

PREVISÃO DE SUCESSO NO LANÇAMENTO DE PRODUTOS E SERVIÇOS A PARTIR DE ANÁLISE DE SENTIMENTOS EM TWEETS
FORECASTING SUCCESS IN THE LAUNCH OF PRODUCTS AND SERVICES FROM THE FEELINGS ANALYSIS IN TWEETS

TECHEIRA, Bruno H.; MARCONDES, Giovanna;
Graduandos do Curso de Engenharia da Computação – Universidade São Francisco
bruno.techeira@mail.usf.edu.br; giovanna.marcondes@mail.usf.edu.br

RESUMO. Este artigo propõe um estudo voltado para o setor de Marketing que, por meio da internet sofreu mudanças abruptas nos conceitos de lançamento de um produto e serviço ou até na forma em que o consumidor adquire, recomenda e opina sobre o mesmo. Com o desdobramento para o Marketing 4.0 e o crescimento de aplicativos disruptivos, a ascensão da democratização dos meios de comunicação via redes sociais tem realizado a ponte entre o vendedor e o comprador. Parte deste princípio a necessidade de remodelar o processo de campanhas obsoletas em rede aberta de televisão e buscar pontos de melhoria dentro do âmbito digital que, por meio de redes sociais, trouxe o grau de incerteza e complexidade na escolha de estratégias compatíveis com a perspectiva dos negócios e clientes. Conseguir captar o conceito de confiança e transparência do público alvo é enxergar o próprio mercado que a empresa está enfrentando e, com isso, encontrar o fundamento-chave do marketing. As redes sociais como o *Twitter* facilitam aos usuários expressarem suas opiniões e compartilhar suas experiências, criando, assim, uma rede de inspiração para outros consumidores que buscam por experiências semelhantes. Diariamente é gerado grande volume de dados a partir de postagens denominadas *tweet*, através da qual é possível realizar coletas para análises de mercado em grande escala. Enxergando essa massa de dados atualizada e alimentada em tempo real como oportunidade de análise, nasce o objetivo do projeto, no qual, aplicando análise de sentimentos e de predição, é possível determinar se um produto ou serviço a ser lançado terá sucesso. Conclui-se, então, que com a criação de um *Dashboard* para visualização é possível trazer para o profissional uma nova ferramenta de observação para inspiração na criação de campanhas publicitárias de acordo com o produto ou serviço concorrente pesquisado.

Palavras-chave: Marketing 4.0, Twitter, Análise de Sentimentos, Análise Preditiva.

SUMMARY. This article proposes a study aimed at the Marketing sector, where through the internet it has undergone abrupt changes in the concepts of launching a product and service or even in the way in which the consumer purchases, recommends and opines on it. With the deployment to Marketing 4.0 and the growth of disruptive applications, the rise of democratization of media via social networks has bridged the gap between seller and buyer. Part of this principle is the need to reshape the process of obsolete campaigns on an open television network and seek improvement points within the digital scope, which through social networks brought the degree of uncertainty and complexity in the choice of strategies compatible with the business perspective and customers. To be able to capture the concept of trust and transparency of the target audience is to see the very market that the company is facing and thereby find the key foundation of marketing. Social networks like *Twitter* make it easy for users to express their opinions and share their experiences, thus creating a network of inspiration for other consumers looking for similar experiences. A large volume of data is generated daily from posts called tweets, where it is possible to carry out collections for

large-scale market analysis. Seeing this mass of data updated and fed in real time as an opportunity for analysis, the objective of the project is born, where by applying analysis of feelings and prediction it is possible to determine whether a product or service to be launched will be successful. It is concluded that with the creation of a Dashboard for visualization it is possible to bring to the professional a new observation tool for inspiration in the creation of advertising campaigns according to the competing product or service searched.

Keywords: Marketing 4.0, Twitter, Sentiment Analysis, Predictive Analysis.

INTRODUÇÃO

A transformação digital induziu mudanças em diversos setores, um deles se aplica na publicidade e propaganda onde o conceito até então conhecido por Marketing tem sua estratégia de comunicação com o cliente alterada e ganha a representação e tratamento de uma figura pública nas redes sociais, por trás do *brand* da empresa. Nos princípios básicos do Marketing 3.0, os consumidores eram facilmente influenciados pelas atividades e opiniões estabelecidas pela empresa, esse comportamento mudou com o aparecimento do Marketing 4.0, onde quanto mais social o consumidor é, mais ele está disposto a buscar produtos e atendimentos personalizados (KOTLER, 2017).

Em paralelo ao mercado de marketing digital e as alterações de perspectiva na construção de uma campanha, vem a necessidade das empresas tomarem novas medidas de destaque para chamar atenção de seu público. Com diversas marcas rivais frisando que a satisfação do cliente é a premissa da empresa, a concorrência se abre e a necessidade de ser interpretada com outros olhos é primordial. Parte deste princípio a importância de coletar opiniões sinceras de produtos ou serviços já existentes no mercado e mensurar o impacto que o mesmo está trazendo ao consumidor e a multidão ao seu redor. Este comportamento pode ser apontado através das redes sociais, como por exemplo o *Twitter* que facilita aos usuários expressarem e compartilharem suas experiências, criando assim uma rede de inspiração para outros consumidores que buscam por experiências semelhantes. Comparando os usuários ativos e o volume de dados criados, o *Twitter* possui 230 milhões de usuários que compartilham 460 milhões de tweets por dia (CANALTECH, 2020). Com esta quantia de dados gerados diariamente fica cada vez mais evidenciado a necessidade de uma ferramenta que consiga captar o sentimento expresso pelo usuário ao comprar, utilizar ou consumir o produto ou serviço ofertado.

A idéia parte desta lacuna, auxiliando através da inteligência artificial, *machine learning* e ciência de dados, os profissionais do setor de publicidade e propaganda à obter uma visão rápida e eficiente através das análises realizadas em *tweets*. Onde o produto ou serviço a ser lançado irá ou não acarretar em sucesso comparando com seus concorrentes. A ferramenta desenvolvida conta com o formato de *Dashboard*, apresentando três painéis e um gráfico pizza contabilizando os pontos positivos, negativos e neutros a partir da palavra-chave pesquisada e um gráfico de regressão linear, onde será aplicado a predição. Acima dessas representações temos um campo de busca (onde é digitado a palavra-chave a ser pesquisada), campo de data inicial, campo de data final e o botão de pesquisar. O resultado é gerado em tempo real, então constantemente sofre alterações.

METODOLOGIA

Neste estudo, inicialmente o foco foi definir a escolha da linguagem de programação a ser utilizada, tendo em mente a necessidade do uso de bibliotecas de análise de dados, conceito de *machine learning* e inteligência artificial. Com este pressuposto, a linguagem de programação conhecida por Python, trás para o desenvolvimento versatilidade, fácil acesso às bibliotecas e implementação com outras aplicações, sendo a escolha ideal para atingir o objetivo do projeto. Com o ponto de partida definido, é necessário obter os dados através da rede social *Twitter*, onde por meio de uma API a aquisição dos dados se define através de um filtro contendo a palavra-chave de busca, data inicial e data final, limitando a quantidade e atualidade da postagem. O armazenamento dessas informações ocorre através da integração com o banco de dados não relacional conhecido por Firebase Realtime Database, onde realiza um papel importante para a rápida manipulação de dados na nuvem em formato de árvore JSON, auxiliando futuramente no tratamento dos dados recém examinados pelas bibliotecas de análise.

Desenvolvida também pela linguagem de programação Python o analisador de sentimentos avalia a partir da entrada de um texto a classificação léxica de aprendizado de máquina o retorno da pontuação entre zero e um, quanto mais próximas de um o sentimento é positivo, enquanto as classificações mais próximas de zero indicam sentimento negativo. Para auxiliar nos experimentos de predição foi definido a classificação entre positivo, negativo e neutro. Com isso, o resultado avaliado por cada *tweet* recebe o modelo de regressão linear para definir a predição de sucesso no lançamento de produtos e serviços, através da avaliação de seus concorrentes. Para exibição dessas informações conduzindo ao objetivo do projeto é necessário a criação de uma estrutura de aplicação *web*, onde o *Dashboard* apresentado se torna destaque, servindo de interação com o usuário que pesquisa por produtos ou serviços concorrentes e recebe o retorno da avaliação realizada pelas análises de sentimentos e predição. Contanto com esta estrutura, torna-se necessário a utilização dos seguintes recursos, conforme a ordem abaixo:

- Framework Django utilizando o padrão model-template-view;
- Tratamento *web* com HTML5, CSS, JS e Ajax;
- Implementação do Chart.js para criação de gráficos.

A etapa de criação da aplicação *web* é considerada final, pois é necessário já estar com as tratativas de dados definidas e em funcionamento, além das implementações de banco de dados, bibliotecas e API. Para maior entendimento de como o projeto foi concluído, as próximas seções irão conter o detalhamento de cada etapa aqui descrita e os estudos realizados para a evolução da mesma.

Obtenção de Dados

Partindo da necessidade de obter opiniões sinceras de consumidores reais dentro da rede social *Twitter*, a primeira tentativa foi através da API oficial que acabou se tornando ineficiente porque os registros consultados são restringidos em uma semana. Com isso, pesquisando dentro das diversas opções encontradas na internet, sendo pagas ou gratuitas, o projeto encontrado foi o “*Get Old Tweets*”¹ onde nos baseamos para realizar o projeto. A ferramenta foi desenvolvida em Python e tem como objetivo listar informações de todos os *tweets* de contas públicas dentro de um filtro que contém data de início, fim e o nome de busca. As informações que podem ser obtidas são as seguintes: o id da postagem (*tweet.id*), o

link da postagem (`tweet.permalink`), o nome do usuário (`tweet.username`), conteúdo do tweet (`tweet.text`), a data da postagem (`tweet.date`), os retweets (`tweet.retweets`), quantidade de favoritos (`tweet.favorites`), às menções relacionadas ao usuário (`tweet.mentions`), as hashtags utilizadas (`tweet.hashtag`), a localização (`tweet.geo`), a URL de anexo (`tweet.urls`) e o id do usuário (`tweet.author_id`). Os conteúdos até então manipulados não estão incluídos no banco de dados em nuvem, pois nesta etapa é necessário já ter a classificação de positivos, negativos e neutros vindos da análise de sentimentos. Em contrapartida pode-se concluir que nesta seção a obtenção dos dados do *Twitter* foi concluída com sucesso.

Análise de Sentimentos

Com os *tweets* tratados corretamente, a implementação de um analisador de sentimentos fica responsável pela ferramenta chamada VADER. Também desenvolvida em Python, ela faz uso de análise de sentimento lexical baseadas em regras e adaptada para detectar sentimentos expressos em redes sociais, pois utiliza de um banco de dados de *tweets* como parâmetro. A construção do dicionário léxico se deu através de uma lista de palavras baseadas em dicionários já bem estabelecidos como o LIWC, ANEW e GI e em seguida na inclusão de numerosas construções lexicais como emoticons, acrônimos e gírias que expressam sentimentos. Sendo uma ferramenta Open Source, os autores Hutto e Gilbert, garantem fácil manipulação e instalação. O VADER tem obtido sucesso na análise de textos em mídias sociais, editoriais, críticas de filmes e produtos, seguindo a proposta do estudo. Neste momento, o conteúdo de texto e *hashtag* retornado da API “*Get Old Tweets*” terá análise obtida pelo VADER e seguirá para próxima etapa.

Estrutura Banco de Dados

A plataforma adquirida pela Google em 2014, conhecida por Firebase auxilia no desenvolvimento e gerenciamento de back-end em aplicações *mobile* e *web*. Entre as ferramentas disponíveis temos o Firebase Realtime Database, utilizando o conceito NoSQL com armazenamento gratuito de dados alocado em nuvem. Após atrelar a conta Google ao Firebase, já é possível realizar a criação da estrutura do banco de dados.

Estrutura da Aplicação Web

Em uma aplicação *web* é necessário a definição das ferramentas a partir da linguagem de programação escolhida para o desenvolvimento, com isso o uso de *frameworks* ou de bibliotecas são voltadas a linguagem Python, complementando o estudo. A estrutura criada foi com o uso da biblioteca Virtualenv, onde ao implementar ao projeto cria-se um ambiente virtual que empacota todas as dependências necessárias e as armazena em um diretório, isolando do sistema operacional a instalação de pacotes. Para auxílio no desenvolvimento front-end, o *framework* Django entra com o conjunto de códigos facilitando o desenvolvimento rápido *web* a partir do modelo padrão de *model-template-view* (MTV).

- *Model*: Mapeamento do banco de dados para o projeto;
- *Template*: Visualização de dados nas páginas *web*;
- *View*: Lógica de negócio.

O item *template*, é o local destinado para a implementação da tríade das tecnologias *web* como o HTML, CSS e o JS. O HTML (Linguagem de Marcação de Hipertexto) tem papel fundamental na construção de blocos básicos sendo interpretado por navegadores, em

customização de aparência temos o CSS (*Cascading Style Sheets*) onde aplica-se estilo ao documento e o JS (JavaScript) entra como o responsável por manipular comportamentos na página. Além da implementação das ferramentas AJAX e Chart.JS. Onde neste estudo, o AJAX (*Asynchronous JavaScript e XML*) auxilia com a interação da aplicação e usuário devido às solicitações assíncronas de informações com a estrutura do HTML. E, por fim, a biblioteca em JavaScript conhecida por Chart.JS e que realiza a visualização de dados a partir de gráficos como de barra, pizza, bolha, polar, radar e dispersão, sendo ponto chave para o objetivo do projeto.

Algoritmo de Predição

Diante das opções de análise preditiva é comum o uso de métodos vindos da análise de regressão, tais como estimativa de parâmetros, análise de variância e de resíduos, testes de hipóteses, lidando com a formulação de modelos matemáticos que descrevem relações entre variáveis e o uso desses modelos com o propósito de predição e outras inferências estatísticas (CAMPOS FILHO, 2018). Entre os métodos de análise de regressão, temos a regressão linear simples que a partir do método estatístico surge a relação entre duas variáveis (X e Y), podendo ser usada para prever o valor numérico futuro. O comportamento de Y em X pode apresentar diversas formas: linear, quadrático, cúbico, exponencial etc... . O modelo linear de primeiro grau consiste no modelo estatístico abaixo na Equação 1:

$$Y_i = \beta_0 + \beta_1 X_i + e_i$$

Equação (1)

Sendo o Y_i , o valor observado para a variável dependente Y no i-ésimo nível da variável independente X. O β_0 , a constante de regressão sendo o intercepto da reta com o eixos do Y. A variável β_1 o coeficiente de regressão, representando a variação de Y em função da variação de uma unidade da variável X. O X_i , é o i-ésimo nível da variável independente X ($i = 1, 2, \dots, n$). E por fim o e_i , que representa o erro associado a distância entre o valor observado Y_i e o correspondente ponto na curva, para o mesmo nível i de X. Na Equação 2, para obter a equação estimada é necessário utilizar o método dos mínimos quadrados (MMQ), minimizando erros e em sequência (Equação 3) elevando os membros da equação ao quadrado.

$$e_i = Y_i - \beta_0 - \beta_1 X_i$$

Equação (2)

$$e_i^2 = [Y_i - \beta_0 - \beta_1 X_i]^2$$

Equação (3)

Aplicando o somatório é possível alcançar a minimização da soma de quadrados dos erros (Equação 4), por meio da obtenção de estimadores β_0 e β_1 . Ao realizar sua derivação em relação a variável de interesse e igualá-la a zero encontra-se o valor mínimo, com isso poderemos obter duas equações que, juntas, vão compor o chamado de sistemas de equações normais na Equação 5.

$$\sum_{i=1}^n e_i^2 = \sum_{i=1}^n [Y_i - \beta_0 - \beta_1 X_i]^2$$

Equação (4)

$$\hat{\beta}_1 = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{n}}{\sum x_i^2 - \frac{(\sum x_i)^2}{n}} = \frac{SPD_{xy}}{SQD_x} \text{ e } \hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{X}$$

Equação (5)

Uma vez obtidas estas estimativas, podemos escrever a equação final da regressão linear simples (Equação 6):

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

Equação (6)

Após a aplicação das equações apresentadas, é necessário realizar a representação gráfica e obter a visualização dos dados observados e a linha calculada da regressão linear. Para integrar a análise preditiva de regressão linear simples com a aplicação *web*, foi utilizado a biblioteca de aprendizado de máquina conhecida por *scikit-learn*, aplicada para estudos de *machine learning*, ciência de dados e inteligência artificial traz vantagens por ser desenvolvida em Python, auxiliando na implementação e desenvolvimento.

RESULTADOS E DISCUSSÃO

Para a obtenção de dados foi utilizado como base o projeto já mencionado “*Get Old Tweets*”, diante das opções de variáveis apresentadas foram selecionadas apenas três, sendo que o restante viria a ser irrelevante para o propósito de análise de sentimentos e predição. Os campos de data (*tweet.date*) e de conteúdo do *tweet* (*tweet.text*), tornando-se fundamento para o estudo realizado. Para assegurar o funcionamento do projeto utilizado, foi criado uma postagem real no *Twitter* no dia 05/07/2020 utilizando um usuário com conta pública e

carregando como conteúdo a *hashtag* “#testegettweets”. Com isso, para realizar a busca pelo tweet recém postado, foi utilizado como variáveis os seguintes parâmetros: a palavra-chave de busca, sendo #testegettweets, a data inicial em 2020-07-05 00:00:00 e a data final em 2020-07-05 23:59:59. O retorno do *tweet* postado apresentou o formato árvore JSON e foi exibido a partir do código da figura 1, com o conteúdo adaptado abaixo:

Tweet 1 sobre #testegettweets: Data: 2020-07-05 16:25:14 Texto: tweet de teste para API #testegettweets

Figura 1 - Código de obtenção de dados

```
1 import GetOldTweets3 as got
2 from datetime import datetime
3
4 def pesquisarTweets(parametro, dataInicial, dataFinal):
5
6     tweetCriteria = got.manager.TweetCriteria().setQuerySearch(parametro).setSince(dataInicial).setUntil(dataFinal)
7     tweets = got.manager.TweetManager.getTweets(tweetCriteria)
8
9     returnTweets = []
10
11     for t in tweets:
12         returnTweets.append({
13             "data": datetime.strftime(t.date, "%Y-%m-%d"),
14             "texto": t.text
15         })
16
17     return returnTweets
18
```

Autor: Fonte própria.

A primeira linha do código, conforme figura 1 foi importado a biblioteca *GetOldTweets3* e em seguida, a biblioteca *datetime* para a tratativa de variáveis com o uso de data e hora. Na linha 4, ocorre a definição da função *pesquisarTweets*, onde é utilizado de parâmetro a variável para palavra-chave, data inicial e final. Utilizando da primeira biblioteca importada, na linha 6 foi criado um critério de pesquisa a partir dos valores mencionados anteriormente, sendo que na linha seguinte é realizada a consulta de todos os tweets que estejam dentro deste critério. O array construído na linha 9 trabalha com um laço iniciado na linha 11, colocando no final da estrutura um objeto com os atributos de data e texto, presentes nos *tweets* obtidos e retornados na linha 17. O funcionamento até então estava ocorrendo com sucesso, porém no dia 18/09/2020 a biblioteca *GetOldTweets3* apresentou problemas na consulta de *tweets* e o seu uso foi descontinuado, a rede social *Twitter* realizou mudanças internas em sua ferramenta e acabou afetando a URL de consulta. Em busca de uma alternativa para continuar com o estudo até então proposto, foi utilizada base de dados disponibilizada pela *Computer Science and Engineering UC San Diego*, trazendo como informação a data da postagem e o conteúdo do *tweet*, sendo o necessário conforme o objetivo proposto de aplicação de análise de sentimentos e predição. Com os dados já obtidos, foi realizada a integração ao banco de dados *Firebase Realtime Database* com o uso da biblioteca *pyrebase*, criando a conexão para o armazenamento dos dados, conforme na figura 2.

Figura 2 - Credenciais para realizar conexão com o banco de dados.

```
import pyrebase

config = {
    "apiKey": "AIzaSyBj. ",
    "authDomain": "mtc2020-bbf91.firebaseio.com",
    "databaseURL": "https://mtc2020-bbf91.firebaseio.com/",
    "projectId": "mtc2020-bbf91",
    "storageBucket": "mtc2020-bbf91.appspot.com",
    "messagingSenderId": "597848420637",
    "appId": "1:597848420637:web: ",
    "measurementId": "G-23TX2WR881"
}

firebase = pyrebase.initialize_app(config)
db = firebase.database()
```

Autor: Fonte própria.

Ainda analisando a figura 2, com a biblioteca importada na linha 1 é realizada em seguida a configuração a partir das credenciais geradas no *Firebase*, conforme o conjunto no objeto na linha 3. Na linha 14, a biblioteca foi inicializada e na linha seguinte a conexão foi estabelecida. Na figura 3, a primeira linha é importada a conexão para adquirir a função de inserção de dados a partir do comando *input*, o qual incluído na coleção baseada nas pesquisas e nomeado por “*posts*”, gerando uma chave e incluindo seu conteúdo. Além disso, foi definida a função para a recuperação dos dados nomeada de “*get*”, onde apresenta a opção de encontrar todos os posts através da pesquisa, que é a coleção inserida anteriormente ou pesquisar por uma chave específica dentro dela. Com a etapa de conexão com o banco estabelecida, o próximo passo é a leitura dos dados junto com a análise de sentimento, que a partir da biblioteca *VADER* a informação examinada irá receber a pontuação e em seguida ser incluída na base de dados, criando sua estrutura.

Figura 3 - Código da estrutura inicial para inclusão no banco de dados.

```
from firebase import db

def Input(pesquisa, chave, conteudo):
    db.child("posts").child(pesquisa).child(chave).set(conteudo)

def Get(pesquisa, chave):
    if chave != "":
        posts = db.child("posts").child(pesquisa).child(chave).get()
    else:
        posts = db.child("posts").child(pesquisa).get()

    return posts
```

Autor: Fonte própria.

Figura 4 - Código da estrutura para inclusão no banco de dados com aplicação da análise de sentimentos.

```
import json
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
import function_firebase as fb

array = [ ]

with open('csvjson.json', 'r', encoding='utf-8') as json_file:
    data = json.load(json_file)
    for p in data:
        array.append({
            'data': p['data'],
            'texto': p['desc']
        })

i = 0
analyser = SentimentIntensityAnalyzer()

while( i < len(array) ):
    analise = analyser.polarity_scores(array[i]['texto'])
    salvar = {
        'data': array[i]['data'],
        'texto': array[i]['texto'],
        'neg': analise['neg'],
        'neu': analise['neu'],
        'pos': analise['pos'],
        'compound': analise['compound']
    }
    print(salvar)
    fb.Input("teste", i, salvar)
    i = i+1
```

Autor: Fonte própria.

Na linha 1, da figura 4 foi importado a biblioteca *json* para trabalhar com o modelo para armazenamento e transmissão de informações de mesmo nome, em seguida a biblioteca para a análise de sentimentos VADER e na terceira linha, o arquivo com as funções de *input* e *get* do banco de dados. Na linha 5, foi criado um array que armazena as datas e postagens do *tweets* do arquivo utilizado de base de dados retirados da *Computer Science and Engineering UC San Diego*¹, conforme estrutura apresentada da linha 7, com o nome do arquivo sendo *csvjson.json*. Com os dados incluídos na variável nomeada *data* na linha 8 e realizando um *loop* na linha 9, os dados são inseridos em um array criado anteriormente na estrutura da linha 10 a 13. Um contador é iniciado na linha 15, onde gera uma chave única e sequencial para cada post que em seguida será analisado pela variável *análise*, onde é aplicado o analisador de sentimentos VADER. Na linha 18, é criado um *loop* para percorrer o conteúdo do *array* criado, em seguida na linha 19 a realização da análise de sentimentos. Por fim, na linha 20 é criado a estrutura a ser enviada para a base de dados através da função *input*. Segundo a figura 5, os campos são representados pelo identificador sequencial (*id*), pontuação lexical do texto (*compound*), data de postagem (*data*), campo negativo (*neg*), neutro (*neu*), positivo (*pos*) e o conteúdo do *tweet* (*texto*).

Figura 5 - Registro do banco de dados Firebase



Autor: Fonte própria.

A realização da predição foi a partir do uso da biblioteca *scikit-learn*, onde na primeira linha foi importado a biblioteca referente a predição linear. Na linha 3, definido a função onde é passado os períodos e resultados relacionados às datas que deseja ser feita a previsão, além da quantidade de elementos informados. Na linha seguinte é criado a variável, na qual é aplicado a regressão. Utilizando a função *fit* na linha 5, a biblioteca trata de treinar a predição para que possamos preencher a variável definida na linha 7, através do *loop* seguido da linha 9, onde o *range* utilizado é a partir da data inicial selecionada até a final + 1/10 deste período definido.

Figura 6 - Código Regressão Linear

```
from sklearn.linear_model import LinearRegression

def predicao(periodos, resultados, quantidadeElementos):
    reg = LinearRegression()
    reg.fit(periodos, resultados)

    predict = []

    for x in range(int(quantidadeElementos+(quantidadeElementos/10))):
        predict.append(reg.predict([[x]]).tolist())

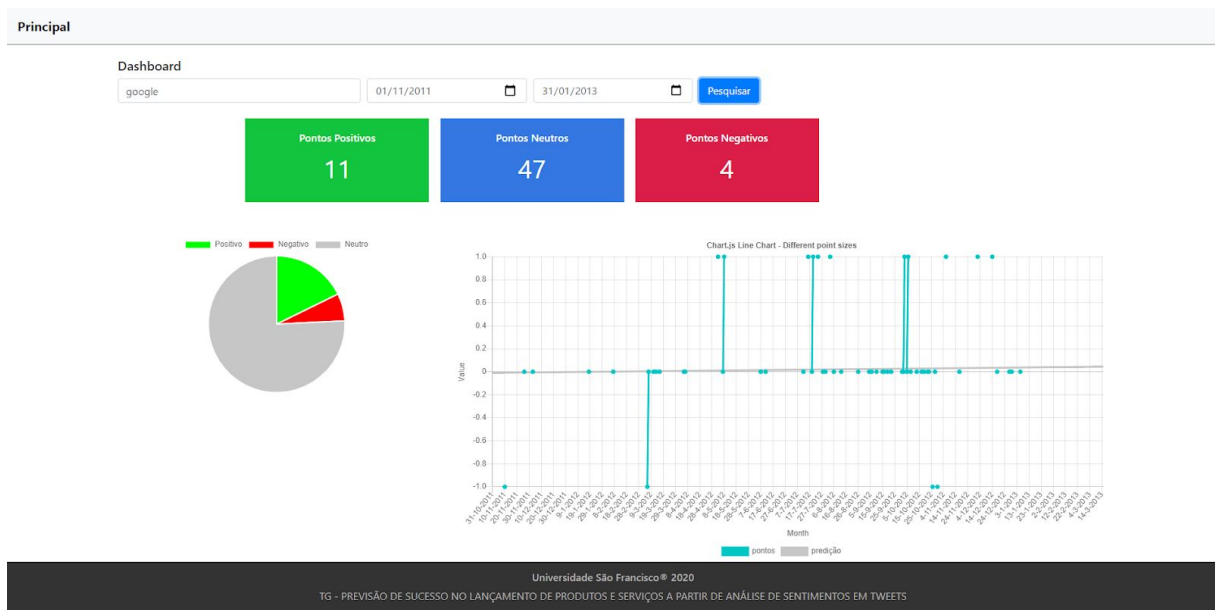
    return predict
```

Autor: Fonte própria.

Com os processos de desenvolvimento *back-end* finalizados e a análise de sentimentos, inclusão de dados no banco Firebase e a aplicação da predição em regressão linear em funcionamento, a etapa seguinte é apresentar as informações trabalhadas em formato de *Dashboard* através da aplicação *web*. Com a estrutura e organização de pastas já pré-definidas a partir do uso da biblioteca *Virtualenv* e framework *Django*, a construção foi separada na criação de um bloco base de HTML, projetando a nomenclatura do título, cabeçalho, corpo do site e rodapé. Com isso, tivemos a inclusão via HTML das ferramentas

JQuery, Bootstrap, FontAwesome e complementada em paralelo com o uso das ferramentas *web* CSS e JS, auxiliando na criação de botões, painéis e campos de digitação presentes na página, conforme imagem 7. A visão inicial foi criada, o ponto seguinte é exibir o conteúdo até então presentes somente na base de dados para os três painéis indicando os pontos positivos, negativos e neutros, este processo é realizado através do AJAX no próprio HTML. Para finalizar, temos a inclusão do Chart.js para criação de um gráfico em formato pizza à direita para representar a análise de sentimentos e à esquerda um gráfico em formato bubble para representar a predição em regressão linear, apresentado na figura 7.

Figura 7 - Aplicação *web*



Autor: Fonte própria.

Os resultados apresentados nos testes foram baseados no peso da quantidade de sentimentos positivos, negativos e neutros sobre a regressão linear, apresentando ou não o distanciamento dos pontos do eixo principal. Quando temos um valor maior na pontuação de sentimentos positivos temos também maior aproximação na reta da regressão, sendo o comportamento contrário em casos de pontuação negativa, nas pontuações neutras essas aproximações são contingentes. Na figura 7, temos o valor 47 representando os sentimentos neutros e observando o gráfico de predição temos alguns pontos se distanciando da reta principal tomada como base e outros se aproximando da mesma. Conclui-se então que, quanto mais pontos próximos à reta presente na análise de regressão linear, maiores chances de os produtos obterem sucesso em seu lançamento.

CONCLUSÃO

A transferência de dados em tempo real e em qualquer lugar traz o conceito de velocidade e volatilidade para a realização de análises a partir dos conteúdos postados em rede social, ação que se tornou rotineira entre os usuários presentes no mundo virtual e que pode auxiliar diversos setores que fazem uso dessa tecnologia. Com a ascensão do mercado digital, surgiram novas empresas e, com elas, o lançamento de produtos ou serviços concorrentes. Com isso, os conceitos de marketing e a relação de empresa x cliente voltaram-se para as redes sociais, acarretando na necessidade de interpretar aceleradamente qual a reputação que o produto ou serviço está refletindo nas mídias.

Uma nova empresa que estuda iniciar o lançamento de seu produto ou serviço necessita a obtenção de uma visão geral através das mídias digitais de como está a reputação de seu concorrente, estando um passo à frente e definindo regras de negócio para alcançar o sucesso. Com o estudo apresentado conclui-se que a análise de sentimentos e de predição funcionam dentro da premissa do objetivo do trabalho, porém não garantem para os profissionais dentro do setor de publicidade e propaganda o sucesso no lançamento de seu produto ou serviço, tornando-se uma análise simples. Para trabalhos futuros, podem ser melhoradas a aquisição de dados através de outra API e a implementação de novas técnicas e ferramentas para definição de pontos de melhoria ou das falhas presentes na empresa concorrente.

REFERÊNCIAS

BARI, A.; CHAOUCHI, M. J. **Análise Preditiva Para Leigos: Tradução da 2ª Edição**. Alta Books. Rio de Janeiro. 2019. p.181.

CAMBRIA, E.; SPEER, R.; HAVASI, C.; HUSSAIN, A. Senticnet: **A publicly available semantic resource for opinion mining**. AAAI Fall Symposium Series, 2010. p.53.

CAMPOS FILHO, F. F. **Algoritmos numéricos - uma abordagem moderna de Cálculo Numérico**. 2018.p. 225

CANALTECH. **Metade dos usuários ativos do Twitter está em apenas cinco países, diz pesquisa**. Disponível em:
<<https://canaltech.com.br/redes-sociais/Metade-dos-usuarios-ativos-do-Twitter-esta-em-ape-na-s-cinco-paises-diz-pesquisa/>>. Acesso em: 20 mai. 2020.

EGAN, J. **Marketing Communications**. London: Thomson, 1988. p. 43.

GONÇALVES, P.; BENEVEDUTO, F. **Panas-t: Uma escala psicométrica para análise de sentimentos no twitter**. BraSNAM, 2012. p.2.

HENRIQUE, J. **Get Old Tweets Programmatically**. Disponível em:
<<https://github.com/Jefferson-Henrique/GetOldTweets-python>> Acesso em: 21 jun 2020

MCAULEY, J. **Recommender Systems Database**. Disponível em:
<<https://cseweb.ucsd.edu/~jmcauley/datasets.html#socialcircles>>. Acesso em: 20 de Outubro de 2020.

KOTLER, P. **Marketing 4.0: Do tradicional ao digital**. Sextante, 2017. p. 9-103.

MACHADO, L. G. P. J.; OLIVEIRA, N. P. **Marketing 4.0: Estudo de caso da gestão de influenciadores digitais em uma startup**. UFRJ, Rio de Janeiro, p. 26-30, jan./mar. 2018

Hutto, C.J. & Gilbert, E.E. (2014). **VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media** Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014

OKADA, I. S.; SOUZA, S. M. E. **Estratégias de marketing digital na era da busca**. Revista Brasileira de Marketing, São Paulo, v. 10, n. 1, p 46-72, jan./abr. 2011

PANG, B.; LEE, L. **Opinion mining and sentiment analysis**. Foundations and Trends in Information Retrieval, U.S.A, v. 2, n. 1-2, p 1–135, 2008

THELWALL, M. **Heart and soul: Sentiment strength detection in the social web with sentistrength**, Cyberemotions. p1-14.