**Nanodegree Engenheiro de Machine Learning**

**Proposta de projeto final**

Yuri Martins Campolongo

20 de março de 2018

**Proposta**

**Histórico do assunto**

Uma das grandes utilizações de machine learning nos últimos tempos, e que se mostra cada vez mais promissora é a interpretação de linguagem natural. Essa técnica se tornou muito útil e procurada nos últimos tempos devido ao grande aumento do uso de chatbots, que são robôs de atendimento que interpretam textos digitados pelo cliente, de maneira natural, para realizar a busca de possíveis soluções para aquele problema.

De acordo com grandes influenciadores do mercado de tecnologia como Mark Zuckerberg (Facebook) o autoatendimento trará mais eficácia das interações entre usuários e empresas, diminuindo o tempo de resposta e auxiliando clientes na resolução de problemas, essa definição foi dada por ele em 2016 durante um evento para desenvolvedores.

A motivação da criação de um algoritmo para a análise de linguagem natural surgiu pois atualmente trabalho em uma companhia que cria sistemas de atendimento para empresas, e após análise da área comercial, verifiquei que as dúvidas dos possíveis novos clientes que entram em contato via chat, podem ser classificadas e posteriormente ensinadas para um algoritmo de machine learning, que pode também utilizar essas classificações para exibir a melhor resposta, sem a necessidade de uma pessoa para responder dúvidas frequentes e recorrentes.

**Descrição do problema**

Será criado um dataset com uma amostra das dúvidas enviadas pelos clientes no chat, essas dúvidas são referentes a produtos vendidos pela companhia que trabalho portanto não são confidenciais. Cada dúvida terá uma classificação de um tipo exemplo: Dúvida sobre produto, dúvida sobre forma de pagamento, dúvida sobre personalização, etc. E será utilizado um algoritmo de classificação para aprender esse dataset e fornecer as previsões necessárias.

**Conjunto de dados e entradas**

Os dados de entrada serão obtidos de um arquivo .csv com duas colunas. Na primeira coluna existirá o texto da dúvida digitada pelo cliente e na segunda a classificação do tipo dessa dúvida.

Com esse arquivo, será possível realizar um tratamento e posteriormente utilizar essas informações em um algoritmo de classificação.

**Descrição da solução**

Será aplicado um tratamento nos textos da primeira coluna do dataset, afim de remover caracteres especiais, palavras sem significância e tratamento de palavras com flexão de gênero e número. Como um computador não trabalha bem com palavras, será aplicado um algoritmo de ‘Saco de palavras’ para indicar quantas vezes aquela palavra ocorre na frase, e isso será realizado no dataset inteiro, sendo cada posição do array a representação de uma palavra específica. Após isso, esses dados traduzidos em números será enviado como entrada para o algoritmo de aprendizagem

**Modelo de referência (Benchmark)**

Os dados serão comparados com as respostas de um modelo de regressão logística, para comparar se a utilização de algoritmos bayesianos realmente são a melhor forma de tratar esse problema. A métrica para avaliação de acuracidade será utilizando o f1 score: isso nos dará uma porcentagem de acuracidade que indica se o algoritmo está conseguindo prever corretamente as corretas respostas para as dúvidas dos clientes.

**Modelo de avaliação**

A forma de avaliação de acuracidade, conforme citada no tópico anterior, será a comparação com as conversas de pessoas reais, indicando se o algoritmo foi capaz de prever corretamente o tipo de dúvida do cliente.

**Design do projeto**

O projeto seguirá a seguinte estratégia:

* Os textos serão tratados para remoção de palavras não importantes;
* O texto limpo será traduzido para a técnica de ‘Bag of words’
* A classificação de cada texto no dataset já foi previamente feita;
* Será utilizado um algoritmo de classificação para a aprendizagem do dataset;
* O dataset será separado em dataset de treinamento e dataset de testes;
* A acuracidade será calculada pelo algoritmo f1 score;

- A importância do tratamento prévio do texto em algoritmos de linguagem natural é fundamental, visto que esse pré-processamento aumenta o score final do algoritmo, já que com menos palavras desnecessárias a capacidade de previsão do algoritmo será incrementada, pois podemos considerar essas palavras menos importantes como ‘outliers’ que poderiam atrapalhar e causar previsões incorretas. Esse processo será feito com técnicas de: remoção de pontuação, remoção de palavras de ligação, artigos e verbos não importante, alteração para letras maiúsculas apenas.

- Como um algoritmo de machine learning depende de entradas numéricas e não textuais, os textos após passarem pelo pré-processamento serão subemetidos a uma técnica chamada de ‘Bag of words’, que nada mais é do que iniciar um vetor em que cada posição representa uma palavra do dataset geral, e para cada texto de entrada, será contada quantas vezes uma determinada palavra ocorre naquele texto e esse numero será adicionado na posição que aquela palavra é representada no vetor, exemplo:

Textos: O curso de machine learning da Udacity é muito bom!

Processamento de linguagem natural é muito interessante.

Textos após tratamento: CURSO MACHINE LEARNING UDACITY MUITO BOM

PROCESSAMENTO LINGUAGEM NATUAL MUITO INTERESSANTE

O vetor do bag of words conterá 10 posições, pois temos ao total 10 palavras diferentes no exemplo, já que a palavra MUITO se repete 2x, assim sendo, segue a tabela de como ficará os vetores para os dois exemplos acima:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CURSO | MACHINE | LEARNING | UDACITY | MUITO | BOM | PROCESSAMENTO | LINGUAGEM | NATURAL | INTERESSANTE |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

Com o exemplo acima é possível ver que com essa técnica é possível utilizar um algoritmo de classificação que espera entradas numéricas.

- Avaliarei algoritmos de classificação do tipo Bayesianos, como: Naive Bayese, Gaussian Naive Bayes e Multinominal Naive Bayes. Escolhi esse tipo de algoritmo pois eles são amplamente utilizados para esse tipo de técnica, e tem a vantagem de serem rápidos, fáceis de treinar e tem uma boa performance. Alguns artigos disponíveis online mostram formas de como utilizá-los para o propósito em questão:

<https://syncedreview.com/2017/07/17/applying-multinomial-naive-bayes-to-nlp-problems-a-practical-explanation/>

<https://medium.com/@theflyingmantis/text-classification-in-nlp-naive-bayes-a606bf419f8c>

- Para avaliar qual tipo de algoritmo se sairá melhor, será utilizado o F1-score da biblioteca sklearn <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html> , dessa forma é possível obter a qualidade de cada algoritmo e sua precisão no dataset de testes;