http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2022.v14.n1. e380
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 17/12/2021 Revisado: 31/01/2022 Aceito: 22/03/2022

APRENDIZADO DE MÁQUINA UTILIZANDO AGRUPAMENTO E REGRESSÃO NA PREVISÃO DE LOCAIS DE ACIDENTES DE TRÂNSITO EM ZONAS URBANAS

MACHINE LEARNING USING CLUSTERING AND REGRESSION IN PREDICTING TRAFFIC ACCIDENT SITES IN URBAN ZONES

Caio Kraut, Helton Molina Sapia

Universidade do Oeste Paulista – UNOESTE, Faculdade de Informática de Presidente Prudente, Presidente Prudente, SP.
E-mail: caiomarin26@gmail.com; helton@unoeste.br

RESUMO – Com a urbanização das cidades brasileiras, a locomoção automobilística se tornou algo indispensável, assim a área de mobilidade urbana aumentou em escala exponencial, acarretando em um crescimento da violência no trânsito, seja causada por engarrafamentos, problemas de viés humano ou de infraestrutura. Esse trabalho propõe uma solução que prevê locais de acidentes dentro de zonas urbanas com base em dados temporais (data e hora) de acidentes. Utiliza o algoritmo *K-Means* para agrupar e *KNN Regressor* para prever, dentro da amostra de dados de acidentes da cidade de São Paulo coletados entre 2019 e 2021, obteve-se um modelo preditivo com precisão de 96.04% dentro de uma tolerância de 500m.
Palavras-chave: K-Means; KNN; Geolocalização; Acidentes de Trânsito; Euclidiana; Haversine.

ABSTRACT – With the urbanization of Brazilian cities, automobile locomotion has become indispensable, so the area of urban mobility has increased on an exponential scale, resulting in an increase in traffic violence, whether caused by traffic jams, human bias or infrastructure problems. This work proposes a solution that predicts accident locations within urban areas based on temporal data (date and time) of accidents. It uses the K-Means algorithm to group and KNN Regressor to predict, within the sample of accident data from the city of São Paulo collected between 2019 and 2021, a predictive model with an accuracy of 96.04% within a tolerance of 500m was obtained.
Keywords: K-Means; KNN; Geolocation; Traffic Accidents; Euclidean; Haversine.

1. INTRODUÇÃO

A urbanização do território brasileiro, apesar de ser algo considerado positivo para a população, também trouxe muitas desvantagens, e uma delas é a mobilidade urbana. Problemas como engarrafamento e acidentes automobilísticos (Violência no Trânsito) cresceram em uma escala exponencial.

Segundo Saragiotto (2020), a cada 15 minutos uma pessoa é vítima de um acidente de trânsito com fatalidade, seja a vítima, motorista ou pedestre. O Brasil está em quinto lugar entre os países com mais mortes no trânsito, atrás

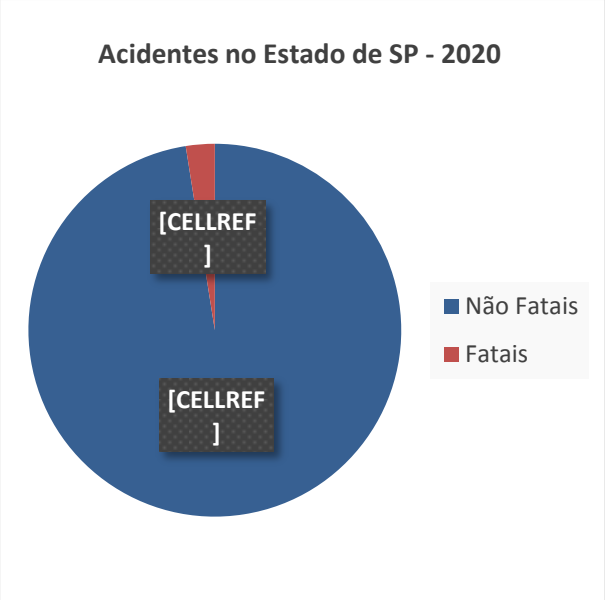
apenas da Índia, China, EUA e Rússia (BSTM - Brazilian Society Of Tropycal Medicine, 2019).

Segundo o DATASUS (2020), acidentes de trânsito foram a causa de aproximadamente 23% das mortes por causas externas no Brasil, resultando em um pouco mais de 32 mil mortos.

Considerando não só os acidentes com fatalidade, mas também os não fatais, a escala se tornou muito maior.

Observando a figura 1, os dados coletados de acidentes ocorridos em 2020 pela secretária do estado de São Paulo (SÃO PAULO, 2021) mostrou a diferença acachapante entre o número de casos de acidentes fatais e não fatais no estado de São Paulo.

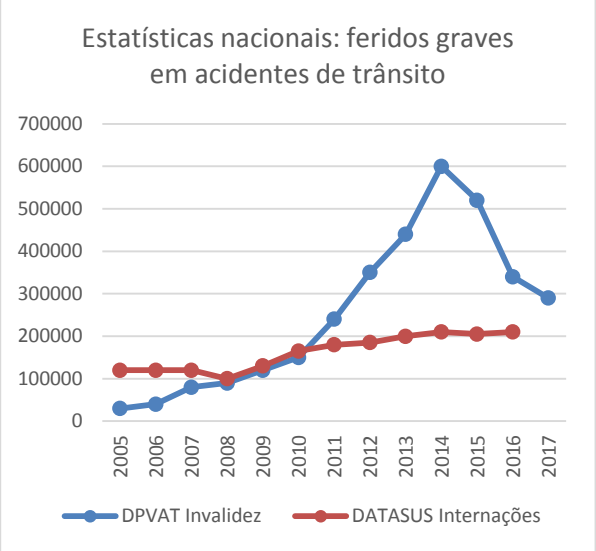
Figura 1. Gráfico de acidentes de trânsito fatais e não fatais no Estado de SP (2020)



Fonte: O autor.

Como pode ser visto na figura 2, o número de internações por ferimento graves causados por acidentes de trânsito chegou a aproximadamente 200 mil em 2016, além disso o DPVAT registrou em 2017 quase 300 mil casos de invalidez causados por acidentes de trânsito.

Figura 2. Gráfico de feridos graves em acidente de trânsito



Fonte: Adaptado de (Vias Seguras, 2020)

Segundo a pesquisa feita pela EcoRodovias (2020), pelo menos 90% dos acidentes de trânsito poderiam ser evitados, visto que os mesmos foram causados por falhas humanas (falta de atenção, embriaguez, ultrapassar o limite de velocidade, etc.).

Usando como contexto as principais causas de acidentes por falhas humanas: falta de atenção (55%), velocidade incompatível (8%) e desobediência à sinalização (5%) foi possível traçar um padrão dos locais onde ocorreram a maior quantidade de acidentes, agrupá-los, e executar cálculos estatísticos para prever os locais de acidentes (latitude e longitude) com base em quando os acidentes ocorreram.

O objetivo dessa pesquisa foi coletar dados de acidentes de trânsito com os atributos: latitude, longitude, data e hora, agrupar os dados com o algoritmo *K-Means* e separar cada grupo como uma amostra e executar o algoritmo de regressão *KNN* para traçar uma função que consiga prever os locais de acidentes, analisar os dados previsto, comparar com os dados reais, analisar funções estatísticas de acertos de previsão e relatar os resultados obtidos a fim de propor uma solução que diminua a quantidade de acidentes dentro do contexto urbano.

1.1. Trabalhos Relacionados

O trabalho de Silva (2020), descreve uma utilização de algoritmos de regressão para prever locais de crimes na cidade de Fortaleza – CE, com base na relação de atributos de tempo para prever latitude e longitude. Em um dos experimentos usados, na qual o autor considerou

latitude e longitude como um atributo único, o *KNN* apresentou um resultado superior aos outros métodos.

O trabalho de Silva (2019), descreve uma ferramenta de apoio a decisão para melhorar as rotas de transporte público da cidade de Braga (Portugal) usando métodos de regressão. Na análise dos resultados dos algoritmos, o *KNN* apresentou o melhor valor do R^2 (coeficiente de determinação) que mostrou o quão boa a regressão se adaptou aos dados originais.

O trabalho de Syakur *et al*. (2021), descreve uma proposta que usou o método do *K-Means* para agrupar dados de clientes com o intuito de identificar questões de fidelidade de clientes utilizando o método do cotovelo. O trabalho concluiu que o método do cotovelo mostrou o resultado padrão para processos característicos com base em estudo de caso.

1.2. Organização do trabalho

Este trabalho foi organizado da seguinte forma: no Capítulo 2, o desenvolvimento do projeto foi descrito com suas funcionalidades, recursos utilizados, metodologia com seus métodos e procedimentos, as métricas e as etapas da implementação do projeto.

No Capítulo 3, foram apresentados os resultados obtidos a partir dos métodos implementados, como os resultados podem ser interpretados e a discussão sobre os mesmos.

Apresentou-se no Capítulo 4 as considerações finais do projeto que traz a conclusão do mesmo, e perspectivas futuras em relação ao problema explorado.

2. DESENVOLVIMENTO

Para desenvolvimento da pesquisa foi implementado um software web com o nome de Streetor. A ferramenta tem como principal função exibir um mapa de uma cidade focando nos locais de acidentes previstos, baseado no dia da semana e no período selecionado (manhã, tarde, noite ou madrugada).

A aplicação lê uma base de dados com informações sobre acidentes já ocorridos, filtrando a cidade, o dia da semana e o período do dia que está sendo analisado.

Após a leitura dos dados, se executa o procedimento e métodos descritos na metodologia, e coleta os resultados da predição fazendo o cálculo das métricas para validar o funcionamento e a precisão dos algoritmos usados.

Além de visualizar as métricas e os atributos da predição, existe a opção de visualizar os gráficos de resíduos resultantes da predição.

É possível também usar a ferramenta para exibir um mapa do tipo “bubblemap” e “heatmap” contendo os locais de maior incidência de acidentes na cidade.

Por questões de facilidade do usuário, há uma opção para alterar as cores tema do mapa e dos pontos do mapa além de uma opção para mostrar junto aos dados preditos, os dados reais usados na predição. Também a funcionalidades para customização do algoritmo de predição, porém o sistema não obriga a utilização do mesmo, e já possui implementado um método para calcular de forma dinâmica todos esses atributos.

Também possui funcionalidades para customização dos dados que são utilizados na predição, como por exemplo, a cidade onde ocorrerá a predição, o dia da semana e o período do dia, além de poder filtrar somente acidentes com fatalidade, acidentes que ocorreram em ruas e a opção de exibir o endereço (nome da rua/avenida/rodovia e número) e por fim, selecionar o intervalo de tempo (em anos) dos dados.

2.1. Recursos

Para desenvolvimento da ferramenta foi utilizado a linguagem python, pela sua facilidade em manipulação e análise de dados. Junto ao python, foram utilizadas as bibliotecas:

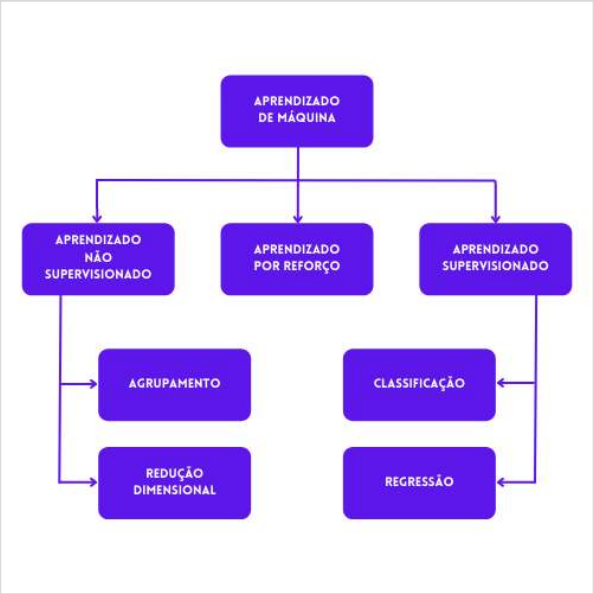
- **Streamlit**: ferramenta utilizada para criação de aplicação web focado em análise e processamento de dados, por utilizar a função de armazenamento de objetos cache.
- **Pandas**: biblioteca em python, utilizada para coleta, análise e processamento de dados.
- **Scikit Learn**: um conjunto de algoritmos de aprendizado de máquina para execução de IA.
- **Plotly**: ferramenta usada para exibição de gráficos em mapas.
- **Seaborn**: ferramenta usada para exibição e customização de gráficos.
- **MongoDB**: sistema de armazenamento de dados não relacional.

2.2. Metodologia

Aprendizado de Máquina é uma ferramenta de ciência de dados que está inserido também na área de inteligência artificial, e utiliza de lógicas estatísticas para "ensinar" a máquina a tomar decisões baseadas em modelos de treino utilizando dados pré-definidos.

Dentro do conceito de Aprendizado de Máquina, existem três tipos de aprendizado: Supervisionado, Não-Supervisionados e o Aprendizado por reforço, como pode ser observado na figura 3.

Figura 3. Hierarquia do aprendizado de máquina



Fonte: O autor.

O supervisionamento é feito quando a partir de um conjunto de dados rotulados, previamente definido, deseja-se encontrar uma função que seja capaz de predizer rótulos desconhecidos. O mesmo também é capaz de tomar decisões precisas quando recebe novos dados não rotulados a partir de um treinamento com dados que possuem rótulos conhecidos. No aprendizado não-supervisionado o conjunto de dados utilizado não possui nenhum tipo de rótulo. O objetivo desse tipo de aprendizagem é descobrir similaridades entre os objetos analisados a fim de detectar similaridades e anomalias (ISI-TICS, 2018).

No aprendizado por reforço, baseado na tentativa e erro. Nesse tipo de aprendizado, tem-se um ambiente dinâmico, e a máquina deve aprender como se comportar nesse ambiente a partir de interações com o mesmo. O agente executa uma ação no ambiente e o mesmo retorna um estado ao agente, assim reforçando o aprendizado das ações executadas.

2.2.1. Agrupamento

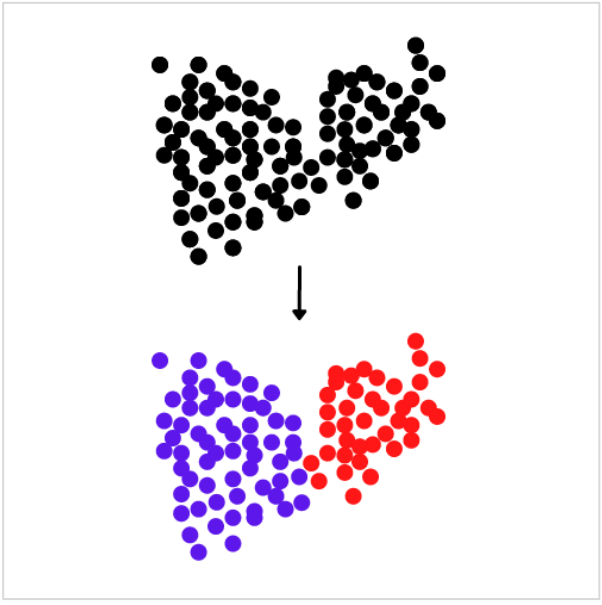
Dentro do aprendizado de máquina, mais especificamente, dentro do aprendizado não-supervisionado, a lógica de agrupamento (*clustering*) permite que os dados, sem necessidade de rótulos ou parâmetros, sejam agrupados em conjuntos (*clusters*) de dados que possuem algum atributo em comum.

Dabbura (2018) explicou que o agrupamento é uma das técnicas de análise de dados mais comum usada para obter intuição sobre a estrutura dos dados. Pode ser definido como a tarefa de identificar subgrupos nos dados de forma que os pontos de dados no mesmo subgrupo (*cluster*) sejam muito semelhantes, enquanto os pontos de dados em diferentes subgrupos são muito diferentes.

Ao contrário da aprendizagem supervisionada, o agrupamento é considerado um método de aprendizagem não supervisionado, uma vez que não se tem a verdade para comparar a saída do algoritmo de agrupamento aos rótulos verdadeiros para avaliar seu desempenho. Apenas tenta investigar a estrutura dos dados agrupando os pontos de dados em subgrupos distintos.

Como pode ser observado na figura 4, o agrupamento reorganiza os dados em subgrupos, e define uma característica (rótulo) para os dados para rotular uma "semelhança" entre os dados dentro do subgrupo.

Figura 4. Exemplo de agrupamento



Fonte: O autor.

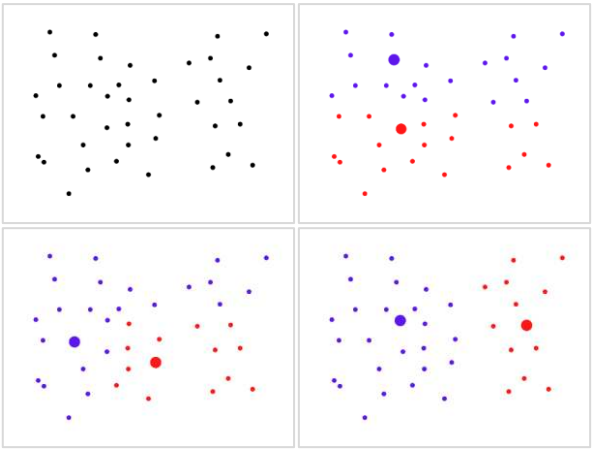
2.2.2. K-Means

O algoritmo *K-Means* é um algoritmo iterativo que tenta particionar o conjunto de dados em subgrupos distintos e não sobrepostos pré-definidos (*clusters*), onde cada ponto de dados pertence a apenas um grupo. Ele tenta tornar os pontos de dados de intergrupo tão semelhantes quanto possível, enquanto também mantém os grupos o mais diferente (o mais longe) possível. Ele atribui pontos de dados a um grupo de modo que a soma da distância quadrada entre os pontos de dados e o centroide do agrupamento (média aritmética de todos os pontos de dados que pertencem a esse grupo) seja mínima. Quanto menos variação tivermos dentro dos grupos, mais homogêneos (semelhantes) serão os pontos de dados dentro do mesmo grupo (DABBURA, 2018).

O algoritmo *K-Means* funciona da seguinte forma:

1. Definir o parâmetro K (o número de grupos que será formado na análise).
2. Embaralhe todo o conjunto de dados (para que não fique, de forma alguma, ordenado) e após isso selecione aleatoriamente uma quantidade K de dados, e defina-os como centroides.
3. Calcule a soma da distância quadrada entre os dados e todos os centroides.
4. Atribua dados ao grupo mais próximo (centroides).
5. Recalcule os centroides para os grupos tomando a média dos valores dos dados que pertencem a cada grupo.
6. Execute os passos 3, 4 e 5 até que não haja modificação nos valores dos centroides (como pode ser observado na figura 3). Geralmente é definido um valor n de interações, para limitar a quantidade de repetições (o número de interações pode mudar o agrupamento em função do valor do K e da quantidade de dados).

Figura 5. Etapas do K-Means



Fonte: O autor.

A representação matemática do K-Means é dada pela função representada na equação 1, onde a função principal é o somatório das interações dos centroides executando a função distância (no caso, a distância euclidiana).

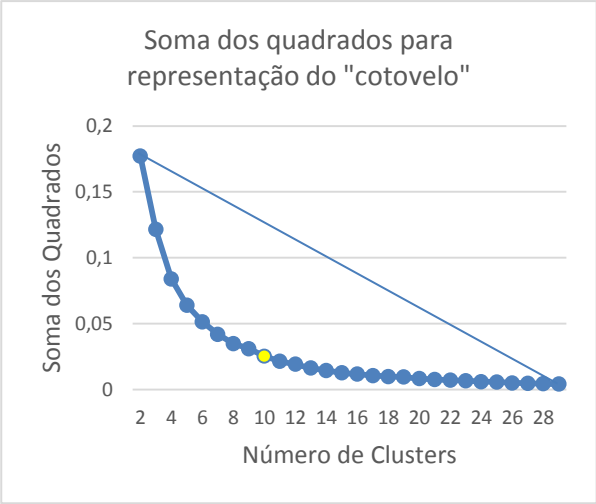
$$j = \sum_{k=1}^{K} \sum_{i=1}^{n} ||x_i^k - C_k||^2 \qquad (1)$$

Para calcular a melhor quantidade de clusters a ser usado como valor do K no algoritmo do *K-Means*, é necessário fazer análises de distorção dos grupos e compará-los.

Ao fazer comparação de agrupamentos usando como valores do K um intervalo entre n e m, em que n < m, e utilizar o atributo de inércia do *K-Means* (a soma dos quadrados da distância até o centroide mais próximo do dado em questão) para referenciar a distorção entre os modelos e exibir isso em gráfico de curva, observa-se o formato de um cotovelo nessa curva, o nome dessa prática é *Elbow Method* (ou método do cotovelo) que se consiste em gerar uma curva com as distorções de modelo do *K-Means* baseado em um intervalo de valores do K, e encontrar o valor do cluster com maior homogeneidade e maior diferença entre os clusters.

Como pode ser observado na Figura 6, para encontrar tal resultado é necessário traçar uma reta entre o primeiro e o último ponto da curva e calcular a distância entre a reta e cada ponto da curva, na qual o ponto com a maior distância entre a reta seria o melhor valor do K.

Figura 6. Exemplo do método do cotovelo



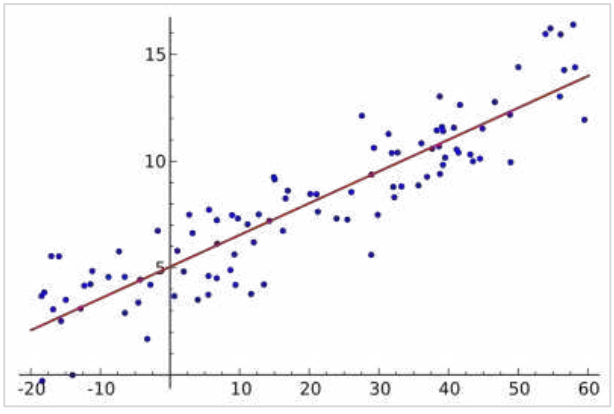
Fonte: O autor.

2.2.3. Regressão

Regressão é um tipo de aprendizado de máquina supervisionado que tem como função prever dados, a partir de um conjunto de dados pré-definido (amostra) e possui uma amostra de dados teste para verificar a autenticidade do modelo preditivo.

Na regressão, existem dois tipos de variáveis: dependentes e independentes. As independentes são as variáveis explicativas, ou seja, são elas que possuem uma correlação com a variável dependente. A variável dependente, tem seu valor totalmente dependente das variáveis explicativas. Em um modelo regressivo, estudasse um método de gerar uma função matemática que retorne o valor da variável dependente, em função das variáveis explicativas, como mostra a figura 7.

Figura 7. Exemplo de regressão linear



Fonte: (BONIN, 2019)

2.2.4. KNN (K-Nearest Neighbors)

O *KNN* é um algoritmo não paramétrico, onde a estrutura do modelo será determinada pelo *dataset* utilizado. Este algoritmo também é conhecido como de aprendizado lento ou preguiçoso (*lazy*).

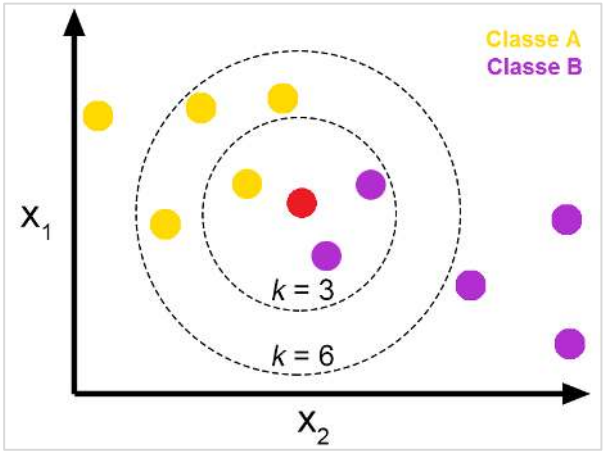
Os algoritmos do tipo *lazy*, não necessitam de dados de treinamento para se gerar o modelo, o que diminui em partes o processo inicial, mas em contrapartida gerará uma necessidade de análise posterior mais apurada.

No caso de algoritmos que não necessitam de treinamento, todos os dados obtidos no dataset serão utilizados na fase de teste, resultando em um treinamento muito rápido e em um teste e validação lentos (LUZ, 2019).

Neste algoritmo possui-se uma variável chamada de K, a qual é parte do nome do modelo e também o principal parâmetro a ser selecionado. Este parâmetro direcionará a quantidade de vizinhos (*neighbors*) (LUZ, 2019).

Usando um método de cálculo da distância, é possível criar classes/rótulos para os dados do modelo, como pode ser observado na figura 7.

Figura 7. Classificação com *KNN*

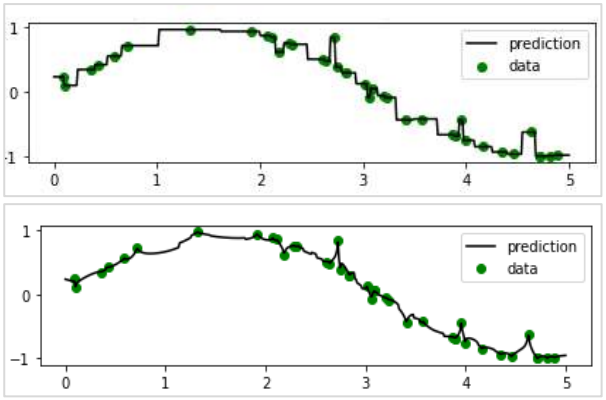


Fonte: (JOSÉ, 2019)

A *KNN* pode também ser usado para solução de problemas com regressão modificando o método para calcular o valor a ser associada a instância que está sendo testada.

Ao invés de usar a classe que aparece com mais frequência nos K vizinhos mais próximos, utiliza-se uma relação matemática (média, desvio-padrão, etc.) dos valores dessas instâncias, como pode ser visto na figura 8 (ROCHA, 2018).

Figura 8. Regressão com *KNN* (K=1 e K=5)



Fonte: O autor.

2.2.5. Métrica de Haversine

Para avaliar distância com valores de geolocalização (latitude e longitude) é necessário converter essa distância para algum sistema métrico padrão conhecido (Quilômetros, Milhas, Jardas, etc.).

Porém, com os valores definidos na latitude e longitude, a distância não pode ser calculada usando os métodos padrões euclidianos.

Para isso a fórmula de Haversine, publicada em 1805 por Florian Cajori (AN BRUMMELEN, 2013) é usada para calcular a "distância do grande círculo" entre 2 pontos contidos na superfície de uma esfera, usando funções trigonométricas para avaliar a curvatura da esfera nos pontos em questão. O cálculo da métrica de Haversine pode ser representado pela fórmula:

$$2r \arcsin\left(\sqrt{\begin{matrix} sin^2(\frac{\varphi_2-\varphi_1}{2}) + \\ (1 - sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right) - \\ sin^2\left(\frac{\varphi_2+\varphi_1}{2}\right)) \cdot \\ sin^2(\frac{\lambda_2-\lambda_1}{2}) \end{matrix}}\right) \quad (2)$$

2.3. Coleta e tratamento dos dados

Para análise e exploração dos dados, foi usado o *framework* OSEMN, que tem como objetivo 5 etapas de análise e exploração de dados: obter, limpar, explorar, modelar, interpretar.

Para etapa de obtenção dos dados, foi usado a base de dados de acidentes do estado de São Paulo coletados diretamente da secretária do estado de SP (SÃO PAULO, 2021), onde está disponível publicamente todos os registros de acidentes que ocorreram nos últimos anos com e sem fatalidade.

Os dados estavam disponíveis no formato de planilhas (Excel e CSV). A base de dados de acidentes fatais possuía 35.647 linhas e 32 colunas, já a base de dados de acidentes não fatais possuía 469.495 linhas e 50 colunas.

Para a etapa de limpagem dos dados, foi analisado dados cruciais ausentes (como data, hora e localização), dados duplicados, inconsistentes e dados errados de ambas as bases de dados. Após a filtragem desses dados, foi necessário adaptar alguns atributos da base de dados de acidentes não fatais para análise prática do mesmo.

Na etapa de exploração dos dados, foi selecionado as colunas de tipo de acidente (Tipo de Acidente – Atropelamento (Pedestre), Tipo de Acidente – Atropelamento (Animal), Tipo de Acidente – Choque, Tipo de Acidente – Colisão, Tipo de Acidente - Outros tipos de Acidente) e adaptados para uma única coluna chamada de Vítimas, onde indicaria a quantidade de vítimas do acidente.

Também foi selecionado as colunas de tipo do veículo (Veículos Envolvidos – Bicicleta, Veículos Envolvidos – Caminhão, Veículos Envolvidos – Automóvel, Veículos Envolvidos – Motocicleta, Veículos Envolvidos – Ônibus, Veículos Envolvidos - Pedestre) e adaptados para a coluna de nome Tipo de Veículo onde armazena o tipo de veículo causador do acidente.

Após as adaptações da base de dados, foi feito um cruzamento entre as duas bases de dados, a fim de que se torna uma única base consistente. Para isso foi feita a escolha das colunas que serão usadas para constituir a nova base de dados.

Na base de dados de acidentes não fatais, foi selecionado as colunas: Dia do Acidente, Mês do Acidente, Ano do Acidente, Hora do Acidente, LAT_(GEO), LONG_(GEO), Tipo do veículo, Vítimas, Tipo de via, Dia da semana, Período do dia, Município e Logradouro. Já na base de dados de acidentes fatais, foi escolhido as colunas: Dia do Acidente, Mês do acidente, Ano do acidente, Hora do acidente, Turno, Município, Lat (GEO), Long (GEO), Quantidade de vítimas, Tipo de via, Outro Veículo Envolvido, Logradouro e Dia da semana.

A colunas Dia do Acidente foi renomeada para dia, Mês do Acidente foi renomeada para mês e Ano do Acidente para ano. Hora do Acidente foi separado entre hora e minuto,

LAT_(GEO) e Lat (GEO) foram agrupados na mesma coluna de nome latitude, assim como LONG_(GEO) e Long (GEO) agrupados na coluna longitude. Tipo de veículo e Outros Veículos Envolvidos foram agrupados na coluna Veículos e Quantidade de Vítimas foi agrupado a coluna Vítimas. Dia da Semana foi renomeado para Semana e Tipo de Via para Via. Para diferenciar acidentes fatais e não fatais, foi criado o atributo fatal contendo os valores 1 ou 0 representando se o acidente foi fatal ou não, respectivamente. A finalização da nova base de dados pode ser observada na tabela 1.

Tabela 1. Atributos criados e renomeados das bases de dados

Acidentes Fatais	Acidentes Não Fatais	Nova Base de Dados
Dia do Acidente	Dia do Acidente	dia
Mês do Acidente	Mês do Acidente	mês
Hora do Acidente	Hora do Acidente	hora
		minuto
Lat (GEO)	LAT_(GEO)	latitude
Long (GEO)	LONG_(GEO)	longitude
Outro Veículo Envolvido	Tipo do Veículo	veículo
Tipo de via	Tipo de Via	via
Dia da semana	Dia da Semana	semana
Turno	Período do dia	turno
Município	Município	município
Logradouro	Logradouro	logradouro

Fonte: Elaborado pelo Autor.

Os dados coletados, tratados e adaptados para utilização no modelo, foram armazenados em um banco de dados não-relacional com MongoDB, para tornar mais prático os filtros de dados do mesmo.

2.4. Modelo

Após a leitura dos dados filtrado pelo dia da semana e período indicado na predição, iniciou-se a execução dos métodos de aprendizado de máquina: primeiro, a partir da quantidade de dados obtida, calculou-se a melhor quantidade do número de clusters a ser processado pelo algoritmo *K-Means*, usando cálculo do "*Elbow Method*" (ou, método do cotovelo). Após isso foi executado o *K-Means* e se separou a base de dados em grupos (*clusters*) e cada grupo foi analisado separadamente.

A primeira etapa da análise feita no grupo, foram os acidentes presentes naquele grupo que serão relativamente relevantes para a execução do algoritmo analisando a quantidade de acidentes presentes dentro do grupo, verificou se a quantidade era menor que a mínima aceitável, o algoritmo assumiu que os acidentes daquele grupo não precedem um padrão e foram obras do acaso, e assim, foram descartados.

A segunda etapa da análise consistiu em separar os dados em treino e teste, isso foi necessário para a execução de qualquer método de aprendizado de máquina, principalmente os métodos de regressão. O método de separação dos dados consistiu em selecionar aleatoriamente um terço dos dados do grupo para teste, e dois terços dos dados para treino.

Após a separação entre treino e teste, os dados de treino foram usados no treinamento do modelo de regressão do *KNN* usando como parâmetro K=1, após o treino ser feito com sucesso foi executado o teste do modelo com os dados de teste e o resultado do teste foram armazenados como valores de predição. Os valores de entrada e saída do modelo preditivo pode ser observado na tabela 2.

Tabela 2. Atributos de entrada e saída de teste do modelo preditivo

Entrada	Saída
Dia	Latitude e Longitude
Mês	
Ano	
Hora	
Minuto	
Vítimas	
Fatal	

Fonte: O autor.

2.5. Exploração

Após o modelo ter sido treinado e a predição ter sido executada com êxito, foram feitos os cálculos de validação da predição. As métricas usadas para a validação foram: o erro médio absoluto e o cálculo dos resíduos.

O erro médio absoluto, ou MAE (*median absolut error*) calculou a média de variação entre a resposta encontrada pela predição, e o valor que a predição deveria ter encontrado. Quanto mais próximo de 0 é o MAE, mais preciso é o modelo preditivo.

O cálculo dos resíduos gerou gráfico de dispersão plotado a partir da variação dos valores reais x preditivos. Para a avaliação do modelo regressivo, quanto mais o gráfico de dispersão

tender a zero, melhor será a qualidade da predição.

Após a avaliação da predição, calculou-se o índice de acertos, para isso foi necessário definir o índice de tolerância na variação dos valores preditos, ou seja, o máximo de variação que o valor predito e o real podem ter para se tornar tolerável.

Com isso, foi mostrado todos os resultados dos cálculos efetuados, além dos gráficos de dispersão dos resíduos, o mapa gráfico de incidência obtido pelos grupos e o mapa gráfico dos dados preditivos com uma opção de comparação visual com os valores reais.

3. RESULTADOS OBTIDOS

Na execução das funções descritas na implementação, foi usado como filtro a cidade de São Paulo – SP, selecionado a sexta-feira como dia da semana no período noturno. Foi lido 5000 dados de acidentes que ocorreram nesse período entre 2019 e 2021.

No cálculo do melhor valor do K no algoritmo do *K-Means* usando o método do cotovelo, obteve-se 500 grupos. Com isso, os acidentes foram avaliados em grau de relevância, e o valor da relevância foi definida como 3, ou seja, grupos com 3 ou menos acidentes foram considerados acidentes do acaso e assim, desconsiderados, resultando em 2476 acidentes selecionados como relevantes para o treinamento do modelo.

Assim parte-se para a regressão, onde o valor do K do algoritmo do *KNN* foi definido como 1, um terço dos acidentes relevantes foram considerados para teste, e 2 terços para treino, obtendo-se assim 1061 acidentes previstos.

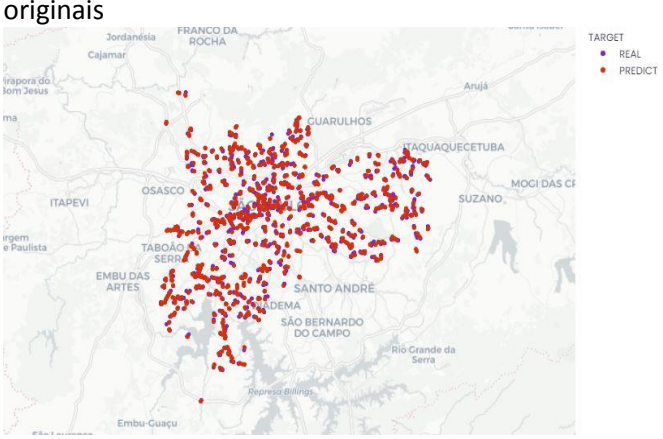
O resultado do cálculo das métricas descritas anteriormente podem ser observados na tabela 3, onde se obteve uma precisão de aproximadamente 96% de acertos, considerando um índice de tolerância de erro, de 0.5Km (ou 500m). Obteve-se também um erro médio convertido em unidade de medida de distância pela métrica de Haversine de aproximadamente 100m, a relação entre os dados pode ser observada visualmente na figura 9, onde se tem um comparativo entre os acidentes previstos e os acidentes originais.

Tabela 3. Atributos métricos do modelo

Entrada	Saída
Precisão	96.04%
MAE	0.0013
Erro Médio (m)	101m
Tolerância (m)	500m

Fonte: O autor.

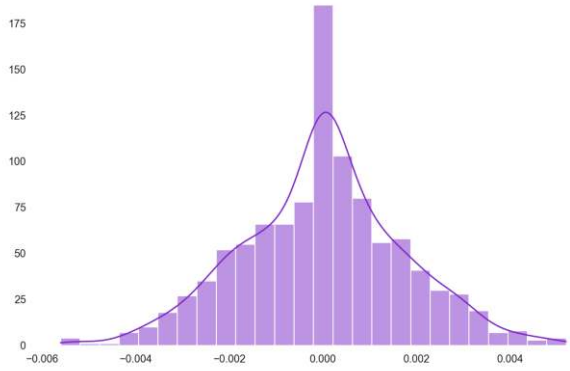
Figura 9. Comparativo entre acidentes previstos e originais



Fonte: O autor.

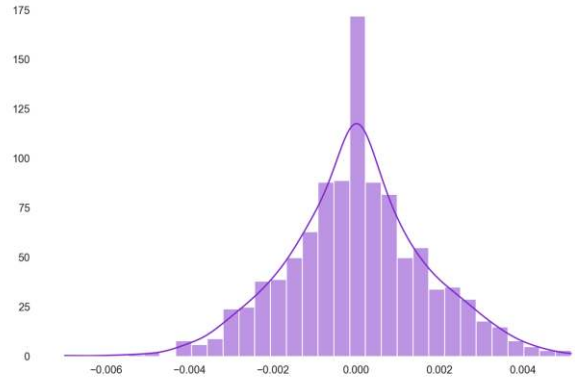
O cálculo dos resíduos da predição foi feito para cada saída da predição, como a predição tem como retorno 2 valores (latitude e longitude), calculou-se os resíduos para ambos os atributos. Os resíduos da latitude podem ser visualizados na figura 10 e os resíduos da longitude podem ser visualizados na figura 11. Analisando ambos os gráficos são possíveis notar que a dispersão dos resíduos de ambos tende a zero.

Figura 10. Resíduos da latitude



Fonte: O autor.

Figura 11. Resíduos da longitude



Fonte: O autor.

4. CONSIDEREÇÕES FINAIS

Neste trabalho foi proposto uma análise de dados de acidentes que já ocorreram, e utilizar regressão com o algoritmo *KNN*, a fim de poder prever locais de acidentes com ênfase na data e hora do acidente em questão, para minimizar os erros, a discrepância na variação da distância e selecionar os dados mais relevantes para treinamento do modelo, foi proposto uma abordagem de agrupamento dos dados utilizando o algoritmo *K-Means* (com o método do cotovelo para encontrar o melhor valor do K) a fim de poder executar as funções regressivas precisamente em cada grupo e considerar a predição como um todo no final da regressão do modelo.

Pode-se concluir que, a utilização dos métodos propostos traz um resultado conclusivo, pois, apesar do *KNN* ser considerado um método de aprendizado de máquina muito lento, pelo fato dele se basear em cálculos de médias, a regressão dos grupos mostrou um alto índice de acertos, sendo aproximadamente 96% de acertos considerando o valor de tolerância.

Conclui-se então, que a regressão de dados com base em geolocalização (latitude e longitude) pode se obter um resultado mais preciso e com menos erros se utilizar um método de amostragem, a fim de separar a base de dados em amostras, para no fim da análise, ser considerado como um todo.

Durante o desenvolvimento desse trabalho muitas opções relevantes tiveram que ser descartadas pelo motivo de foco da pesquisa em si, mas seria de interesse em projeto futuros, considerar essas opções.

Uma dessas opções foi a análise de acidentes baseado não só em atributos de tempo (data e hora) como também, causa do acidente, a qual pode-se ter uma relação entre os locais e as causas do mesmo.

Futuramente seria viável, utilizar métodos diferentes de regressão e fazer um comparativo sobre os resultados da regressão por grupos.

Uma outra ideia para uma pesquisa futura, seria um sistema de apoio a decisão, que utilizaria dos dados de acidentes previstos, para conseguir calcular melhores rotas de navegação dentro de regiões urbanas.

5. REFERÊNCIAS

AN BRUMMELEN, Glen. **Heavenly Mathematics:** the forgotten art of spherical trigonometry. New Jersey: Princeton University Press, 2013. 9 v. Disponível em: https://books.google.com.br/books?id=0BCCz8Sx5wkC. Acesso em: 12 dez. 2021.

BONIN, Morvana. **Regressão Linear Simples:** um pouco sobre regressão linear. 2019. Disponível em: https://tableless.com.br/regressao-linear-simples/. Acesso em: 30 abr. 2021.

BRASIL. DATASUS. **Sistema de Informações de Mortalidade - SIM**. Disponível em: http://tabnet.datasus.gov.br/cgi/deftohtm.exe?sim/cnv/ext10uf.def. Acesso em: 09 fev. 2021.

BSTM - BRAZILIAN SOCIETY OF TROPYCAL MEDICINE (Brasília). **Traffic accidents:** Over 1.35 million people lose their lives, says WHO: according to report from the who, traffic injuries are the leading cause of death among children and young people between the ages of 5 and 29. 2019. Disponível em: https://www.sbmt.org.br/portal/traffic-accidents-over-1-35-million-people-lose-their-lives-says-who/?locale=en-US&lang=en#:~:text=By%2020 20%2C%20Brazil%20must%20comply,China%2C%20the%20US%20and%20Russia. Acesso em: 10 fev. 2021.

DABBURA, Imad. **K-means Clustering:** Algorithm, Applications, Evaluation Methods, and Drawbacks. 2018. Disponível em: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a. Acesso em: 10 fev. 2021.

ECORODOVIAS. **Pelo menos, 90% dos acidentes de trânsito poderiam ser evitados**. 2020. Disponível em: https://g1.globo.com/sp/campinas-regiao/especial-publicitario/ecorodovias/noticia/2020/09/23/pelo-menos-90percent-dos-acidentes-de-transito-poderiam-ser-evitados.ghtml. Acesso em: 8 jun. 2021.

ISI-TICS. Senai. **Aprendizagem supervisionada ou não supervisionada.** 2018. Disponível em: https://isitics.com/2018/08/28/aprendizagem-supervisionada-ou-nao-supervisionada/. Acesso em: 12 fev. 2021.

JOSÉ, Italo. **KNN (K-Nearest Neighbors) #1**. 2019. Disponível em: https://medium.com/brasil-ai/knn-k-nearest-neighbors-1-e140c82e9c4e. Acesso em: 30 abr. 2021.

LUZ, Filipe. **Algoritmo KNN para classificação**. 2019. Disponível em: https://inferir.com.br/artigos/algoritimo-knn-para-classificacao/. Acesso em: 30 abr. 2021.

ROCHA, Julio. **Classificação e regressão com K-nearest neighbors**. 2018. Disponível em: https://profes.com.br/julio.c.p.rocha/blog/classificacao-e-regressao-com-k-nearest-neighbors. Acesso em: 30 abr. 2021.

SÃO PAULO. Governo do Estado de SP. Respeito à Vida: São Paulo dirigindo com reponsabilidade. São Paulo dirigindo com reponsabilidade. **Infosiga.SP**, 2021. Disponível em: http://www.respeitoavida.sp.gov.br/. Acesso em: 14 ago. 2021.

SARAGIOTTO, Daniela. Mortes no Trânsito: Tráfego brasileiro mata 1 pessoa a cada 15 minutos. Mobilidade.**Estadão**, São Paulo, 15 set. 2020. Disponível em: https://mobilidade.estadao.com.br/mobilidade-com-seguranca/mortes-no-transito-brasileiro-mata-1-pessoa-a-cada-15-minutos/. Acesso em: 09 fev. 2021.

SILVA, Paulo *et al.* Modelos de regressão aplicados a predição do desempenho escolar de estudantes do ensino fundamental. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 30., 2019. São Paulo. **Anais** [...]. São Paulo: Sociedade Brasileira de Computação,2019. Disponível em: http://dx.doi.org/10.5753/cbie.sbie. Acesso em: 12 fev. 2021. https://doi.org/10.5753/cbie.sbie.2019.1621

SILVA, Andrio Rodrigo Correa da. **Predição de localização de crimes em região urbana usando algoritmos de análise de regressão**. 2020. 99 f. Dissertação (Mestrado) - Universidade Federal do Ceará, Sobral, 2020. Disponível em: http://repositorio.ufc.br/bitstream/riufc/56162//2020_dis_arcdsilva.pdf. Acesso em: 12 fev. 2021.

SYAKUR, M. A. *et al.* **Integration K-Means clustering method and elbow method for identification of the best customer profile cluster**. 2021. Disponível em: https://iopscience.iop.org/article/10.1088/1757-899X/336/1/012017. Acesso em: 14 dez. 2021.

TEMPORAL, Jessica. **Como definir o número de clusters para o seu KMeans**. 2019. Disponível em: https://medium.com/pizzadedados/kmeans-e-metodo-do-cotovelo-94ded9fdf3a9. Acesso em: 11 dez. 2021.

VIAS SEGURAS (Brasil). ASSOCIAÇÃO BRASILEIRA DE PREVENÇÃO DOS ACIDENTES DE TRÂNSITO. **Estatísticas nacionais de acidentes de trânsito**. 2020. Disponível em: http://vias-seguras.com/layout/set/print/os_acidentes/estatisticas/estatisticas_nacionais. Acesso em: 23 nov. 2021.