Aprendizagem de Máquina utilizando WEKA

Douglas Dallavale e Vinícius Vianna

PUCRS – Pontifícia Universidade Católica

douglas.dallavale@acad.pucrs.br, vinicius.vianna@acad.pucrs.br

**Abstract.** In this article we will demonstrate techniques of text classification in a system of questions and answers with the thematic of the Random Forest algorithm. We will also show you the pre-processing, classification, analysis and results of this process.

**Resumo.** Neste artigo demonstraremos técnicas de classificação de texto em um sistema de perguntas e respostas com a temática do algoritmo Random Forest. Nós também mostraremos as fazes de pré-processamento, classificação, análise e os resultados obtidos.

# 1. Introdução

No presente trabalho vamos primeiramente explicar o que exatamente é o algoritmo Random Forest, que será o tema de nosso corpus(o dataset que contém todos os dados das perguntas e respostas que criamos), e como o corpus foi criado. O algoritmo Random Forest é um algoritmo de aprendizagem de máquina que pode produzir excelentes resultados na maioria das vezes. Ele também é um dos algoritmos mais utilizados, devido à sua simplicidade e o fato de que pode ser utilizado para tarefas de classificação e também de regressão.

Já o corpus, foi criado com a turma de Inteligencia Artificial com mais de 1000 perguntas e respostas, para que tivéssemos um bom dataset para ser treinado. Estas perguntas e respostas também receberam classificações quanto a sua temática, por exemplo, se a pergunta falava sobre a história do algoritmo ou sobre suas funcionalidades, e assim por diante.

Nós iremos utilizar a biblioteca do CoGroo em python para o pré-processamento do corpus. E, por fim, devemos acrescentar que para a tarefa de aprendizado de máquina será utilizado a biblioteca SkLearn disponível em Python. Esta biblioteca possuí diferentes métodos para que possamos realizar um treinamento adequado em cima do corpus.

# 2. Fundamentação Teórica

A execução deste trabalho consistiu em três grandes etapas: pré-processamento dos dados, categorização, e a análise dos resultados obtidos. A seguir iremos explicar cada uma delas em suas teorias, e em seguida, mostraremos como elas foram implementadas na próxima seção.

A primeira etapa, o pré-processamento, consistiu basicamente na criação do corpus, com decisões importantes como a definição das classes de perguntas que seriam utilizadas e a criação de perguntas e respostas variádas para uma melhor aprendizagem. Também foi nesta etapa que os dados foram normalizados, ou seja, nós tranformamos todas as palavras em seus lemas, o que significa que os verbo, substantivos e demais, voltaram as suas formas não conjugadas. Nós também realizamos uma anotação linguistica nas palavras já normalizadas, o que consistiu em classificar estas palavras em suas devidas classes gramaticais, por exemplo, verbo, substantivo, adjetivo, etc. Porém no fim, nós não utilizamos destas anotações linguisticas.

Em seguida, nós extraímos os termos mais relevantes de cada uma destas perguntas e os armazenamos em um novo arquivo .csv, ignorando artigos, preposições, conjunções, entre outros. Após a criação deste novo arquivo, nós escolhemos os K(ele será explicado na próxima seção) termos mais relevantes, ou seja, os que tinham mais ocorrências em todo o corpus e os agrupamos em um conjunto. E, por fim, realizamos a estruturação dos dados que passaram até a etapa anterior, para que ficassem em um formato que pudessemos treinar o modelo.

Na segunda etapa, a categorização das perguntas por meio do SkLearn, foi feita com base no pré-processamento realizado com o corpus. Nós escolhemos esta biblioteca por sua faclidade de entendimento e utilização.

Por fim, a última etapa, de análise de resultados, que é o principal foco deste artígo, será feita com base nos resultados obtidos pelo treinamento com o Sklearn e será melhor explicado na seção 4.

# 3. Codificação

A codificação para este trabalho ocorreu principalmente na parte de pré-processamento, que nescessita de uma maior atenção pois são com os dados analisados e extraídos dele que iremos realizar o aprendizado de máquina com o SkLeran. Para realizar a normalização morfológica e anotação linguistica, explicadas anteriormente, nós utilizamos da biblioteca do COGROO, que está disponível em python, com uma simples instalação. Porém antes de decidirmos utiliza-lo, nós tentamos o TreeTagger, no entanto, ele foi difícil de entender e não parecia funcionar direito, então por isso optamos pelo Cogroo.

Em Python, a implementação foi bem simples, apenas utilizamos os métodos de normalização morfológica e anotação linguistica disponíveis no cogroo e trabalhamos com matrizes em python, para separar e armazenar os dados relevantes das perguntas e então colocá-los em um arquivo .csv para que fosse lido pelo Sklearn. Por fim, nós separamos todos os dados, dividindo-os em dois arquivos arff, um contendo 80% de todo o dataset selecionado, que foi utilizado para o treinamento, e o outro contendo 20%, sendo utilizado para o teste ou validação do treinamento realizado.

Após a separação destes dados, nós utilizamos o modelo LinearSVC, disponibilizado pelo SkLearn, que aparentou ser melhor que o modelo RandomForest, também testado por nós. E então treinamos e testamos o modelo com base nos dados do corpus.

# 4. Resultados Obtidos

Abaixo, temos os resultados obtidos por meio do classificador do LinearSVC. Se analisarmos os resultados obtidos, veremos que obtivemos uma boa precisão com o treinamente e a classificação dos modelos. A precisão, que esta localizada na primeira coluna, foi obtida com base nos dados que correspondem a 20% do modelo, ou seja, os dados de teste. Para os resultados que foram abaixo de 0.4, podemos chegar a algumas conclusões: o modelo pode não ter sido treinado com aquele tipo de classe suficientemente ou o modelo pode não ter casos de teste suficientes ou aceitáveis para esta classe.

precision recall f1-score support

aplicacao 0.50 0.25 0.33 4

funcionamento 1.00 0.50 0.67 2

conceito 0.00 0.00 0.00 2

historia 1.00 1.00 1.00 2

definicao 1.00 1.00 1.00 2

nEstimators 1.00 1.00 1.00 2

comparacao 1.00 1.00 1.00 1

hiperparametro 1.00 1.00 1.00 1

funcionamento 0.38 1.00 0.55 3

desvantagem 0.50 1.00 0.67 2

usabilidade 0.00 0.00 0.00 1

hiperparametroVelocidade 0.75 0.92 0.83 13

história 0.80 0.67 0.73 6

paradigma 0.77 0.71 0.74 28

vantagem 0.00 0.00 0.00 1

hiperparametroPreditivo 0.00 0.00 0.00 0

comparação 0.00 0.00 0.00 3

metodos 0.56 0.64 0.60 22

bagging 0.50 1.00 0.67 5

sobreajuste 0.81 0.93 0.87 42

parametro 0.00 0.00 0.00 2

maxFeatures 0.00 0.00 0.00 0

diferença 0.50 1.00 0.67 1

aplicação 0.00 0.00 0.00 0

metodo 0.80 0.40 0.53 10

minSampleLeaf 0.00 0.00 0.00 5

extra trees 1.00 0.25 0.40 4

hiperparametros 0.00 0.00 0.00 0

overfitting 0.67 0.75 0.71 8

proposito 1.00 1.00 1.00 1

melhoria 0.80 1.00 0.89 4

tratamentoDeDados 1.00 1.00 1.00 3

baggedForest 1.00 1.00 1.00 3

oobScore 1.00 1.00 1.00 2

conceitoBootstrap 1.00 0.67 0.80 3

conceitoArvoreDecisao 0.67 1.00 0.80 4

caracteristica 0.50 1.00 0.67 2

desbalanceamento 0.33 1.00 0.50 1

conceitoHiperparametro 1.00 1.00 1.00 1

nJobs 1.00 1.00 1.00 1

classificacao 0.00 0.00 0.00 1

pré-processamento 0.00 0.00 0.00 1

nEstimators 1.00 1.00 1.00 1

conceitoArvore 0.00 0.00 0.00 2

tarefa 0.67 0.33 0.44 6

overfit 1.00 1.00 1.00 1

pós-processamento 1.00 1.00 1.00 4

randomState 1.00 1.00 1.00 1

maxDepth 1.00 1.00 1.00 1

implementacao 0.67 0.67 0.67 3

limitação 1.00 0.14 0.25 7

conceitoBoosting 0.00 0.00 0.00 1

treinamento 1.00 0.50 0.67 2

definição 1.00 1.00 1.00 2

conceitoBagging 1.00 1.00 1.00 1

desvantagens 0.00 0.00 0.00 0

vantagens 0.00 0.00 0.00 2

Classe 0.00 0.00 0.00 2

micro avg 0.72 0.69 0.70 235

macro avg 0.59 0.59 0.56 235

weighted avg 0.70 0.69 0.67 235

# 5. Conclusão

Durante o decorrer deste trabalho nos deparamos com alguns contra-tempos, como o fato de entender melhor como um modelo deve ser treinado por meio do SkLearn e de como interpretar seus resultados, bem como o fato de escolher qual o analisador deveríamos utilizar, pois nem todos eles são fáceis e intuitivos de serem utilizados. No fim, conseguimos chegar a resultados satisfatórios com as escolhas que fizemos durante o trabalho, e agora temos um melhor entendimento sobre aprendizagem de máquina e a importância de termos um bom pré-processamento dos dados e de saber interpretar os resultados que nos são informados pelo aprendizado.