



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

SAMUEL PEREIRA DE VASCONCELOS

**SAFETRIP: UMA FERRAMENTA PARA DETECÇÃO E NOTIFICAÇÃO
DE ACIDENTES EM RODOVIAS BRASILEIRAS A PARTIR DO
TWITTER**

CAMPINA GRANDE - PB

2022

SAMUEL PEREIRA DE VASCONCELOS

**SAFETRIP: UMA FERRAMENTA PARA DETECÇÃO E NOTIFICAÇÃO
DE ACIDENTES EM RODOVIAS BRASILEIRAS A PARTIR DO
TWITTER**

**Trabalho de Conclusão Curso apresentado
ao Curso Bacharelado em Ciência da
Computação do Centro de Engenharia
Elétrica e Informática da Universidade
Federal de Campina Grande, como
requisito parcial para obtenção do título de
Bacharel em Ciência da Computação.**

Orientador: Professor Dr. Cláudio de Souza Baptista.

CAMPINA GRANDE - PB

2022

SAMUEL PEREIRA DE VASCONCELOS

**SAFETRIP: UMA FERRAMENTA PARA DETECÇÃO E NOTIFICAÇÃO
DE ACIDENTES EM RODOVIAS BRASILEIRAS A PARTIR DO
TWITTER**

**Trabalho de Conclusão Curso apresentado
ao Curso Bacharelado em Ciência da
Computação do Centro de Engenharia
Elétrica e Informática da Universidade
Federal de Campina Grande, como
requisito parcial para obtenção do título de
Bacharel em Ciência da Computação.**

BANCA EXAMINADORA:

Professor Dr. Cláudio de Souza Baptista

Orientador – UASC/CEEI/UFPG

Professor Dr. Leandro Balby Marinho

Examinador – UASC/CEEI/UFPG

Professor Dr. Tiago Lima Massoni

Professor da Disciplina TCC – UASC/CEEI/UFPG

Trabalho aprovado em: 06 de Abril de 2022.

CAMPINA GRANDE - PB

ABSTRACT

The road system is the main means of transport in Brazil. Traffic accidents are quite common in this mode of transport, incurring in one of the biggest causes of death in the country. Profiles on social networks of the Federal Highway Police (FHP) and other sources of information, contribute to alert drivers about road accidents that have occurred, in order to reduce the frequency of this incident. However, accessing this information by reading while driving is illegal and further increases the risk of accidents. Therefore, this work proposes to develop a study about publications in social networks, in particular Twitter, to create a classification model, based on machine learning, that is capable of classifying tweets about information about the occurrence or not from road accidents. Expected results include a tweet classification model, as well as a mobile application that can notify drivers of reported accidents on their way, in real time and through audio.

SafeTrip: Uma Ferramenta para Detecção e Notificação de Acidentes em Rodovias Brasileiras a partir do Twitter

Trabalho de Conclusão de Curso

Samuel Pereira de Vasconcelos
Universidade Federal de Campina Grande
Campina Grande, Paraíba
samuel.vasconcelos@ccc.ufcg.edu.br

Cláudio de Souza Baptista
Universidade Federal de Campina Grande
Campina Grande, Paraíba
baptista@computacao.ufcg.edu.br

RESUMO

O sistema rodoviário é o principal meio de transporte no Brasil. Os acidentes de trânsito são bastante comuns nesse modal de transporte, incorrendo em uma das maiores causas mortis no país. Perfis em redes sociais da Polícia Rodoviária Federal (PRF) e de outras fontes de informações, contribuem para alertar os motoristas sobre acidentes rodoviários ocorridos, no intuito de reduzir a frequência desta incidência. Porém, acessar essas informações por leitura ao dirigir, é ilegal e eleva ainda mais o risco de acidentes. Portanto, este trabalho se propõe a desenvolver um estudo a respeito de publicações em redes sociais, em particular o Twitter, para criação de um modelo de classificação, baseado em aprendizagem de máquina, que seja capaz de classificar tweets acerca de informações sobre a ocorrência ou não de acidentes rodoviários. Os resultados esperados incluem o modelo de classificação de tweets, bem como um aplicativo de celular que possa notificar os condutores sobre acidentes reportados no seu trajeto, em tempo real e através de áudio.

Palavras-chave

Acidentes de Trânsito, Machine learning, Rotulagem de dados, Coleta de Tweets, Aplicativo Mobile, Modelos de Classificação, Pré-processamento, Georreferenciamento.

Repositórios

<https://github.com/samuelcomputacao/maps-servidor>
<https://github.com/samuelcomputacao/trafego-seguro-app>
<https://github.com/samuelcomputacao/maps-crawler>

1. INTRODUÇÃO

O tráfego rodoviário brasileiro ocupa a quarta posição no ranking mundial [1], sendo responsável pela principal forma de locomoção tanto de pessoas como de mercadorias no Brasil [2]. A malha rodoviária brasileira possui uma extensão maior do que 1.720.700 quilômetros, por onde são transportadas, segundo a pesquisa Custos Logísticos no Brasil, realizada pela Fundação Dom Cabral [3], 75,9% de tudo que é produzido no país, restando apenas 9,2% para o modal marítimo, 5,8% para o aéreo, 5,4% para o ferroviário, 3% via cabotagem e 0,7% pelo transporte hidroviário.

Por se tratar de um meio de transporte bastante requisitado, o número de acidentes ocorridos é preocupante. Apenas em rodovias federais, de 2007 até 2020 foram registrados 1.852.483 acidentes de trânsito com um total de 99.365 mortes [4], impactando

negativamente a vida das pessoas de forma a gerar inúmeros prejuízos financeiros e psicológicos à sociedade brasileira.

Por outro lado, o avanço tecnológico torna as pessoas cada vez mais conectadas, via por exemplo, smartphones, permitindo uma comunicação rápida de informações. É nesse contexto que perfis em redes sociais, como páginas de jornais, da PRF e concessionárias de rodovias, realizam postagens de acidentes ocorridos no intuito de despertar a atenção de quem está transitando nas proximidades do local do acidente, com a finalidade de evitar que novos acidentes ocorram em decorrência do episódio em questão.

Destarte, desenvolver tecnologias que permitam que tais informações cheguem, em tempo real e de forma apropriada, aos motoristas que cruzarão um tal acidente é um desafio de interesse sócio-econômico e de saúde pública, que enseja no foco deste trabalho.

2. PROBLEMA E SOLUÇÃO

Uma das principais causas de mortalidade no Brasil decorre dos acidentes rodoviários. Tais acidentes também trazem outros danos à sociedade brasileira como amputações e várias sequelas impossibilitantes que afetam as famílias e trazem grande prejuízo sócio-econômico. Existem vários canais na Internet para alertar a população acerca de acidentes, sendo grande parte deles na forma de texto (e.g. *tweets*), o que inviabiliza o acesso em tempo real por parte dos motoristas. Também encontramos o aplicativo para smartphone *waze* [5], que tem como um de seus objetivos alertar motoristas enquanto trafegam nas rodovias, entretanto o aplicativo gera sua base de dados com base na comunidade, qualquer tipo de usuário, confiável ou não, pode contribuir na alimentação do sistema e isso pode torná-lo pouco confiável.

O uso de smartphones é algo indispensável para praticamente todas as pessoas, sendo a principal forma de comunicação para fins de trabalho, família e vida social. No trânsito, esse tipo de aparelho reduz a atenção do condutor, desviando sua visão e elevando os riscos de um acidente ocorrer, podendo comprometer sua vida e a de todos ao seu redor. Conduzir um automóvel requer total atenção, por este motivo, o Código de Trânsito Brasileiro [6] em seu artigo de número 252, inciso VI define o uso do telefone celular ou fones de ouvidos conectados a algum dispositivo sonoro enquanto dirige, como infração gravíssima, prevendo uma multa de duzentos e noventa e três reais e quarenta e sete centavos, além da imputação de sete pontos na Carteira Nacional de Habilitação do condutor.

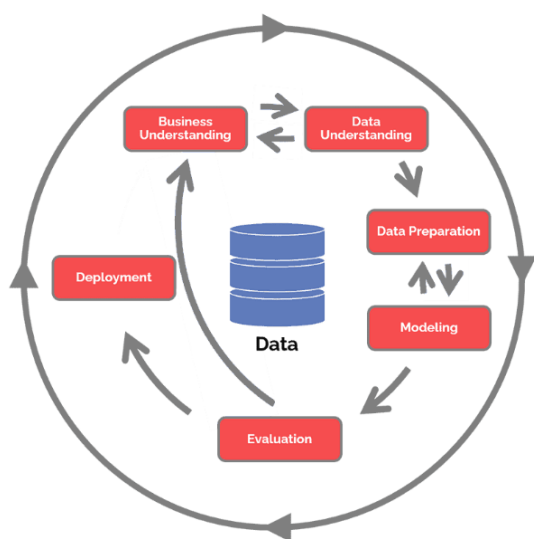
Pesquisas apontam que o hábito de manusear um smartphone ao volante é tão perigoso quanto dirigir sob efeito de álcool, pois enquanto que o envio ou leitura de mensagens de texto retarda em 35% a reação do condutor, a embriaguez tem um percentual bem menor, atingindo 12% [7]. Portanto, a iniciativa de realizar publicações em redes sociais pode não contribuir para a redução de acidentes, visto que a maioria desses *posts* estão em formato de texto.

Portanto, desenvolver tecnologias de comunicação via áudio, que propiciem alertar os motoristas, em tempo real e com antecedência, acerca de acidentes em rodovias, em trechos que ainda vão percorrer, monitorando a posição geográfica em tempo real do condutor e comparando-a à posição geográfica do acidente reportado é um importante aliado para mitigar o número de acidentes em rodovias brasileiras. Este trabalho se propõe a utilizar as publicações de perfis confiáveis da PRF no Twitter para fornecer uma solução que possa identificar, monitorar e alertar automaticamente os motoristas, através de áudio e em tempo real acerca de trechos de rodovias perigosos do ponto de vista de acidentes, usando técnicas de aprendizagem de máquina, geoprocessamento e Processamento de Linguagem Natural.

3. METODOLOGIA

Este projeto utilizou uma metodologia baseada no *Cross Industry Standard Process for Data Mining* (CRISP-DM) [8]. Esta metodologia preconiza seis etapas: (i) *Business Understanding*, (ii) *Data Understanding*, (iii) *Data Preparation*, (iv) *Modeling*, (v) *Evaluation* e (vi) *Deployment*. Conforme indicado na figura 1.

Figura 1: Etapas da metodologia CRISP-DM



Fonte: Figura disponibilizada pela Data Science Process Alliance¹

A etapa de Entendimento do Negócio (*Business Understanding*) foi contemplada na seção 2. A etapa de Entendimento dos Dados (*Data Understanding*) consistiu na identificação dos perfis da PRF no Twitter e na análise das palavras chaves a serem utilizadas para filtrar os tweets, bem como o processo de coleta dos dados. A etapa de Preparação dos Dados (*Data Preparation*) consistiu na implementação de técnicas de pré-processamento e de rotulagem

¹ What is CRISP DM?. Disponível em: <<https://www.datascience-pm.com/crisp-dm-2>>. Acesso em 28 de fev. de 2022.

manual dos dados para a criação de um corpus anotado. A etapa de Indução dos Modelos (*Modeling*) consistiu na escolha e implementação de vários modelos de indução, utilizando-se de classificação supervisionada, com os devidos ajustes de hiperparâmetros. A etapa Avaliação dos Modelos (*Evaluation*) realiza a análise dos resultados dos modelos que foram gerados na etapa anterior, escolhendo o modelo que apresentou o melhor desempenho. Por fim, a etapa Implementação da Ferramenta (*Deployment*), consiste na implementação do modelo na ferramenta SafeTrip, com uso em aplicativo móvel.

3.1. Entendimento dos Dados

3.1.1 Seleção de perfis

Durante a busca por perfis no twitter foram escolhidos perfis oficiais da PRF de vários estados do Brasil, pois esses possuem como características principais em suas publicações, alertas sobre acidentes ocorridos nas rodovias.

Os perfis escolhidos estavam todos ativos em Novembro de 2021 e são eles: @PRFParana, @PRFCeara, @PRF191RJ, @PRF191PR @PRF191TOCANTINS, @PRF191SP, @PRF191SERGIPE, @PRF191RORAIMA, @PRF191RONDONIA, @PRF191PE, @PRF191PA, @PRF191MS, @PRF191ES, @PRF191AM, @prf_sc, @PRF191ACRE, @prf_rn, @prf_pi, @prf_pb, @PRF_MS, @prf_mg, @prf_df, @prf_ba e @prf_al.

3.1.2 Seleção de Palavras

Para a seleção das palavras chaves que são comuns em publicações sobre acidentes em perfis da PRF, foram analisadas publicações semelhantes a da figura 2. Algumas palavras escolhidas possuem apenas o seu radical, em virtude de muitas palavras se repetirem em diferentes tempos verbais e conjugadas em diferentes pessoas do plural e singular.

As palavras escolhidas foram: 'ferid', 'acidente', 'mort', 'choque', 'tragedia', 'atropel', 'capota', 'vitima', 'colisao', 'virada', 'virou', 'engavetamento', 'incendio', 'abaloamento', 'tombamento', 'batida', 'saiu da pista', 'bateu', 'morre', 'faleceu' e 'tombou'.

Figura 2: Tweet de alerta sobre acidente



Fonte: Publicação do perfil @prf_pb².

² Disponível em: <https://twitter.com/prf_pb/status/1471339049898201088?ref_src=twsrc%5Etfw%7Ctwcamp%5Etweetembed%7Ctwt_erm%5E1471339049898201088%7Ctwtgr%5E%7Ctwtcon%5E1_c10&ref_url=https%3A%2F%2Fpublish.twitter.com%2F%3Fquery%3Dhttps3A2F2Ftwitter.com2Fprf_pb2Fstat2Fus2F1471339049898201088widget%3DTweet> . Acesso em: 05 de fev. de 2022.

3.1.3 Coleta de tweets

Após escolher os perfis e as palavras chaves relacionadas a acidentes de trânsito, foram realizadas requisições a API do Twitter através da biblioteca tweepy 4.5.0 [9].

Para ter acesso a API do Twitter, foi necessário cadastrar uma conta de usuário na plataforma e solicitar os tokens para serem utilizados durante as requisições como forma de autorização. Os dados retornados pela API contemplam: número de identificação, usuário, data, local da publicação, texto, curtidas, dentre outras informações.

3.2 Preparação dos Dados

3.2.1 Rotulagem dos dados

Foi realizada a rotulagem manual dos tweets da seguinte forma: classe positiva quando em seu conteúdo estava presente informações sobre acidentes de trânsito incluindo o local onde o incidente ocorreu; classe negativa quando o tweet não apresentava dados de acidentes de trânsito.

Os *corpus* resultante tem uma estrutura semelhante a da Tabela 1.

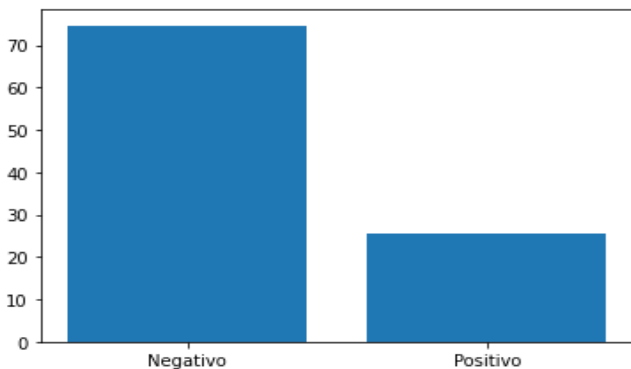
Tabela 1: Exemplo de classificação de tweet

tweet	classificação
corredor dom pedro tem trechos com chuva nesta tarde motoristas devem dobrar a atencao obras que eram realizadas nesta sexta ja foram encerradas nao ha acidentes e transito flui bem	NEGATIVO
atencao br 116 km 643 jequeie rodovia parcialmente interditada devido acidente reduza a velocidade	POSITIVO

Fonte: Autoria própria, 2022.

Ao final do processo de rotulagem dos dados, foram obtidos 3.311 tweets, porém o conjunto de dados ficou bastante desbalanceado, a classe negativa ocupou cerca de 75% dos registros, conforme pode ser visto na figura 3.

Figura 3: Distribuição dos tweets rotulados



Fonte: Autoria própria, 2022.

3.2.2 Pré-Processamento dos dados

Os dados coletados foram submetidos a várias técnicas de pré-processamento, que são descritas nesta subseção. A primeira técnica diz respeito à capitalização e limpeza dos dados, isto é, a normalização do texto em uma só tipografia, caixa alta ou baixa, e remoção de informações não relevantes, *links*, *emoticons*, e sinais de pontuação dos tweets. Para tanto, foram utilizadas expressões regulares e a biblioteca *re* da linguagem Python.

Outra técnica de pré-processamento utilizada diz respeito ao balanceamento do corpus para evitar possível enviesamento nos modelos de classificação a serem utilizados. Utilizou-se da técnica estatística de subamostragem, com o auxílio do algoritmo *RandomUnderSampler* da biblioteca *imblearn* para remover instâncias da classe negativa de forma aleatória [10]. Dessa forma, o conjunto balanceado resultou em 832 instâncias para cada classe.

Por fim, também foram aplicadas as técnicas de pré-processamento: remoção de stopwords e tokenização, fazendo uso da biblioteca *nlTK* [11] e do algoritmo *TfidfVectorizer* da biblioteca *sklearn* [12], respectivamente.

3.3 Indução dos modelos

Nesta subseção, foram realizadas etapas de indução dos modelos para identificar qual o melhor modelo para o problema de classificação de acidentes a partir de tweets.

3.3.1 Criação dos modelos e ajuste de hiperparâmetros

Para obter o melhor o modelo de classificação supervisionada para o problema aqui proposto, foram utilizadas as seguintes técnicas: Naive Bayes, Regressão Logística, XGBoost, Support Vector Machine (SVM), Random Forest e o transformer BERT. Para os cinco primeiros foi utilizada a biblioteca *sklearn*. Para o BERT, foi utilizado o BERTimbau disponível na biblioteca *transformers* [13].

O conjunto de dados foi dividido em treino e teste, usando o algoritmo *train_test_split* da biblioteca *sklearn*. Dessa forma, 20% dos dados foram destinados ao conjunto de testes, enquanto que 80% destinados para treinamento, sendo que 10% deste conjunto de treinamento foram utilizados durante o treinamento como conjunto de validação. Para reduzir as chances dos modelos sofrerem de *overfitting*, foi utilizado o método de validação cruzada *k-folds* com o parâmetro $k = 10$ através do algoritmo *StratifiedKFold* disponibilizado pelo *sklearn*.

Para um melhor ajuste dos hiperparâmetros dos modelos criados durante o processo de treinamento, foi utilizado o algoritmo *GridSearchCV* do *sklearn*. As métricas utilizadas para avaliação dos modelos foram acurácia, *f1-score*, precisão e *recall*. Conforme as tabelas 2, 3, 4, 5 e 6 foram testadas diferentes combinações de hiperparâmetros a fim de encontrar o modelo que melhor maximizou as métricas de acurácia e *f1-score* [14]. Todo o código utilizado para o treinamento dos modelos podem ser consultados através do *google colaboratory*³.

³ Treinamento de modelos do *sklearn*. Disponível em: <https://colab.research.google.com/drive/1bSWqT7AyHK5Za prG7m_kf0fPkhcypPN_?usp=sharing>.

Tabela 2: Combinações de hiperparâmetros para o modelo Naive Bayes.

MultinomialNB	
hiperparâmetro	valores (melhor valor)
alpha	0.01, 0.012, 0.015, 0.019 , 0.02, 0.5, 1
fit_prior	False, True , None

Fonte: Autoria própria, 2022.

Tabela 3: Combinações de hiperparâmetros para o modelo de Regressão logística.

LogisticRegression	
hiperparâmetro	valores (melhor valor)
penalty	l1, l2
tol	0.1, 0.2 , 0.3, 0.4, 0.5
C	0.1, 0.2, 0.3, 0.4 , 0.5
max_iter	35 , 40, 45

Fonte: Autoria própria, 2022.

Tabela 4: Combinações de hiperparâmetros para o modelo de Boosting.

XGBoost	
hiperparâmetro	valores (melhor valor)
gamma	1, 1.5, 2, 2.5
max_depth	1, 2, 3
learning_rate	0.1 , 0.2, 0.3, 0.4
n_estimators	50 , 55, 60

Fonte: Autoria própria, 2022.

Tabela 5: Combinações de hiperparâmetros para o modelo de SVM.

LinearSVC	
hiperparâmetro	valores (melhor valor)
C	0.5, 1, 1.5 , 2
max_iter	50 , 100, 500, 1000, 1200
tol	0.01 , 0.001, 0.002, 0.0001

Fonte: Autoria própria, 2022.

Tabela 6: Combinações de hiperparâmetros para o modelo de árvore de decisão.

RandomForestClassifier	
hiperparâmetro	valores (melhor valor)
max_leaf_nodes	5, 10, 20
max_depth	5 , 10, 20
criterion	gini , entropy
n_estimators	50, 100 , 150

Fonte: Autoria própria, 2022.

3.3.2 Modelo Transformers

O BERTimbau é uma rede neural pré-treinada para tratar problemas relacionados à língua portuguesa desenvolvido pela NeuralMind e disponibilizado pela Hugging Face [15]. Para indução do modelo, foram utilizados tensores pytorch [16] durante 5 épocas, número recomendado pela literatura, utilizando o mesmo conjunto de treinamento dos modelos do sklearn. O código utilizado pode ser acessado através do *google colaboratory*⁴.

3.4 Avaliação do Modelos

Nesta subseção são apresentados os resultados dos modelos de classificação utilizados e feita uma análise para a escolha do melhor modelo de indução para o problema proposto.

Os modelos de indução testados, que visam realizar a inferência de conhecimento a partir dos dados, foram divididos em duas subseções. Primeiramente, são abordados os modelos extraídos a partir do Sklearn. Por fim, o modelo baseado em rede neural profunda BERTimbau será discutido.

3.4.1 Modelos do Sklearn

Após encontrar os melhores hiperparâmetros de alguns modelos fornecidos pelo sklearn, foram encontradas as métricas listadas nas tabelas 7, 8, 9, 10 e 11 baseadas no conjunto de testes.

Tabela 7: Métricas do modelo de Naive Bayes.

MultinomialNB	
métrica	valor
acurácia	92%
precisão	92%
recall	91%
f1-score	92%

Fonte: Autoria própria, 2022.

⁴ Treinamento do BERTimbau. Disponível em: <<https://colab.research.google.com/drive/1h7RziptLVCBSaJpii3ZbVH2N7Q6n33?usp=sharing>>.

Tabela 8: Métricas do modelo de Regressão logística.

LogisticRegression	
métrica	valor
acurácia	93%
precisão	93%
recall	93%
f1-score	93%

Fonte: Autoria própria, 2022.

Tabela 9: Métricas do modelo de Boosting.

XGBoost	
métrica	valor
acurácia	95%
precisão	95%
recall	95%
f1-score	95%

Fonte: Autoria própria, 2022.

Tabela 10: Métricas do modelo de SVM.

LinearSVC	
métrica	valor
acurácia	94%
precisão	94%
recall	94%
f1-score	94%

Fonte: Autoria própria, 2022.

Tabela 11: Métricas do modelo de árvore de decisão.

RandomForestClassifier	
métrica	valor
acurácia	94%
precisão	94%
recall	94%
f1-score	94%

Fonte: Autoria própria, 2022.

Com base nas métricas obtidas, podemos concluir que o melhor é assumir que todos os cinco modelos podem ser utilizados para a

classificação de tweets em classe positiva e negativa. Embora o XGBoost apresente melhores resultados em comparação com os outros modelos, é importante destacar que seu processo de treinamento é bastante lento por ser mais complexo.

3.4.2 BERTimbau

Apesar de termos encontrado bons modelos provindo do sklearn, ao treinar o modelo pré-treinado BERTimbau os resultados foram ainda mais satisfatórios. Todas as métricas calculadas com o mesmo conjunto de testes dos modelos anteriores foram superadas, conforme a tabela 12 e a matriz de confusão da figura 4, tornando-o o melhor modelo para a construção do sistema de classificação de tweets.

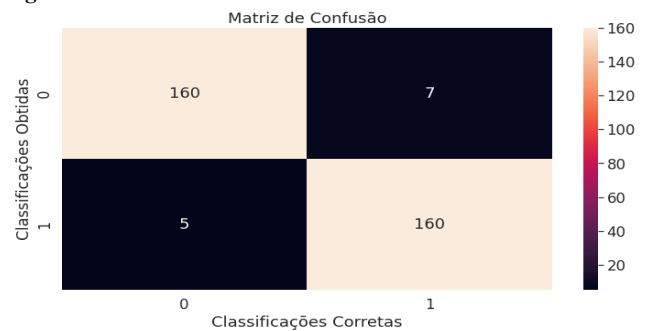
Para verificar a ausência de *overfitting*, foi gerado o gráfico da figura 5, representando o comportamento do erro durante o treinamento com os dados de treino e validação. Dessa forma, podemos observar que o erro com dados de validação vai caindo à medida que o de treinamento também cai. Logo, podemos concluir que o modelo está generalizando bem.

Tabela 12: Métricas do modelo por transformers.

BERTimbau	
métrica	valor
acurácia	96%
precisão	96%
recall	96%
f1-score	96%

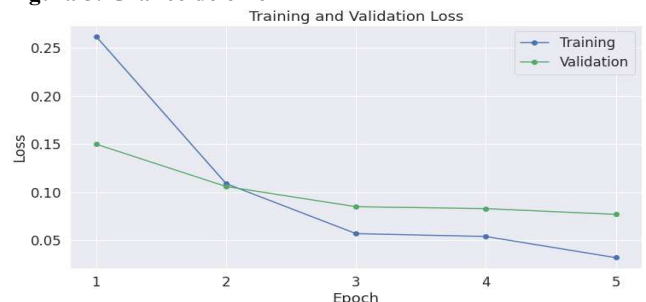
Fonte: Autoria própria, 2022.

Figura 4: Matriz de confusão



Fonte: Autoria própria, 2022.

Figura 5: Gráfico do erro



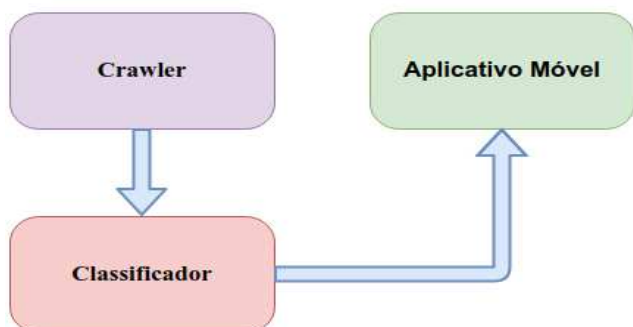
Fonte: Autoria própria, 2022.

3.5 Deployment: Sistema de Alerta de Acidentes

A infraestrutura do sistema de alerta de acidentes foi desenvolvida com o auxílio da linguagem de programação Python 3.10.2 [17] e com o banco de dados Postgresql 9.6 [18] para gestão dos dados. Além disso, foi utilizada a extensão Postgis 3.2, que é responsável pela manipulação de dados espaciais no Postgresql [19].

A arquitetura do SafeTrip é composta por três componentes principais demonstrados na figura 6, são eles o crawler, o classificador e o aplicativo móvel.

Figura 6: Componentes principais.

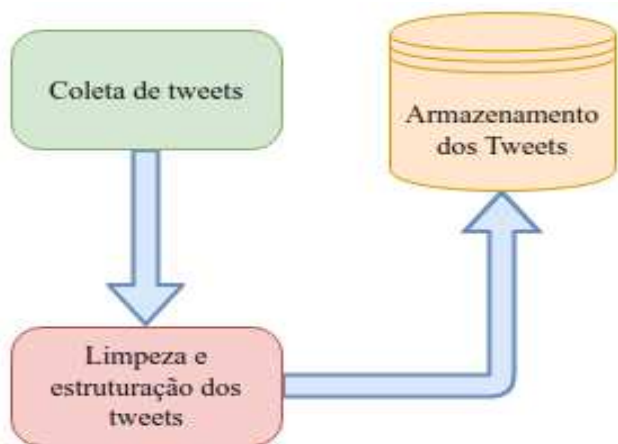


Fonte: Autoria própria, 2022.

3.5.1 Crawler

O crawler é a porta de entrada para o sistema. Com o auxílio da API tweepy, foram coletados tweets periodicamente, realizando o pré-processamento dos dados e armazenamento das informações no banco de dados, conforme a figura 7.

Figura 7: Estrutura do crawler

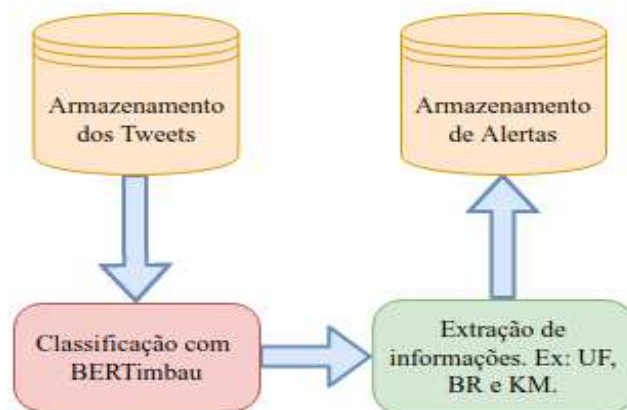


Fonte: Autoria própria, 2022.

3.5.2 Classificador

O módulo classificador, figura 8, é responsável por recuperar o tweet armazenado pelo crawler, aplicar tokenização e normalização dos dados para extrair a classe do tweet de acordo com o modelo de classificação escolhido. Quando os tweets são classificados como positivos, é aplicado um script baseado em expressão regular para a extração de informações como unidade federativa, código e o quilômetro da rodovia para permitir a geolocalização do acidente. Ao final, o tweet gera um alerta contendo informações de latitude, longitude, horário de ocorrência e próprio texto que é armazenado no banco de dados.

Figura 8: Estrutura do módulo classificador.



Fonte: Autoria própria, 2022.

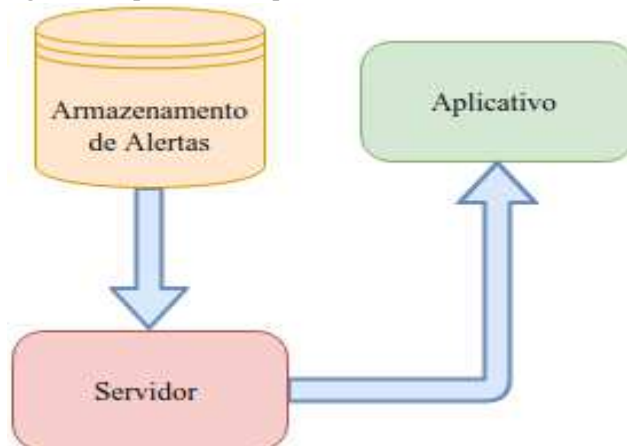
3.5.3 Aplicativo Móvel

Com o intuito de validar os resultados obtidos e colocá-los acessíveis à comunidade, foi desenvolvido um aplicativo móvel para notificar por áudio os motoristas acerca de acidentes detectados no Twitter. Este aplicativo móvel foi desenvolvido com base na arquitetura cliente-servidor conforme a figura 9. Segundo Vaskevitch [20], nessa arquitetura, ocorre a separação dos processos em plataformas independentes, permitindo a comunicação entre os processos enquanto se obtém o máximo de benefício de cada dispositivo distinto.

O servidor foi desenvolvido utilizando a linguagem de programação Node 15.14.0 [21]. O módulo é responsável pela comunicação, monitoramento e envio de alertas para os clientes através de *websocket* [22]. Além disso, são realizadas requisições à *Google directions API* [23], que é responsável pelo fornecimento de rotas de uma origem até um destino.

Com o auxílio do React Native, que é uma biblioteca baseada em *javascript* e criada pela Meta⁵ para a criação de códigos nativos de aplicativos Android e IOS [24], foi desenvolvido um aplicativo móvel para atuar como cliente, fornecendo dados de geolocalização e emitindo alertas, quando encontrados.

Figura 9: arquitetura do aplicativo móvel



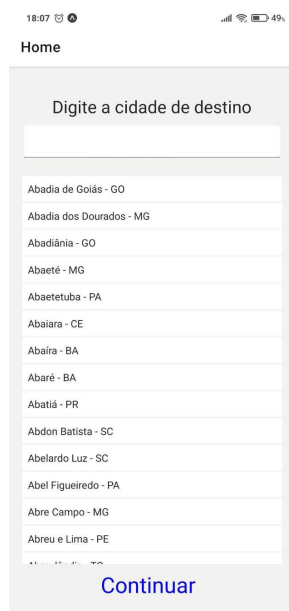
Fonte: Autoria própria, 2022.

⁵ Meta Platforms. Disponível em: <<https://about.facebook.com/br/meta/>>. Acesso em 22 de fev. de 2022.

O aplicativo desenvolvido atua com o papel de cliente da arquitetura construída. São necessárias permissões de acesso a internet e também ao sistema de localização do aparelho em que ele se encontra instalado.

Em sua tela inicial o usuário pode escolher uma entre todas as cidades brasileiras como a cidade de destino, conforme a figura 10.

Figura 10: Tela Inicial

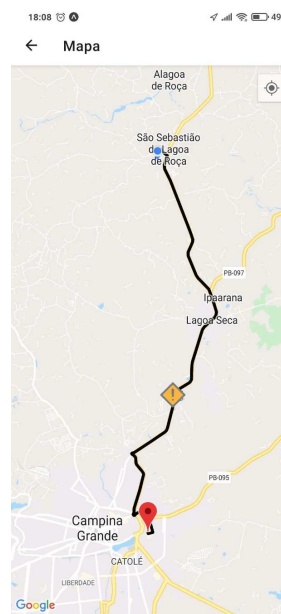


Fonte: Autoria própria, 2022.

O trajeto entre a localização atual do usuário e o destino, figura 11, será exibido após a cidade ser selecionada e o botão “Continuar” ser pressionado.

Ao detectar um alerta de acidente presente no trajeto do usuário, o servidor notifica o aplicativo que representa essa informação através de um ícone em forma de exclamação conforme a figura 11. Além disso, o conteúdo textual do tweet que gerou o alerta é emitido em forma de áudio.

Figura 11: Trajeto



Fonte: Autoria própria, 2022.

4. EXPERIÊNCIA

Nesta seção será discutido a experiência adquirida ao longo do processo de desenvolvimento do SafeTrip, bem como os principais desafios encontrados, as limitações do sistema e os trabalhos futuros propostos.

4.1 Processo de Desenvolvimento

Inicialmente, foram analisadas algumas tecnologias a serem utilizadas no desenvolvimento do sistema. Essa escolha levou em consideração tanto os trabalhos realizados ao longo da graduação quanto a alta popularidade das tecnologias no mercado. Em seguida, foram definidas as principais funcionalidades da aplicação a serem desenvolvidas seguindo um processo de desenvolvimento ágil baseado no Scrum, mesmo não sendo aplicado totalmente por envolver apenas um desenvolvedor, mas com entregas e planejamentos pré definidos semanalmente.

4.2 Principais Desafios

Um dos principais desafios foi a rotulagem manual do conjunto de dados para o treinamento dos modelos de classificação. Embora a API tweepy tenha ajudado bastante na coleta, foi necessário ler todos os tweets e classificar um a um de maneira consciente.

Outro grande desafio foi o de buscar entender a função de cada hiperparâmetro nos modelos de classificação e escolher aqueles que trouxessem melhores métricas a partir do ajuste de seus valores.

4.3 Limitações

A maior limitação encontrada foi a impossibilidade de incluir a compatibilidade do SafeTrip com o sistema IOS. Embora a tecnologia escolhida permita a geração de sistemas multiplataformas, é necessário possuir um Macbook da Apple para a criação de aplicativos IOS. Sendo assim, o SafeTrip garante a compatibilidade apenas para dispositivos que tenham o sistema operacional Android.

4.4 Trabalhos Futuros

Após finalizar este trabalho, notamos que melhorias podem ser realizadas para obtenção de melhores resultados, dentre essas melhorias podemos listar: (i) aumento do corpus para um melhor processo de treinamento, (ii) verificar se a rotulagem dos dados pode ser melhorada para tornar os modelos mais confiáveis. (iii) implementar alguma estrutura de prioridade no sistema de alerta para garantir uma melhor performance do sistema em relação a comunicação entre aplicativo e servidor e (iv) realizar testes de usabilidade do aplicativo. Além disso, com modelos confiáveis de classificação de tweets, é possível contribuir com a atualização da base de dados da PRF referentes ao histórico de acidentes.

5. AGRADECIMENTOS

Agradeço primeiramente a Deus por permitir o desenvolvimento deste trabalho. Aos meus pais José Erivaldo e Gerlane Pereira e a minha companheira Jéssica Gregório pelo apoio durante minha caminhada. Ao professor Cláudio Baptista pela contribuição e auxílio incomparável. Aos meus amigos, em especial a Hugo Feitosa, Anderson Almeida, Damião R. de Almeida, Paulo Mendes, Hércules Anselmo, Rich Elton, Igor Silveira e Vinicius Barbosa, que juntamente com todos os professores contribuíram para o crescimento do meu conhecimento.

Agradeço também ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) por financiar este projeto, processo número: 304853/2020-8.

6. REFERÊNCIAS

[1] LEAL, Ceci. TOP 5 - Maiores malhas rodoviárias do mundo. **Made for minds**, 17 de nov. de 2012. Disponível em: <<https://p.dw.com/p/16kHn>>. Acesso em: 01 de set. de 2021.

[2] MALHA rodoviária total. **Anuário CNT do Transporte**, 2018. Disponível em: <<https://anuariodotransporte.cnt.org.br/2018/Rodoviario/1-3-1-1-1-/Malha-rodovi%C3%A1ria-total.>>Acesso em: 01 de set. de 2021.

[3] RESENDE, Paulo Tarso Vilela *et al.* Custos Logísticos no Brasil. **Fundação Dom Cabral**, 2017. Disponível em: <[https://ci.fdc.org.br/AcervoDigital/Relat%C3%B3rios%20de%20Pesquisa/Relat%C3%B3rios%20de%20pesquisa%202018/Apresenta%C3%A7%C3%A3o%20de%20Custos_Logisticos_no%20Brasil%202018_FDC%20_%20revRVC%20abr18%20\(002\).pdf](https://ci.fdc.org.br/AcervoDigital/Relat%C3%B3rios%20de%20Pesquisa/Relat%C3%B3rios%20de%20pesquisa%202018/Apresenta%C3%A7%C3%A3o%20de%20Custos_Logisticos_no%20Brasil%202018_FDC%20_%20revRVC%20abr18%20(002).pdf)>. Acesso em: 04 de set. de 2021.

[4] PAINEL de acidentes rodoviários. **Confederação Nacional do Transporte**, 2020. Disponível em: <<https://www.cnt.org.br/painel-acidente>>. Acesso em: 01 de set. de 2021.

[5] SOBRE o Waze. **Google**, c2022. Disponível em: <https://support.google.com/waze/answer/6071177?hl=pt-BR&ref_topic=9022747>. Acesso em: 18 de mar. de 2022.

[6] BRASIL. Código de Trânsito Brasileiro (CTB). Lei Nº 9.503, de 23 de set. de 1997 que institui o Código de Trânsito Brasileiro. **Presidência da República**, s.d. Casa Civil. Subchefia para Assuntos Jurídicos. Brasília. Disponível em : <http://www.planalto.gov.br/ccivil_03/Leis/L9503.htm>. Acesso em: 01 de set. de 2021.

[7] SOUZA, Uendel. O uso indevido do celular no trânsito. **Detran - TO**, 12 de mar. de 2020. Disponível em:

<<https://www.to.gov.br/detran/noticias/o-uso-indevido-do-celular-no-transito/4fi7cjfe84h>>. Acesso em: 02 de set. de 2021.

[8] CROSS Industry Standard Process for Data Mining (Crisp-DM). **Wayback Machine**, 05 de nov de 2005. Disponível em: <<https://web.archive.org/web/20051105013719/http://www.crisp-dm.org/index.htm>>. Acesso em: 04 de set de 2021.

[9] ROESSLEIN, Joshua. Tweepy Documentation. **Tweepy**, c2009-2022. Disponível em: <<https://docs.tweepy.org/en/stable/>>. Acesso em: 04 de fev. de 2022.

[10] USER Guide. **The imbalanced learn developers**, c2014-2022. Disponível em: <https://imbalanced-learn.org/stable/user_guide.html>. Acesso em: 10 de fev. de 2022.

[11] NATURAL Language Toolkit. **NLTK Project**, c2022. Disponível em: <<https://www.nltk.org/>>. Acesso em: 10 de fev. de 2022.

[12] USER Guide. **Scikit-learn developers**, 2007-2021. Disponível em: <https://scikit-learn.org/stable/user_guide.html>. Acesso em: 10 de fev. de 2022.

[13] SOUZA, Fabio; NOGUEIRA, Rodrigo; LOTUFO, Roberto. **BERTimbau**: Pretrained BERT Models for Brazilian Portuguese. In: Cerri R., Prati R.C. (eds) Intelligent Systems. BRACIS 2020. Lecture Notes in Computer Science, vol 12319. Springer, Disponível em: <https://doi.org/10.1007/978-3-030-61377-8_28>. Acesso em 21 de fev. de 2022.

[14] PORTOROŽ, Slovenia. **Complementarity, F-score, and NLP Evaluation**. Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 1 (2016), 261–266.

[15] JULIANA. BERTimbau, da NeuralMind, é recordista em downloads. **Neuralmind**, 29 de nov. de 2020. Disponível em: <<https://neuralmind.ai/2020/11/29/bertimbau-da-neuralmind-e-recordista-em-downloads/>>. Acesso em: 17 de fev. de 2022.

[16] PYTORCH Documentation. **Torch Contributors**, c2019. Disponível em: <<https://pytorch.org/docs/stable/index.html>>. Acesso em 15 de fev. de 2022.

[17] PYTHON Documentation. **Python Software Foundation**, 01 de fev. de 2022. Disponível em: <<https://docs.python.org/3/>>. Acesso em 01 de fev. de 2022.

[18] POSTGRESQL 9.6.24 Documentation. **PostgreSQL Global Development Group**, c1996-2021. Disponível em: <<https://www.postgresql.org/docs/9.6/index.html>>. Acesso em: 01 de fev. de 2022.

[19] POSTGIS 3.2.1dev Manual. **PostGIS**, 02 de fev. de 2022. Disponível em: <<https://postgis.net/docs/manual-3.2/>> . Acesso em: 03 de fev. de 2022.

[20] VASKEVITCH, David. **Client/Server strategies : a survival guide for corporate reengineers**. Foster City, IDG Books Worldwide, 1995. p. 375.

[21] NODE v15.14.0 documentation. **Node.js**, s.d. Disponível em: <<https://nodejs.org/docs/latest-v15.x/api/documentation.html>>. Acesso em 22 de fev. de 2022.

[22] WEBSOCKETS Documentação, **MDN contributors**, 21 de fev. de 2022. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/API/WebSockets_API>. Acesso em: 22 de fev. de 2022.

[23] DIRECTIONS api, **Google Developers**, s.d. Disponível em: <<https://developers.google.com/maps/documentation/directions>>. Acesso em 22 de fev. de 2022.

[24] REACT Native, **Meta Platforms Inc**, c2022. Disponível em: <<https://reactnative.dev/>>. Acesso em 22 de fev. de 2022.