Arv. Regressão", "Random Forest")
erro_med<-c(per_mod$erro_medio, per_mod2$erro_medio2, per_mod3$erro_medio3)
erro_permed<-c(per_mod$erro_percmed, per_mod2$erro_percmed2,
               per_mod3$erro_percmed3)
resultado<- cbind(Modelo, erro_med, erro_permed)
resultado
```

```
##      Modelo             erro_med            erro_permed
## [1,] "Regressão Linear" "0.766374311740891" "0.000243566801619433"
## [2,] "Arv. Regressão"   "1911.98098579908"  "0.575851318110582"
## [3,] "Random Forest"    "186.284159012146"  "0.0350397651821862"
```

A regressão linear apresentou a melhor perfomance de predição para os valores de aluguel residencial (R^2= 0.999). Com um erro médio de 0.766 e um erro percentual de 0.0002%. Já Os algoritmos de árvore de regressão e random forest apresentaram 57 e 3.5% por cento de erro médio.