Universidade Estadual de Santa Cruz – UESC

Atividade 07 - Árvores de decisão

Documentação referente a atividade 07 da disciplina "Tópicos Avançados em Computação" feito por Breno Piropo, Bruno Santos, Flavia Jesus e Vítor Coutinho.

Ilhéus – BA
2024

Descrição do dataset

Nosso dataset é composto por dados gerados aleatoriamente sobre produção de laranjas e algumas variáveis relacionadas a esse contexto.

Usando o summary para entender sobre os dados, percebemos que há 2 colunas com dados não numéricos, Cidade e Clima:

```
> summary(laranjas)
    Cidade               Ano        Área_Plantada      Produção        Clima           Fertilizantes        Preço       
 Length:100         Min.   :2010   Min.   :101.3   Min.   :451.0   Length:100         Min.   : 70.51   Min.   : 532.6  
 Class :character   1st Qu.:2012   1st Qu.:280.8   1st Qu.:481.0   Class :character   1st Qu.:168.45   1st Qu.: 865.9  
 Mode  :character   Median :2014   Median :482.7   Median :495.5   Mode  :character   Median :190.46   Median :1089.6  
                    Mean   :2015   Mean   :494.7   Mean   :497.3                      Mean   :192.69   Mean   :1072.4  
                    3rd Qu.:2016   3rd Qu.:665.1   3rd Qu.:512.2                      3rd Qu.:221.46   3rd Qu.:1298.4  
                    Max.   :2020   Max.   :999.7   Max.   :563.0                      Max.   :349.08   Max.   :1499.3  
```

Separando os valores únicos, temos os seguintes valores para essas colunas e suas respectivas quantidades:

```
> table(laranjas$Cidade)

Belo Horizonte       Curitiba   Porto Alegre Rio de Janeiro      São Paulo 
            16             23             14             17             30 

> table(laranjas$Clima)

Subtropical   Temperado    Tropical 
         30          37          33 
```

Em relação a coluna Ano, podemos descrever ela de uma maneira que faça mais sentido sobre o tipo de dado, nesse caso, representando como valores não numéricos:

```
> table(laranjas$Ano)

2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 
   8   11    9    7   17   14   11    5    5    8    5 
```

Criando a árvore de decisão

Decidimos avaliar como os fatores afetam a produção. Para isso precisamos criar uma nova coluna no dataset que transforma os dados numéricos da produção em dados categóricos, no caso “Baixa”, “Média” e “Alta”.

```
# definindo os limites para o corte
high_limit <- quantile(laranjas$Produção, 0.75) # corta em 75%
low_limit <- quantile(laranjas$Produção, 0.25) # corta em 25%

# nova coluna no dataset com as categorias
laranjas$cat_prod <- cut(laranjas$Produção,
                         breaks = c(-Inf, low_limit, high_limit, Inf),
                         labels = c("Baixa", "Média", "Alta"))
```

Como podemos ver acima, definimos dois limites para os dados. O limite em 0.25 indica que todos os valores da coluna produção abaixo do primeiro quartil serão considerados “Baixa”, os valores entre os limites será considerado “Média” e os valores acima de 0.75 será considerado “Alta”.

Após isso separamos os valores dessa nova coluna em dois tipos, valores para treino e valores para teste:

```
# Definir uma semente para reprodutibilidade
set.seed(42)

# Dividir os dados
train_index <- createDataPartition(laranjas$cat_prod, p = 0.8, list = FALSE)
train_data <- laranjas[train_index, ]
test_data <- laranjas[-train_index, ]
```

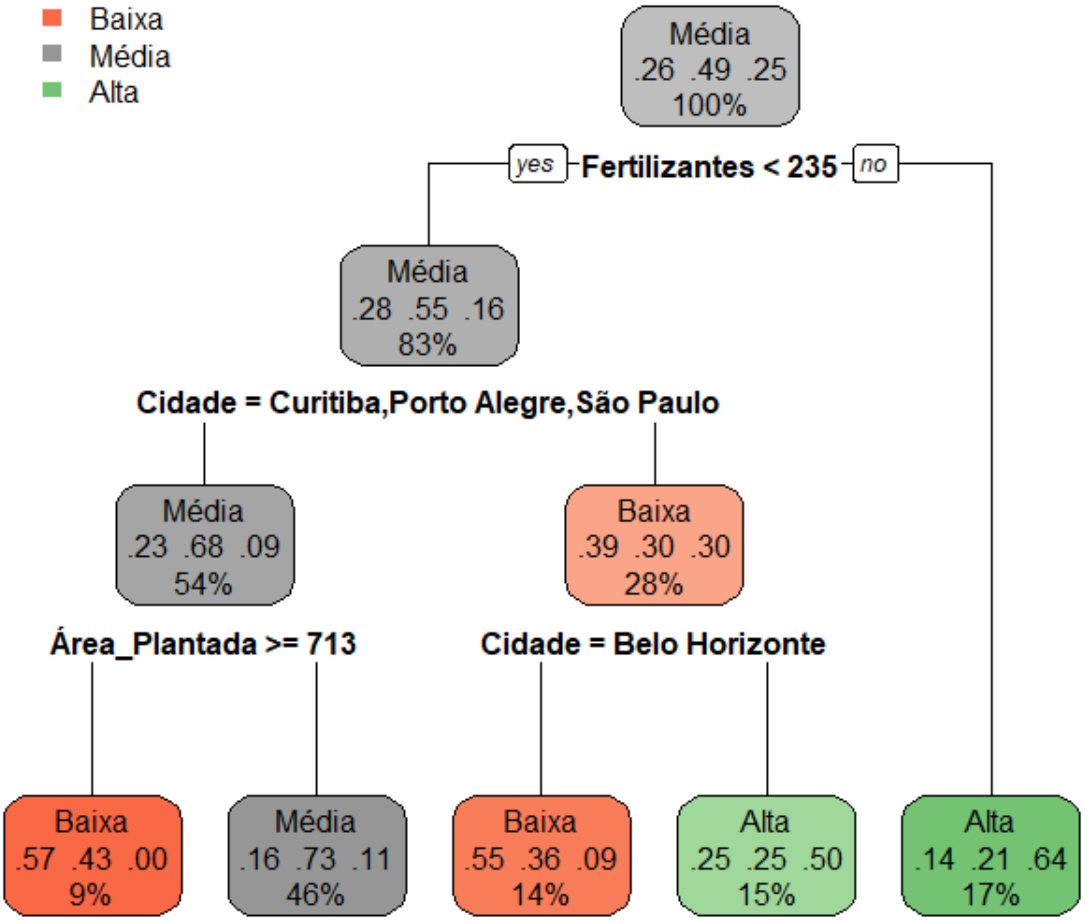
Como podemos ver na imagem acima, utilizamos a função createDataPartition da biblioteca caret para separar 80% dos dados para treino e 20% para teste.

Com os dados para treino definidos, geramos uma árvore de decisão utilizando a coluna “cat_prod” em função das outras colunas, como podemos ver a seguir:

```
tree_model <- rpart(cat_prod ~ Cidade + Ano + Área_Plantada + Clima + Fertilizantes + Preço,
                    data = train_data, method = "class")

rpart.plot(tree_model)
```

Lembrando que utilizamos o método “class” pois estamos lidando com um problema de classificação. Para montar a árvore de decisão utilizamos a biblioteca “rpart” e para plotar utilizamos a biblioteca “rpart.plot”. Como podemos ver a seguir:



Na árvore acima a parte esquerda significa que estamos cumprindo a verificação do nó, por exemplo, no nó raiz temos que se a quantidade de fertilizante for menor que 235 vamos fazer as análises para os nós da esquerda e caso essa verificação não se cumpra, vamos para o lado direito da árvore.

Cada nó conta com três valores numéricos, por exemplo, no nó raiz temos .26 .49 .25, esses números são a proporção dos dados que estão sendo analisados por aquele no que tem respectivamente produção baixa, média e alta, cada número desses representa a proporção de dados que tem essa categoria.

A porcentagem abaixo desses três números significa a porcentagem do total de dados que está sendo analisado pelo nó, por exemplo, no nó raiz temos a divisão de 83% para a esquerda e 17% para a direita, somando os dois teremos 83% + 17% = 100%, essa porcentagem é a quantidade de dados que o nó está pegando do nó pai.

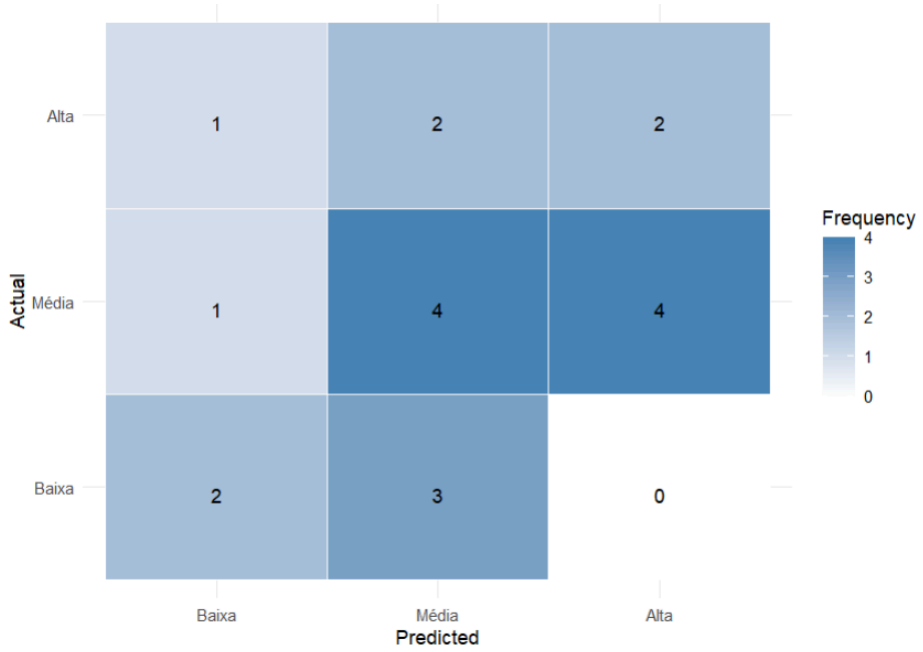
A interpretação que podemos ter dessa árvore é a seguinte: Com base na árvore que temos aqui suponha que queremos estudar um jeito de melhorar a produção de laranjas, então vamos olhar a árvore, partindo do nó raiz podemos olhar da seguinte forma, se vamos usar uma quantidade grande de fertilizante então vamos olhar para o ramo esquerdo do da árvore, depois olhamos para a região, se estivermos plantando nas cidade de Curitiba, Porto

Alegre ou São Paulo olhamos para a esquerda, vamos supor que estamos de São Paulo, vamos para a esquerda, agora vamos olhar para a área que estamos plantando, se for uma área maior que 713 então teremos uma produção baixa, se for uma área menor que isso teremos uma produção média.

Basicamente temos que saber a quantidade de fertilizante que vamos usar, a cidade em que vamos plantar e o tamanho da área em que vamos plantar.

Analisando os resultados

Aplicando o modelo criado anteriormente como predição para os dados de teste, obtemos os resultados para essa previsão. Aplicando uma matriz de confusão a esses resultados obtivemos:



A matriz acima indica como as classes foram preditas. Podemos ver que a classe “Baixa” foi predita 2 vezes como “Baixa”, 1 vez como “Média” e 1 vez como “Alta”, a classe “Média” foi predita 3 vezes como “Baixa”, 4 vezes como “Média” e 2 vezes como “Alta”, a classe “Alta” não foi predita como “Baixa”, foi predita 4 vezes como “Média” e 2 vezes como “Alta”.

```
Overall Statistics

               Accuracy : 0.4211
                 95% CI : (0.2025, 0.665)
    No Information Rate : 0.4737
    P-Value [Acc > NIR] : 0.7534

                  Kappa : 0.0913

 Mcnemar's Test P-Value : 0.4459
```

Na imagem acima temos, em sequência: Accuracy, que é a proporção de previsões corretas entre todas as previsões; Intervalo de confiança, onde é o intervalo dentro do qual a acurácia real do modelo está com 95% de confiança; No information rate (NIR), taxa de accuracy se o modelo sempre previsse a classe mais frequente; P-value, A probabilidade de observar uma accuracy maior do que a NIR por acaso. Um valor maior que 0.05 indica que a acurácia não é significativamente melhor que a NIR; Kappa, Métrica que ajusta a acurácia levando em conta as previsões corretas ao acaso. Valores próximos de 0 indicam baixo acordo entre previsão e verdade; Mcnemar 's Test P-Value, Teste para comparar a diferença nas taxas de erro de classificação. Um valor maior que 0.05 indica que não há diferença significativa nas taxas de erro entre as classes.

Resumindo, o modelo teve uma accuracy de 42% e um kappa igual a 0.0913, próximo de 0 indicando um baixo acordo entre previsão e verdade.

Statistics by Class:

	Class: Baixa	Class: Média	Class: Alta
Sensitivity	0.4000	0.4444	0.4000
Specificity	0.8571	0.5000	0.7143
Pos Pred Value	0.5000	0.4444	0.3333
Neg Pred Value	0.8000	0.5000	0.7692
Prevalence	0.2632	0.4737	0.2632
Detection Rate	0.1053	0.2105	0.1053
Detection Prevalence	0.2105	0.4737	0.3158
Balanced Accuracy	0.6286	0.4722	0.5571

Sensibilidade: O modelo detecta entre 40% e 44.44% dos casos reais de cada classe. A classe Média tem a maior sensibilidade. Especificidade: A classe Baixa tem a maior especificidade (85.71%), indicando que o modelo é bom em identificar corretamente os casos que não são da classe Baixa. Valor Preditivo Positivo: As previsões do modelo são mais confiáveis para a classe Baixa (50% de precisão) do que para Média e Alta. Valor Preditivo Negativo: O modelo tem um bom desempenho em não prever as classes incorretamente, especialmente para a classe Alta (76.92% de valor preditivo negativo). Acurácia Balanceada: A classe Baixa tem a maior acurácia balanceada (62.86%), enquanto a classe Média tem a menor (47.22%).

Em resumo, esses resultados indicam que o modelo tem desempenho variado entre as classes, com desempenho geral mediano. A baixa precisão e acurácia balanceada sugerem que há espaço significativo para melhorar o modelo,

Possibilidade do SVM:

Quanto ao outro algoritmo visto em sala (o SVM), foi pensado na possibilidade de implementação para o nosso dataset, entretanto, para o dataset de laranjas, que contém características como *Cidade, Ano, Área Plantada, Clima, Fertilizantes, Preço,* e foi posteriormente adicionada a variável *Categoria,* a dimensionalidade não é alta, pois o número de variáveis independentes é relativamente pequeno. O que acaba por descartar o uso do método, e mantendo apenas a Árvore de Decisão.

Possibilidade do KNN:

O método K-Nearest Neighbor(KNN) é um algoritmo de aprendizado supervisionado usado tanto para classificação quanto para regressão.
O objetivo desse algoritmo é basicamente prever a classe (em problemas de classificação) ou o valor contínuo (em problemas de regressão) de uma amostra de dados com base nas classes ou valores das amostras mais próximas no espaço de características.
Ele serviria bem para o nosso objetivo pois o que queremos é prever as classes em que os dados de teste vão estar, e nesse ponto o algoritmo funciona muito bem.
Entretanto, teríamos que normalizar (padronizar a escala dos dados) os dados pois ele é sensível à escala das características.