Aluguel de casas: fazendo predições sobre o valor de aluguel decasas.

Importação dos dados

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

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("rpart")
library("randomForest")
```

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

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

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

Limpeza e Preparação dos dados

Organização do cabeçalho para que seja possível manusear as variáveis com mais facilidade.

```
# Exlcuindo os caracteres ..R.. que estão depois do nome da variavel.
cab<-names(dados)
cab<-cab%>% str_remove_all("..R..")
colnames(dados)<- cab
```

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 0.0000600 0.0002021 0.0001500 0.2589400
```

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  0.7203906  0.000202048 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     1976.68     0.6174459
```

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000388 0.188901 0.385246 0.617446 0.720171 6.023329
```

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.
##  0.000000  0.005537  0.011725  0.034447  0.024032 10.178200
```

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" "0.720390595386534" "0.000202048004987531"
## [2,] "Arv. Regressão"   "1976.67992501656"  "0.617445941510005"
## [3,] "Random Forest"    "183.223950888404"  "0.0344469856608479"
```