Projeto Demonstrativo 2

*

Matheus Eiji Endo
Estudante de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
matheus.endo@hotmail.com

Johannes Peter Schulte
Estudante de Ciência da Computação
Universidade de Brasília
Brasília, Brasil
johpetsc@gmail.com

***Resumo*—Relatório referente ao projeto demonstrativo 2 da matéria Fundamentos de Sistemas Inteligentes.**

***Index Terms*—Árvores de decisão, Florestas Randômicas e atributos.**

I. Introdução

Essa relatório apresenta alguns conceitos importantes acerca de modelos de classificação, como Àrvores de Decisão e Florestas Randômicas. Além disso aborda o problema da classificação de espécies de plantas com base em suas folhas e os resultados obtidos através do uso de Florestas Randômicas.

II. Árvores de Decisão

Árvores de decisão é modelo de classificação supervisionado e não-paramétrico, onde há treinamento com dados rotulados mas não é escolhida uma forma pré-definida para a função de transferência, baseado na estrutura de dados conhecido como Árvore, onde os dados são armazenados de forma hierárquica em nós, onde os nós folha são aqueles que não possuem ramos(nós abaixo deles) e o nó raiz aquele que está mais 'acima' na hierarquia.Nesse modelo utiliza-se os dados rotulados para criar-se uma árvore, onde os nós chamados de decisão são os aqueles que testam um atributo, os ramos desses nós são equivalentes a um valor do determinado atributo testado e as folhas são associadas às classes. A partir da árvore obtida no treinamento o algoritmo a usa como base para classificar outros dados, de forma recursiva, partindo do nó raiz, a cada nó de decisão, analisa-se o resultado do teste avaliando o atributo do dado e a partir desse resultado tomar um novo nó raiz da subárvore continua-se realizando os testes para os diferentes nós de decisão subsequentes até chegar em uma folha(classe).

É interessante observar que para um mesmo conjunto de dados pode-se criar diferentes Árvores, mudando o nó raiz e a ordem dos nós de decisão, sendo assim podem haver Árvores que sejam mais eficientes em termos de custo computacional. Para criar-se uma árvore mais eficiente, na hora de sua construção deve-se tentar associar a cada nó de decisão o 'melhor' atributo entre os ainda não testados no determinado caminho. Uma forma de se avaliar o 'melhor' atributo é utilizar a entropia como medida de impureza e comparar a entropia do nó raiz com o seus ramos de forma a maximizar o ganho de informação, como é utilizado no algoritmo ID3, de forma que o grau de impureza é máximo se o número de objetos das classes são iguais e mínima quando os objetos são todos da mesma classe.[1]

Um problema que pode ocorrer ao utilizar esse classificador é o chamado *Overfitting*, que ocorre quando há um sobreajuste sobre dados de treinamento, isso quer dizer que o modelo se ajustou de forma muito específica aos dados de treinamento, tendo acurácia muito baixa para classificar novos dados. Isso pode ocorrer devido a erros ou ruídos do conjunto de treino, uma forma de evitar esse problema é utilizar o método de Poda da Árvore, que pode ser dividido em pré-podagem e pós-podagem. No método de pré-podagem a poda ocorre durante a criação da árvore, uma das formas é calcula-se o erro de cada nó de decisão e de seus ramos, se o erro do nó de decisão é menor ou igual ao somatório dos erros dos ramos então o nó de decisão é transformado em folha. Já no método de pós-podagem a poda ocorre depois da criação da árvore, removendo sub-árvores e as transformando em folhas da Árvore.

Cálculo da entropia dado um conjunto S, com instâncias de uma classe i e probabilidade p_i :

$$Entropia(S) = \sum -p_i log_2 p_{ieq}{}^{2} (1)$$

Cálculo do ganho(*Gain*):

$$Gain(S, X_i) = Entropia(s) - \sum_j \frac{|S_{xij}|}{|S|} Entropia(S_{xij}) eq^{3} \quad (2)$$

A figura 1 apresenta um exemplo de Árvore de Decisão onde há duas classes, joga e não-joga e os atributos clima, umidade e quantidade de chuva.

[1] Slides usados em aula disponibilizados pelo Professor
[3] Ver 1.

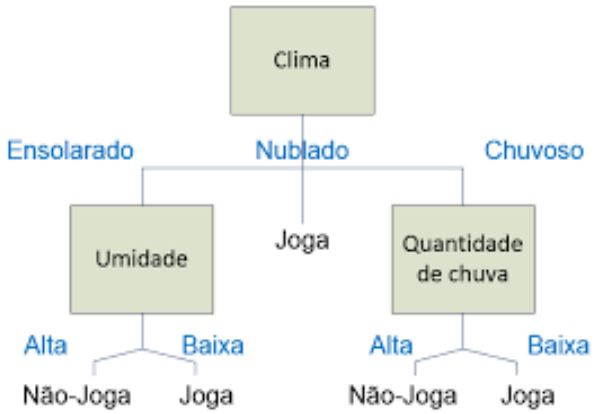


Figura 1. [4]
Exemplo de Árvore de Decisão.

III. Florestas Randômicas

Florestas Randômicas ou Aleatórias é um algoritmo de classificação ou predição, que consiste em criar um conjunto de Árvores de Decisão. Como foi dito anteriormente, Árvores de Decisão podem ser bem diferentes dependendo dos dados utilizados em seu treinamento e nos atributos escolhidos como raízes, isso faz com que possa ocorrer grande variância entre elas. Para diminuir este problema utiliza-se a técnica chamada de *bagging* que baseia-se em outra técnica chamada de *bootstrap*. Em *bootstrap* cria-se diversos subconjuntos de dados de forma aleatória e com reposição, do mesmo tamanho do conjunto original. *bagging* usa a ideia de *bootstrap* para gerar n diferentes subconjuntos de dados de treinamento, utilizar esses subconjuntos para o treinamento do algoritmo gerando assim n Árvores de Decisão, e para classificar certo dado pode-se usar a moda, ou seja, a predição que mais comum, ou utilizar a média das probabilidades e escolher a classe com maior probabilidade, dado que o algoritmo produza essas probabilidades.

No algoritmo de Florestas Randômicas, as árvores não são relacionadas, pois ao criar as árvores na fase de treinamento ao invés do algoritmo poder considerar todos os atributos e seus valores para escolher o melhor atributo para testar no nó de decisão, é especificado um número limite de atributos, e eles são escolhidos de forma aleatória. No caso de m ser esse número limite e p sendo os atributos totais, esse número geralmente é calculado com a fórmula (3).

$$m = \sqrt{p} \tag{3}$$

IV. Métodos de Validação

Dados modelos de aprendizagem supervisionada, onde há uma fase de treinamento, é importante haver métodos para validar como os modelos se comportam para dados diferentes, esses são chamados de Métodos de Validação.

[4]REVISTABW. Aprendizagem de Máquina: Árvores de Decisão. Disponível em :http://www.revistabw.com.br/revistabw/aprendizagem-arvore-de-decisao/

A. Método Hold-out

Nesse método os dados são divididos em dois subgrupos distintos chamados de treinamento e teste, utilizando-se dos dados de treinamento o algoritmo deve então prever os dados de teste.

B. Método Validação Cruzada

Esse método consiste em dividir os dados em k subconjuntos distintas e de mesmo tamanho, e os testes serão feitos k vezes, e cada vez um subconjunto é escolhido de forma aleatória como o de teste e os demais subconjuntos são utilizados como treino, considerando a média dos resultados. Um exemplo pode ser visto na fig.2, onde k = 4 e os retângulos vermelhos representam os subconjuntos de teste e os azuis os de treino.

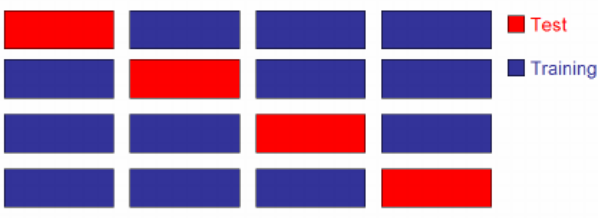


Figura 2. [5]Exemplo de Validação Cruzada.

V. Problema do Projeto

O problema do Projeto 2 é utilizar o algoritmo de classificação Florestas Randômicas para classificar espécies de plantas baseadas em suas folhas, fazendo a divisão entre dados de teste e treinamento utilizando o método de validação cruzada e analisar os resultados com relação ao desempenho e melhores valores, tendo uma taxa de acerto de no mínimo 50%.

Os dados utilizados podem ser encontrados aqui. São dadas 40 espécies de plantas, e para cada uma dessas espécies existem de 5 a 16 fotos de folhas dessa espécie. O algoritmo de Florestas Randômicas deve utilizar 14 atributos dessas folhas para classificá-las, sendo que esses atributos podem ser divididos em atributos de formato e de textura. Os atributos quanto à textura são: *Average Intensity, Average Contrast, Smoothness, Third moment, Uniformity* e *Entropy* , já os atributos quanto ao formato são:

Eccentricity,*Aspect Ratio*, *Elongation* e *Isoperimetric Factor* são atributos que medem se o formato da folha é mais arredondado ou alongado, variam de 0 a 1 e em *Eccentricity* e *Elongation* quanto mais próximo de 0 mais arredondada a folha enquanto que em *Aspect Ratio* valores próximos de 0 indicam uma folha mais alongada.

Solidity, *Stochastic Convexity*, *Maximal Indentation Depth*, e *Lobedness* são atributos que medem o quanto as folhas são convexas, sendo que para ser convexa, para qualquer par de pontos na imagem a reta entre eles também tem que

[5]Diagnosis of long QT syndrome via support vector machines classification. Disponível em: https://www.researchgate.net/figure/k-fold-cross-validation-scheme-example$_{fig2_228403467}$

estar contido na imagem. Esses atributos medem se a folha possui muito ou poucos lóbulos e se esses são mais ou menos acentuados. A fig.3 exemplifica de forma ilustrativa isso.

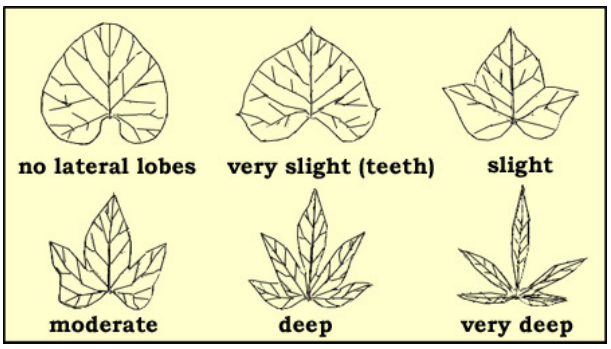


Figura 3. [6]Exemplo de folhas com e sem lóbulos(lobes em inglês).

VI. Implementação

Para a implementação do segundo projeto da disciplina foi utilizada a linguagem de programação *Python(3)*, devido a sua vasta coleção de bibliotecas referentes à aprendizagem de máquinas. A biblioteca *sklearn* foi novamente escolhida para a parte aplicada do projeto, devido a eficiência que teve no primeiro projeto e por apresentar funções específicas para o tema do projeto, Florestas Randômicas.

O script deve ser executado num terminal Linux com as seguintes bibliotecas instaladas: *pandas*, para leitura do arquivo csv; *sklearn*, para os cálculos referentes a Floresta Randômica; *matplotlyb*, para gerar a matriz de confusão em gráfico; e *sys*, para manter a execução do script de forma linear porém com mais controle.

Depois de importar todas as bibliotecas, o código começa lendo o arquivo .csv especificado no projeto, onde apresenta todas as espécies de folhas e seus atributos:

- Class
- Specimen Numer
- Eccentricity
- Aspect Ratio
- Elongation
- Solidity
- Stochastic Convexity
- Isoperimetric Factor
- Maximal Identation Depth
- Lobdness
- Average Intensity
- Average Contrast
- Smoothness
- Third moment
- Uniformity
- Entropy

[6]Huaman, Z. Systemic botany and morphology of the sweetpotato plant.Technical Information Bulletin 25. International Potato Centre, Lima, Peru. 22 p.

Então são criadas duas matrizes, uma para as classes e outra com todos os atributos, para que seja feita a comparação. A biblioteca *sklearn* então é utilizada para criar a árvore, onde então são utilizadas funções para prever o valor cruzado com k = 10, assim como o resultado do valor cruzado. Utilizando então essa previsão junto com a matriz de classes, é feito o cálculo da precisão do algoritmo da árvore randômica.

Para o cálculo das features mais importantes foi utilizada a biblioteca *sklearn* e sua função *feature_importances_*.

Por fim são feitos gráficos com a biblioteca *pandas* e *matplot* para melhor visualização dos resultados.

VII. Resultados

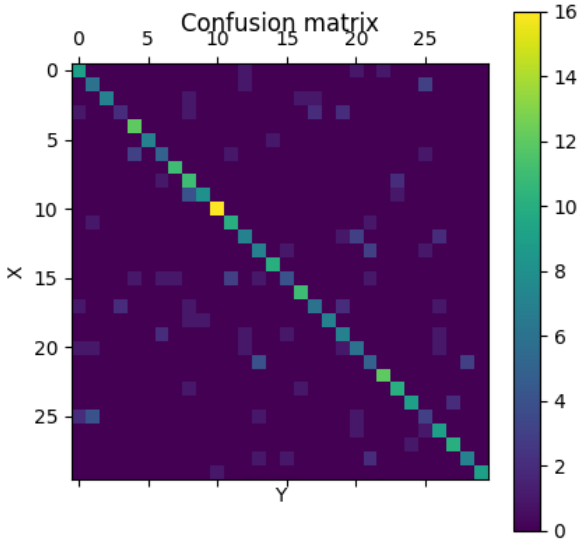
Utilizando o algoritmo de Florestas Randômicas com validação cruzada com k = 10 e escolhendo o número de árvores n como 50, descrito anteriormente para classificar espécies de plantas baseadas em suas folhas conseguimos superar a taxa de acerto médio mínimo de 50% obtendo em torno de 77,8% em nossos testes. Porém, testando outros dois valores para o número de árvores obtivemos os seguintes resultados. Com n = 25 e rodando o algoritmo 5 vezes, a média da taxa de acerto médio foi de de 75,4%, já com n = 10 essa média foi de 71,5%. Logo é possível observar que mesmo diminuindo o número de árvores pela metade a taxa de acerto teve um decréscimo muito menor, somente cerca de 3,5%, dessa forma o custo computacional pode ser reduzido de forma significativa mas a taxa de erro se manteve elevada. A matriz de confusão da Floresta Randômica com n = 25 pode ser visto no gráfico da fig 4, onde as diferentes cores representam o número de instâncias das classes.

Além disso, utilizando Florestas Randômicas é possível observar quais os atributos/features mais importantes, mais descriminatórios, essa observação na forma de um gráfico na fig 5. Analisando o gráfico é possível perceber que as 2 features mais importantes são a *Solidity* e *Eccentricity*, mas que nenhuma delas é insignificante o bastante a ponto de justificar sua remoção.

Referências

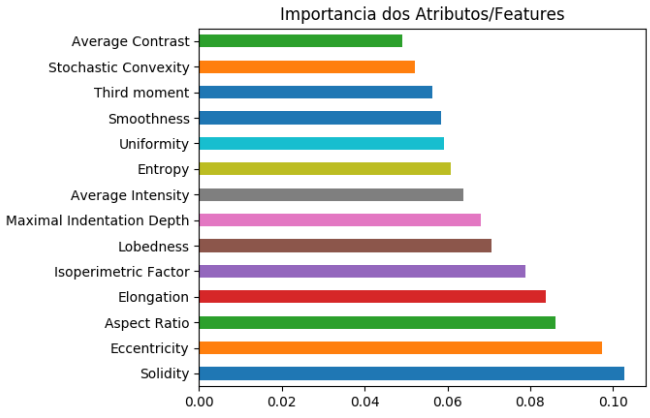
[1] Von Zuben, IA004–Profs Fernando J., and Romis RF Attux. "Árvores de Decisão."

[2] SILVA, LM. "Uma aplicação de Árvores de Decisão, Redes Neurais e KNN para a Identificação de Modelos ARMA não Sazonais e Sazonais."Disponível on-line em http://www2. dbd. pucrio. br/pergamum/tesesabertas/0024879_05_cap_03. pdf (2005).



H

Figura 4. Matriz de Confusão.



b

Figura 5. Features mais importantes.