O Uso da inteligência computacional para classificação do Dataset ”Wine Quality”

1st Gabriel Silva Oliveira
Centro de Informática
UFPE
gso@cin.ufpe.br

2nd João Marcos Alcântara Vanderley
Centro de Informática
UFPE
jmav@cin.ufpe.br

3rd Kennedy Edmilson Cunha Melo
Centro de Informática
UFPE
kecm@cin.ufpe.br

4th Rafael dos Reis de Labio
Centro de Informática
UFPE
Rrl3@cin.ufpe.br

***Abstract*—No aprendizado de máquina, o método de classificação refere-se a um procedimento que recebe uma determinada entrada de dados e designa para qual categoria certo dado pertence. Além disso, existem diferentes tipos de algoritmos de classificação, como os baseados em árvores, regras, dentre outros. São muito utilizados! Neste Projeto, procuraremos utilizar o modelo de classificação ingênua de bayes para fazer a classificação de notas de vinhos vermelhos a partir de seus atributos físicos e químicos.**

***Index Terms*—I.C, Dataset Wine Quality, classificação ingênua de bayes, ML**

I. Introdução

O aprendizado de máquina, como definido por Arthur Samuel é: *”O campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados”*. Por conta dessa característica e de alguns acontecimentos passados como as eleições presidenciais dos Estados Unidos com a presença da Cambridge Analytics em 2016 e do surgimento de grandes companhias de streaming de vídeos, filmes e jogos, o uso dessa tecnologia veio se tornando cada vez mais presente no dia a dia e, além disso, cada vez mais famoso. A aprendizagem automática, explora o estudo e a construção de algoritmos que podem aprender de seus erros e fazer previsões sobre dados. Esses algoritmos operam construindo modelos a partir de amostras a fim de realizar previsões ou decisões guiadas pelos dados ao invés de só seguir instruções programadas. Algumas partes do aprendizado de máquina estão ligadas à estatística computacional, complexidade computacional e otimização matemática, a qual produz métodos, teoria e domínios de aplicação para o campo da estatística computacional. A aprendizagem automática possui diversas aplicações e uma delas é a detecção de spam a partir do uso do classificador Ingênuo de Bayes. O método de classificação ingênua de bayes é muito utilizado na área de machine learning e esse algoritmo foi baseado no Teorema de Bayes, um teorema probabilístico que descreve a probabilidade de um evento, baseando-se em um conhecimento a priori que pode estar relacionado ao evento. O algoritmo recebe esse nome ”ingênuo” (naive) porque desconsidera a correlação entre as variáveis(features), ou seja, se determinada fruta é rotulada como ”Limão”, caso ela também seja descrita como “Verde” e “Redonda”, o algoritmo não vai levar em consideração a correlação entre esses fatores. Isso se dá porque o algoritmo trata cada um de forma independente. Além da aplicação para determinar se uma mensagem é spam ou não, o método de bayes é frequentemente aplicado em processamento de linguagem natural e diagnósticos médicos, o método pode ser usado quando os atributos que descrevem as instâncias forem condicionalmente independentes. Ou seja, o teorema de Bayes trata sobre probabilidade condicional. Isto é, qual a probabilidade de o evento A ocorrer, dado o evento B. No caso do projeto que pretendemos realizar, aplicamos o algoritmo classificador de Bayes para determinar a qualidade de um Vinho, a partir do dataset que temos e das características do Vinho que são passadas a ele. Por fim, o documento encontra-se da seguinte forma: a Seção II aborda os objetivos do projeto proposto, Seção III descreve a justificativa para a abordagem adotada, a Seção IV demonstra a metodologia escolhida, a Seção V cita o cronograma do projeto e, por fim, segue as referências e a explicação de cada link.

II. Cronograma de atividades

Pretendemos realizar a seguinte sequência de passos para a entrega do projeto do algoritmo Ingênuo de Bayes e a continuação do papper com os outros tópicos necessários:

1) Carregar a base de dados em um Notebook python.
2) Análise exploratória dos dados.
3) Inferir novas distribuições dos dados de acordo com o classificador de Bayes.
4) Plotar gráficos.
5) Divisão entre treino e teste.
6) Treinamento.

7) Validação.
8) Conclusões.

III. Objetivos

O objetivo do nosso projeto é treinar um modelo de classificação, no caso utilizaremos o algoritmo de classificação Ingênua de Bayes. O treino do nosso modelo tem como intuito, definir a qualidade do Vinho que está sendo analisado(numa escala de 0 a 10), que é dividida em dois tipos: Red(tinto) e White(Branco). Isso será feito a partir de dados específicos, como por exemplo a densidade do Vinho analisado, seu PH e algumas outras séries de características Físico-Químicas que serão utilizadas durante a análise.

IV. Justificativa

A escolha deste dataset se deu por conta da quantidade de atributos e instancias presentes nele,visto que, quanto maior a quantidade de ambos, maior será a quantidade de dados e de classes que poderemos ultilizar para treinar nosso algoritmo e classificar os tipos de vinho.

Além disso, o dataset se enquandra muito bem para a task de classificação, tanto por conta da quantidade de informação sobre os vinhos, quanto a quantidade de classes, que irá ajudar nos metodos de classificação, no qual temos de separar os atributos independentes da base de dados

V. Base de Dados

A partir de uma analise inicial do dataset e algumas consultas feitas atraves de funções presentes nas bibliotecas do python:

- As funções read.csv(img1),head(img1) Realizam a transformação do dataset para csv e o primeiro contato com o dataset.
- Shape,types,nestas funções são feitas as verificações do dataset.Em que é visto a quantidade de instancias,atributos e os tipos de dados de ambos(img2).
- Describe,transpose, são funções que auxiliam na análise da quantidade, média, mínimo e máximo dos valores de cada atributo.(img3)
- Info, Verifica a presença de um dado faltante no dataframe.(img4)
- Isnull().sum(), verifica se existe algum dado nulo no dataset e caso exista,soma o valor na tabela.(img5)
- df[i].unique(), Verifica a quantidade de valores distintos em cada atributo, por meio de um loop.(img6)
  Todas essas funções vieram da biblioteca pandas.

VI. Análise Exploratória dos dados

As bibliotecas que ultilizamos para realizar as análises exploratórias dos dados foram: seaborn,pandas,numpy,matplotlib.pyplot e plotly.express.

- Ultilizamos a Seaborn,matplot.pyplot e plotly.express para visualizar as classes e atributos do dataframe.Fizemos isso atravez do plot de cada classe e sua distribuição
- Ultilizamos o plotly.express para gerar graficos que correlacionavam os atributos : volatile acidity, alcohol, sulphates. Além disso, ultilizamos uma propriedade do grafico para ver a nota do atributo quality em cada um dos graficos com os outros atributos.
- Outra analise feita ultilizando o plotly.express foi ultilizando um grafico dinamico, no qual relacionava o pH com as frequencias dos seus valores. Dentro de algumas celulas,mostra as frequêcias dos valores do alcohol para o determinado pH selecionado. Selecionando algum dos valores do alcohol, mostra as frequências dos valores do citric acid do pH e alcohol selecionado.
- A partir da analise de dados,conseguimos chegar a algumas conclusões,como os valores da qualidade que eram de 5 e 6 são extremamente maiores em quantidade do que os outros, dessa forma optamos por excluir todas as outras linhas que são diferentes de 5 e 6, pois elas são outliers que só iriam atrapalhar o desenvolvimento do modelo.
- Comprovamos a tese a com o uso do pandas e de 2 laços e em seguida, retiramos os index's que possuiam valores de qualidade diferente de 5 e 6. Para que logo em seguida deletassemos do dataframe, esses index's.
- Por fim, Plotamos o mapa de calor das correlações entre os atributos e as classes ,ultilizando as bibs pandas,numpy e seaborn. Podemos perceber que os sulfatos e a quantidade de álcool são os valores com maior correlação direta e a volatilidade do ácido possui a maior correlação inversa.

VII. Classificador Ingênuo de Bayes

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

Ultilizamos uma das implementações do Naive-Bayes da biblioteca scikit-learn, o GaussianNB, Que é a implementação da distribuição gaussiana aplicada ao algoritmo do naive bayes. Ultilizamos a distribuição gaussiana por termos atributos com valores continuos.

Os parametros ultilizados foram gerados a partir da análise exploratória dos dados, a partir do uso das bibliotecas pandas e sklearn.preprocessing

Ultilizamos a bib do preprocessing para realizar a normalização dos dados a partir da função StandardScaler.

Antes de se dar início aos testes, é realizado o treinamento para gerar uma tabela de probabilidade, que é feita considerando os eventos independentes. Após a tabela de probabilidade gerada, é dado início aos testes.

Ao se comparar o treino com o test, recebemos uma variancia baixa temos um bom sinal, por tanto, não tivemos nem um overfeating e nem um underfeating

VIII. Experimentos

O experimento foi realizado em duas fases:

1) Treinamento do algoritmo do naive bayes: Após a analise exploratória de dados, conseguimos limpar o dataframe para evitar outliers, que iriam atrapalhar o treinamento. Após a limpeza, realizou-se a separação das

linhas de treinamento e de teste com os seus respectivos previsores(X's) e classes (Y's), além disso, ultilizamos a biblioteca pickle para guardar as variaveis em um arquivo. Em seguida,foi realizada a normalização dos dados numéricos, a fim de que o algorítmo não dê preferência a valores maiores (com distinções bastante relevante). O cálculo é feito da seguinte forma. x = (x - media(x)) / desvio-padrão(x). Em seguida colocamos o algoritmo para treinar e gerar a tabela de relação de classes e preditores.

2) Aplicação do algoritmo e a tabela gerada por ele:Após termos o algoritmo treinado, nós realizamos a previsão a partir da tabela gerada pelo treino dele. Em seguida,realizamos a checagem da acurácia para verificar o rendimento dele sobre os casos testes.Após isso, Fazemos a previsão e a acurácia para ver o rendimento do algorítmo com os casos de treinamento, para compararmos o do treino com teste a fim de vermos se ocorreu overfitting.

IX. Análise dos resultados

Os resultados que foram obtidos foram bons, tivemos uma variancia de 0.0084 entre o modelo de treino e o modelo de teste e a acuracia foi satisfatória, ficou entre 70% de acerto e tivemos uma variancia de 0.0084 entre o modelo de treino e o modelo de teste. esse resultado mostrou que não ocorreu um overfitting, no algoritmo. Acreditamos também que se as notas predominantes tivessem valores mais distantes (exemplo 2 e 9) a acurácia do modelo poderia ser melhor, visto que quanto mais diferentes as notas seus atributos tendem a ser mais diferentes. Dessa forma por 5 e 6 mesmo sendo notas próximas ainda assim termos conseguido uma acurácia de 70%, consideramos um suceesso.

Alem disso, para poder comprovar essa acuracia e verificar alguns pontos, ultilizamos a matriz de confusão para auxiliar nesse estudo. A matriz de confusão é uma tabela com duas linhas e duas colunas que relata o número de falsos positivos,falsos negativos,verdadeiros positivos e verdadeiros negativos. Isso permite uma análise mais detalhada do que a somente uma proporção de classificações corretas,ou seja a precisão. A precisão produzirá resultados enganosos se o conjunto de dados estiver desequilibrado. Que seria quando o número de observações em diferentes classes variam muito do seu grafico. A partir dela e do seu grafico, conseguimo definir os resultados e essas analises vistas no Experimento[VIII].

X. Metodologia

Como dito no objetivo desse projeto, o classificador de Bayes será implementado para fazer as classificações. Em resumo, inicialmente será feita a análise exploratória e tratamento dos dados, plotando seu histograma para identificarmos a distribuição dos dados, além de outros recursos de visualização de dados. Após isso, iremos dividir nossa base de dados entre treino e validação (80 % e 20 %, respectivamente), então iremos utilizar normalizar os dados e usar o classificador de bayes da biblioteca sklearn para realizar o treinamento dos dados de treino, e após isso iremos comparar com o resultado dos dados de validação para verificarmos a acurácia do modelo em dados que nunca foram vistos anteriormente. Porém conseguimos dividir todos esses métodos em alguns blocos específicos, para ficar mais organizado e didático.

A. Itens/Ferramentas

Utilizar o Colab para compilar e executar os códigos criados na linguagem Python para aplicar os modelos utilizados para a realização desse projeto. As bibliotecas utilizadas serão Numpy, Sklearn, Pickle, Pandas, Matplotlib.pyplo e Seaborn, justamente por serem mais práticas e apresentarem uma visão computacional interessante para tratar os dados dos modelos. Ademais, o ambiente de trabalho será Google Colab ou Jupyter Notebook.

B. Implementação

Utilizar as técnicas a partir dos dados do Vinho recolhido da UCI Machine Learning Repository. Esses dados possuem uma variedade de informações e estatísticas peculiares para cada Vinho, apresentando características como Acidez, Porcentagem de Álcool, Volatilidade Ácida e outros diversos atributos para que se consiga determinar a qualidade do Vinho analisado em questão. Portanto, a implementação fica da seguinte forma

- O conjunto de dados do Vinho é utilizado como entrada(que vai conter as informações peculiares e específicas para o Vinho em análise).
- Converter todos os dados de string para número, pois iremos receber da base de dados as informações todas em texto, e faremos associações de cada string de uma classe para seu respectivo inteiro, pois quando formos fazer consultas, iremos ter que passar como entrada como inteiro para pesquisa.
- Em seguida, faremos a divisão dos dados que serão utilizados como treino e testamento, que nesse caso será de 80 por cento e 20 por cento, respectivamente.
- Após isso, distinguir o conjunto de dados de treinamento com base nos valores da classe, ou seja, 1, 2 e 3.
- Com isso, determinar o desvio padrão e os valores médios para o caso de dados individuais com base no valores de classe.
- Em seguida, escolher o parâmetro ingênuo Bayes como entrada para a otimização de pesquisa da grade do algoritmo.
- Aplicar o valor ideal do parâmetro como um valor inicial para o processo de classificação usando Bayes ingênuo.
- Utilizar o modelo e gerar previsões, para que consigamos ter noção.
- Por último, achar a precisão da previsão por meio da comparação dos dados de classe do conjunto de dados de teste. Essa precisão é avaliada com base na proporção entre 0 e 100

C. Implementação do código metódica:

- Importar as bibliotecas
- Importar a base de dados

- Dividir a base de dados em conjunto de treino e conjunto de teste
- Dimensionamento de recursos
- Treinar o classificador do modelo de Bayes no conjunto de treinamento
- Prever os resultados do conjunto de testes
- Analisar a precisão do modelo e plotar a matriz de confusão
- Comparar os valores reais com o valores preditos

XI. Conclusão

Os resultados que foram obtidos foram bons, tivemos uma variancia de 0.0084 entre o modelo de treino e o modelo de teste e a acuracia foi satisfatória, ficou entre 70% de acerto. esse resultado mostrou que não ocorreu um overfitting, no algoritmo.

References

[1] [Matriz de erro] https://towardsdatascience.com/understanding-the-confusion-matrix-and-how-to-implement-it-in-python-319202e0fe4d
[2] https://www.youtube.com/watch?v=WqMnQuC19Rg&list=PLZ3V9XyVA529XBeS6ew6SRHAAaWTIml0t
[3] https://www.youtube.com/watch?v=WqMnQuC19Rg&list=PLZ3V9XyVA529XBeS6ew6SRHAAaWTIml0t
[4] https://scikit-learn.org/
[5] https://pandas.pydata.org/
[6] https://numpy.org/
[7] https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html
[8] https://www.plotly.express/
[9] https://archive.ics.uci.edu/ml/datasets/wine+quality