Machine Learning Engineer Nanodegree

Twitt Detector Celebrities

Bruno Santos

Março, 18, 2018

**Definição**

1. **Project Overview**

Com um mundo cada vez mais conectado, as pessoas utilizam de mídias sociais para expressar suas opiniões e sentimentos. Muitas pessoas utilizam as mídias como fonte de notícias no dia a dia, um dos meios mais populares para isso é o Twitter.

Nos últimos meses vimos como alguns ‘twitts’ quase provocaram guerras, já outros twitts servem para comunicar o andamento do projeto de levar o homem para Marte. Sendo assim, é possível identificar de quem é aquele ‘twitt’ sem nem mesmo saber quem o escreveu?

Sendo assim foram escolhidas 4 celebridades do twitter que se encontram no top 100 com mais seguidores:

* Barack Obama: ex-presidente dos EUA (3° mais seguido do twitter)
* Donald J. Trump: presidente dos EUA (20° mais seguido do twitter)
* Bill Gates: Fundador da Microsoft e filantrópico (22° mais seguido do twitter)
* Ellon Musk: Fundador do Paypal, SpaceX e Tesla ( 92° mais seguido do twitter)

1. **Problem Statement**

O objetivo desse projeto é criar um reconhecimento de texto para identificação de quem é seu autor, o universo foi reduzido a 4 personalidades, porem pode ser aplicado em larga escala utilizando a mesma técnica mudando apenas o data set. Também para rápida consulta da predição, será criado um endpoint para consumo dessa predição. Para isso acontecer serão seguidos os seguintes passos:

* Consulta a API do twitter buscando twitts dos usuários pré-definidos.
* Tratamento e separação dos campos uteis para esse estudo.
* Preparação do dataset de dados.
* Utilização de mecanismos de varredura de texto como split words.
* Vetorização dos textos e preparação da base de treino e teste.
* Resultado em diferentes classificadores.
* Coleta dos resultados e escolha do melhor classificador.
* Implantação de enpoint para consulta da predição em formato json.

1. **Metrics**

Para cálculo das métricas será utilizada uma matriz de confusão para garantir que os dados de teste estão corretamente sendo realizados.

Como a análise de texto em um âmbito aberto não é garantia de nada, não será esperado um índice de acerto altíssimo. O algoritimo que possuir a média de 80% de acertos será considerado aprovado para esse estudo.

Também serão utilizados dados reais de twitts mais atuais das celebridades escolhidas para esse projeto.

**Analysis**

1. **Data Exploration**

Toda a base de dados foi extraído da API do twitter, que disponibiliza dados completos dos últimos 200 twitts do usuário escolhido. Para realização dessa etapa foi necessário criar uma API no twitter (<https://developer.twitter.com/en/docs>) realizar um HTTP Post para obter um token de acesso, e só assim consultar os dados dos usuários escolhidos (<https://api.twitter.com/1.1/statuses/user_timeline.json?screen_name=elonmusk&count=200&tweet_mode=extended>) e assim obter a resposta como exemplo abaixo:

{

"created\_at": "Fri Mar 09 22:53:25 +0000 2018",

"id": 972243992153739265,

"id\_str": "972243992153739265",

"full\_text": "Boring Co urban loop system would have 1000’s of small stations the size of a single parking space that take you very close to your destination &amp; blend seamlessly into the fabric of a city, rather than a small number of big stations like a subway",

"truncated": false,

"display\_text\_range": [

0,

250

],

"entities": {

"hashtags": [],

"symbols": [],

"user\_mentions": [],

"urls": []

}

Todos os dados recolhidos estão anexados nesse trabalho.

Para parse do recebimento do texto foi utilizado um método criado em .NET que realiza a leitura do json e extrai somente o texto e a data que foi publicado, além de realizar limpezas básicas no campo do texto. Esse método está em anexo a esse trabalho.

Sendo assim já temos em um arquivo único o autor, texto e data de cada twitt das celebridades escolhidas separadas por virgula em formato csv.

Com esses dados já é possível começar a criação do algoritmo de machine learning para

1. **Exploratory Visualization**

Como já foi definido que cada celebridade possui 200 twitts, não existe muita comparação quanto aos textos nesse ponto, porem foi feita uma análise de como esses 200 twiits estão divididos em uma linha de tempo, que revela a frequência com que cada celebridade costuma publicar na rede social.

TODO INCLUIR AQUI IMAGEM COM DATA TWITT

1. Algorithms and Techniques

Para análise de texto é de boa pratica primeiramente utilizar a limpeza de texto chamado de “stop words” que são palavras auxiliares, diferentes em cada língua. Em português “para, que, até” são consideradas stop words, e assim removidas do texto original para não causar erros de analise de palavras que são comuns para qualquer universo.

Com isso foi utulizado uma biblioteca em python chamada de “nltk.corpu”, que realiza a manutenção de “stop words” em diversas línguas.

Para realização de gráficos rápidos foi gerada uma classe auxiliar para criação da matriz de confusão. Com isso rapidamente após o treino de um classificador, era possível visualmente saber se estava no caminho correto.

Foi utilizado a página do scikit-learn (<http://scikit-learn.org/stable/tutorial/machine_learning_map/>) que auxilia na escolha do melhor classificador para cada caso, com isso foram analisados 4 tipos.

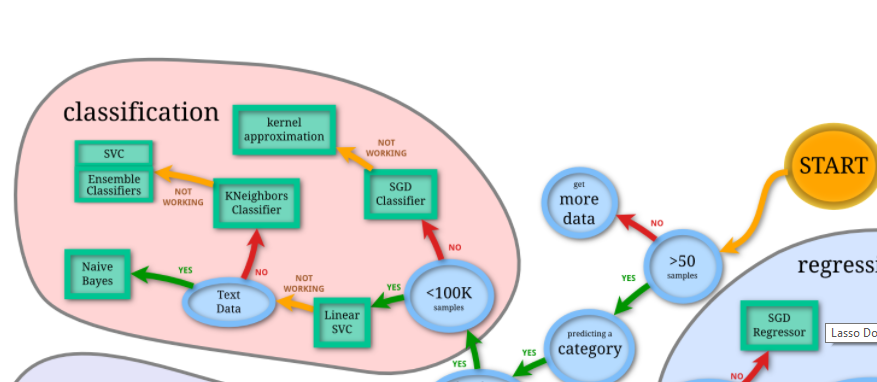


Imagem 1 : Scikit-learn :: Choosing the right estimator

Sendo assim foram analisados os métodos para realização da aprendizagem supervisionada:

* LogisticRegression
* GaussianNB
* NearestCentroid
* LinearSVC

1. **Benchmark**

Para análise dos resultados foram incluídas as matrizes de confusão, sendo assim de uma forma simples e fácil é possível entender se o método estava obtendo sucesso na predição.

Conforme já mencionado, se a média da matriz de confusão for de 80% já considero esse projeto um sucesso, pela quantidade de dados inciais, além do quão abrangente análise de texto em um ambiente sem restrições como o twitter pode ser.

**Methodology**

1. **Data Preprocessing**

Como já descrito, após HTTP GET diretamente do twitter, colocar em um arquivo json, parser via código em .NET com limpeza de caracteres especiais e virgulas. A junção em um documento único foi feita manualmente. A limpeza por split words utilizando python e remoção de endereços http para o máximo de limpeza no campo de texto.

Outro dado que precisou de pré processamento foi o campo de data, já que estava em formato não reconhecido facilmente pelo datetime do python, então foi feito um parser manual para adequação desse formato.

1. **Implementation**

Para a implementação completa um dos requerimentos era a criação de uma API com endpoint para consulta dos resultados dessa predição. Para isso foi utilizado o microframework Flask, utilizado para transformar códigos python em ambientes consultivos via web.

Outro diferencial da implementação foi o encapsulamento da predição final, para isso foi utilizado o serializador dill, que é um modulo baseado em pickle para serialização e deserialização de objetos. Com isso todo um ambiente produtivo consultivo desse algoritmo fica extremamente rápido e fácil de implantação.

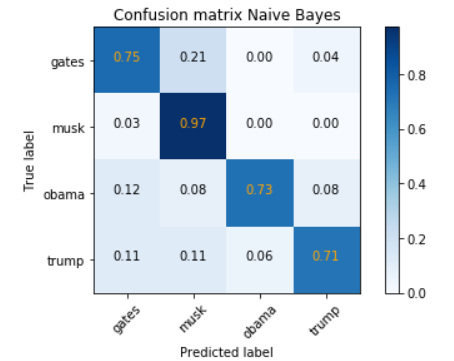
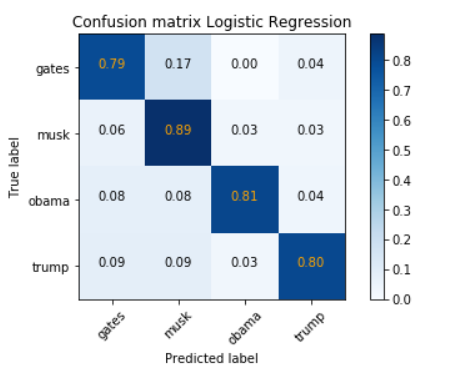
1. **Refinement**

Todo o trabalho em refinamento de algoritmo parecia refletir em breves pontos do resultado na matriz de confusão, mas assim que era alterado o Randon state ou o tamanho da base de dados os refinamentos e tunnings não faziam sentidos, por isso todos classificadores foram utilizados em suas chamadas padrões.

**Results**

1. **Model Evaluation and Validation**

O classificador feito utilizando o método de Regressão Logistica foi o que mais chamou a atenção devido a média de acertos que teve realizando a predição na base de teste, por esse motivo foi escolhido para representar o projeto, conforme mostra a Imagem 2.



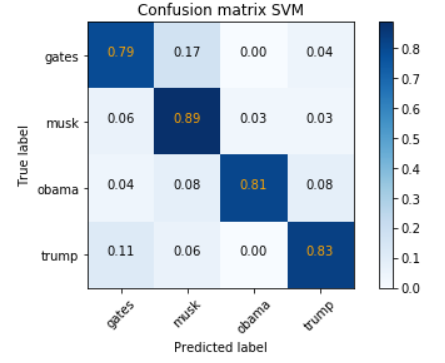
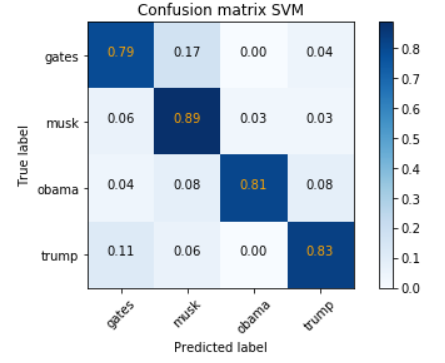


Imagem 2 : Matriz de confusão

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model’s solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- \_Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?\_

- \_Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?\_

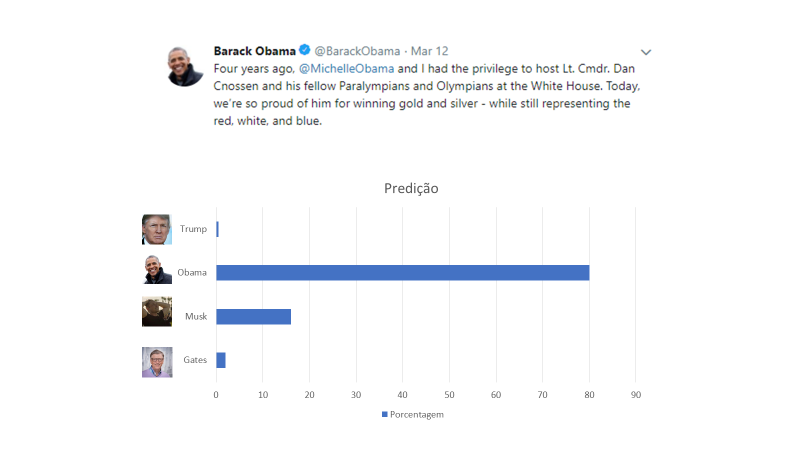
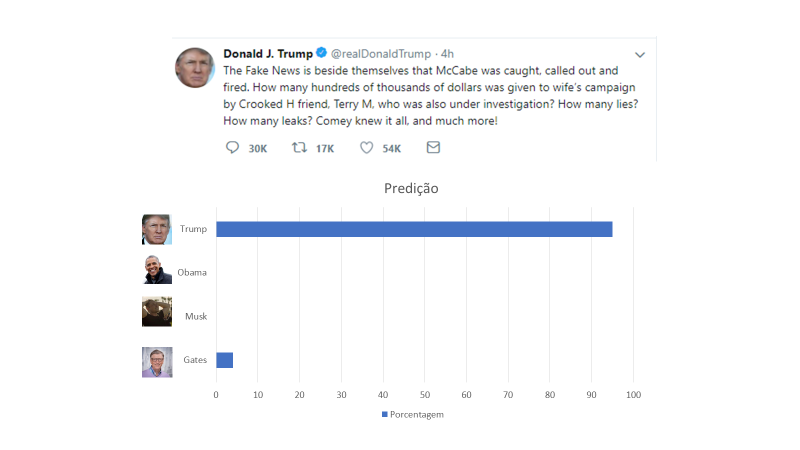
- \_Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?\_

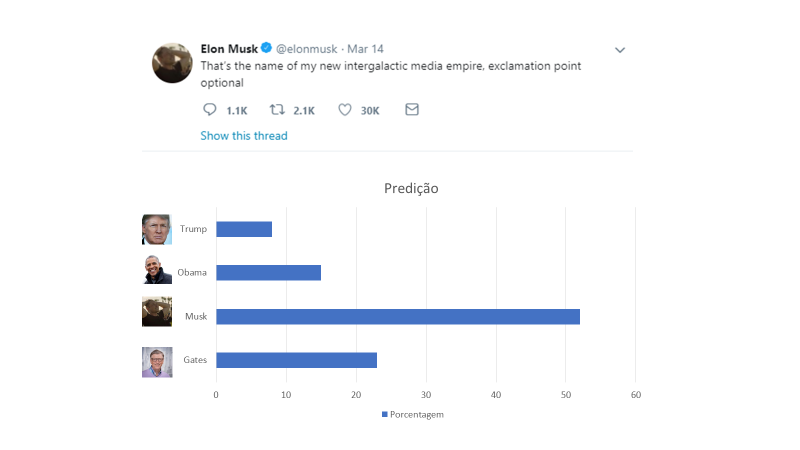
- \_Can results found from the model be trusted?\_

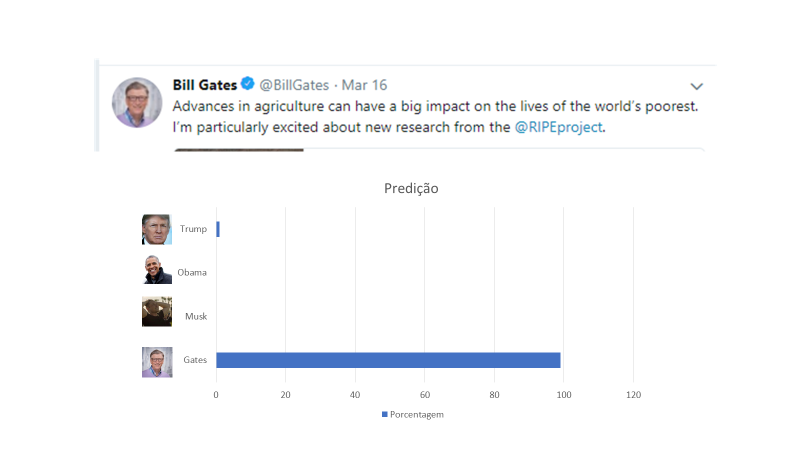
1. **Justification**

Considerando a média de 80% de sucesso estabelecida e utilizando todo o cenário funcionando acredito que o que foi estabelecido antes da finalização do projeto foi alcançado, mesmo estando passível a diversas melhorias e implementações para melhoria.

Realizando um teste “ao vivo” com o ultimo twitt de cada uma da celebridade (não presente no treino ou teste) para validar se a predição está correta:







**Conclusion**

\_(approx. 1-2 pages)\_

1. **Free-Form Visualization**

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- \_Have you visualized a relevant or important quality about the problem, dataset, input data, or results?\_

- \_Is the visualization thoroughly analyzed and discussed?\_

- \_If a plot is provided, are the axes, title, and datum clearly defined?\_

1. **Reflection**

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- \_Have you thoroughly summarized the entire process you used for this project?\_

- \_Were there any interesting aspects of the project?\_

- \_Were there any difficult aspects of the project?\_

- \_Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?\_

1. **Improvement**

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- \_Are there further improvements that could be made on the algorithms or techniques you used in this project?\_

- \_Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?\_

- \_If you used your final solution as the new benchmark, do you think an even better solution exists?\_