**APS – Modelagem Preditiva**

Otávio Farhat Fernandes

**Introdução:**

Este trabalho tem o objetivo de aplicar o conteúdo estudado nas aulas de Modelagem Preditiva para realizar a análise de duas bases de dados: a) *churn*: informações sobre os clientes de uma instituição bancária, tentar determinar se o cliente cancelará o serviço ou não; b) *used\_cars*: informações sobre os preços de veículos usados da Mercedes, tentar prever o preço. Para isso, utilizarei os métodos de Regressão Logística, Árvore de Classificação, *Random Forest* e *Boosting*. As bases serão separadas em um conjunto de treino, onde o modelo será obtido, e um conjunto de testes, onde o modelo será aplicado para verificar sua capacidade de previsão.

Depois da realização dos métodos para cada base de dados, será realizada uma discussão quanto à acurácia no conjunto de testes. Para isso, serão comparadas as curvas ROC de cada método para a base *churn* e para a base *used\_cars*, utilizarei a raiz quadrada do erro quadrático médio de teste para observar qual modelo foi “melhor”.

O trabalho está separado em duas partes. Na primeira, a Parte Teórica, dou uma breve explicação histórica de cada método e como funcionam seus mecanismos. Na segunda, a Parte Aplicada, separo em duas seções onde analisarei as bases de dados, apresento os códigos do R utilizados e realizo uma discussão dos dados.

1. **Parte Teórica:**

Para prover uma explicação de como funciona uma *Random Forest*, temos que primeiro explicar as árvores de regressão individuais e o mecanismo de *bagging*.

* 1. **Árvores de Regressão:**

As árvores de regressão fazem parte do conjunto de instrumentos de aprendizagem supervisionada. Ou seja, possui tanto a variável explicada (y) quanto as explicativas (x). para as árvores de regressão, y vai ser um número real, não vai ser um valor discreto, como 0 ou 1, por exemplo, como é o caso de classificações. Utilizam o CART (*Classification And Regression Trees*) como algoritmo para criar suas separações. Esse mecanismo foi introduzido por Breiman et al. (1984).

O CART vai primeiro definir a raiz da árvore, e a partir de um parâmetro de decisão que será discutido a seguir, vai criando as divisões da árvore a partir de nós, criando níveis diferentes. Pegando um conjunto de dados, o CART vai dividir esses dados (vai criar um split) em conjuntos menores, como pode ser observado na imagem abaixo para o caso de duas variáveis.

Gráfico, Gráfico de dispersão, Gráfico de bolhas

Descrição gerada automaticamente

Gráfico, Gráfico de dispersão

Descrição gerada automaticamente

No exemplo, temos dados do preço de uma casa, representado pelo valor abaixo dos pontos, temos a taxa de criminalidade do local em que está a casa e a distância até o centro da cidade. Admitindo que estes são os dados de treinamento, ou seja, apenas uma parte dos conjunto total de dados, o CART vai separá-los em splits, de acordo com uma medida. No caso, utilizamos o EQM (Erro Quadrático Médio). O CART vai encontrar o método de separação dos dados que minimize o EQM, formando os retângulos verificados na figura.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

A imagem acima representa a árvore formada pelo CART. Nos nós, temos a média dos valores dentro de cada retângulo. Começando na parte de cima, caso a distância do centro seja menor que 3.2, a previsão do valor da casa será de $8,1. Caso contrário, se a taxa de criminalidade for menor do que 35%, o valor previsto é de 3 etc.

Para verificar a eficiência do modelo, deve-se utilizar o conjunto de testes, ou seja, os dados que não foram utilizados após a escolha dos dados de treinamento. Então, rodamos um modelo de Regressão Linear, um OLS (*Ordinary Least Squares*), e verificamos qual possui a melhor previsão.

* 1. ***Bagging***

O *Bootstrap Aggregation* (*Bagging*) foi introduzido por Leo Breinman em 1996. O objetivo desse mecanismo é de reduzir a variância de algum método de aprendizagem. Para isso, utiliza da técnica de *Bootstrap*, proposta em 1979, que consiste em formar um conjunto de dados a partir do sorteio de observações dentro de uma amostra, com reposição.

O *Bootstrap* se vale do Teorema de Glivenko-Cantelli, onde a Função de Distribuição Empírica vai se aproximar da Função Geradora dos Dados na medida em que a quantidade de observações coletadas aumenta. Nisso, o objetivo do *bootstrap* é obter uma estimativa do erro quadrático médio de um estimador através do erro quadrático médio de amostras da função de distribuição empírica.



O *Bagging* basicamente permite utilizar o *Bootstrap* para agregar diversas árvores de decisão de modo que a performance da árvore melhore, dado que árvores muito altas podem ter uma variância e um viés grandes, tornando-as sensíveis a pequenas mudanças nos dados. O *Bagging* permite que se contorne o problema de não ter diversas bases de dados para treinar o método de aprendizagem. Para uma árvore de regressão, são geradas B amostras *bootstrap* que servirão como dados de treinamento para cada regressor treinado, e o regressor agregado será igual à média deles. A variância do método será derrubada quando se tira essa média.

Abaixo temos o esquema de funcionamento do *Bagging* e qual será a previsão para um problema de regressão.

Gráfico

Descrição gerada automaticamente com confiança média

* 1. **Random Forests:**

*Random Forest* é um método de aprendizagem supervisionada de classificação e de regressão de alta performance. É um método que utiliza árvores mais altas, via CART e é muito bem-sucedido principalmente para dados tabulares. A *Random Forest* foi inventada por Leo Breiman no começo dos anos 2000. Dado um conjunto de dados, a *Random Forest* usa a ideia de *Bagging* de árvores de classificação ou regressão e acrescenta um novo elemento.

O *Bootstrap* vai pegar uma tabela de dados e amostrar cada linha com repetição, onde o fundamento matemático disso é a função de distribuição empírica e o Teorema de Glivenko-Cantelli. A partir disso, formam-se B novas tabelas com dados originados da amostragem, e é como se tivessem vindo da mesma distribuição. De cada tabela, teremos uma árvore de classificação ou de regressão, dependendo do problema que está sendo avaliado. As B árvores formadas serão diferentes entre si, pois as amostras são diferentes.



Para formar os preditores pelo *Bagging*, varia se o problema é de classificação ou de regressão. Para classificação, pode utilizar o voto da maioria. o estimador será a classe majoritária dentro do conjunto de previsões dos B preditores. Para regressão, a previsão *Bagging* de uma nova observação será a média das previsões de cada uma das B árvores.

Algumas observações importantes são: 1) as árvores não são podadas ao fazer o *Bagging*, elas podem ser altas, onde o viés será baixo e a variância alta; 2) o *Bagging* foi criado para reduzir a variância, mas pode não funcionar se a covariância entre as previsões for alta; para diminuir as covariâncias, Breiman introduziu uma nova ideia que deu origem às *Random Forests*, que consiste em reduzir a correlação entre as árvores. Para isso, utilizou a ideia de subespaço aleatório. No primeiro split da árvores, utiliza apenas um subconjunto aleatório m de variáveis preditoras para decisão. Em cada split de cada árvores, utiliza essa ideia, mas sem reposição. 3) Erro Out-of-bag, onde aproximadamente 37% das observações não entram na amostra gerada pelo *Bootstrap*.

Para encontrar quais são as variáveis mais importantes se verifica as que estão em splits mais acima e que foram utilizadas mais vezes nas decisões. As duas medidas de importância dada para uma variável na *Random Forest* são: 1) quanto que a precisão diminui quando a variável é excluída; 2) quanto a medida de impureza Gini diminui quando uma variável é escolhida para dividir um nó. Para medir a primeira, se utiliza a amostra *out-of-bag*, que não foi utilizada no processo de construção das árvores. Se calcula a acurácia da predição nessa amostra, depois, os valores das variáveis são embaralhados, e por último, se calcula a diminuição da acurácia da predição na amostra *out-of-bag* embaralhada1.

Para se calcular o intervalo de confiança para uma predição de uma Random Forest se utiliza o método de Jackknife

1. **Parte Aplicada:**
   1. **Churn**

**Objetivo:** prever a variável Exited, que indica se o cliente cancelará o serviço ou não.

Importação das Bibliotecas e da base de dados:

Separação em dados de teste e de treinamento:

Regressão Logística:

Árvore de Classificação:

Random Forest:

Boosting:

Curvas ROC:

Gráfico

Descrição gerada automaticamente

AUC:

* 1. **Used\_cars**

**Objetivo:** prever a variável price.

Importação das Bibliotecas e da base de dados:

Separação em dados de teste e de treinamento:

Regressão Linear Múltipla:

Árvore de Regressão:

Random Forest:

Boosting:

Raiz Quadrada do Erro Quadrático Médio:

**Referências:**

**https://www.displayr.com/how-is-variable-importance-calculated-for-a-random-forest/**