UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

RICARDO AUGUSTO MÜLLER

APLICAÇÃO DE APRENDIZADO DE MÁQUINA PARA IDENTIFICAR O MEIO DE TRANSPORTE BASEADO EM LOCALIZAÇÕES DE GPS

CHAPECÓ
2019

RICARDO AUGUSTO MÜLLER

APLICAÇÃO DE APRENDIZADO DE MÁQUINA PARA IDENTIFICAR O MEIO DE TRANSPORTE BASEADO EM LOCALIZAÇÕES DE GPS

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.
Orientador: Dr. Denio Duarte

CHAPECÓ
2019

Müller, Ricardo Augusto

Aplicação de aprendizado de máquina para identificar o meio de transporte baseado em localizações de GPS / Ricardo Augusto Müller. – 2019.

46 f.: il.

Orientador: Dr. Denio Duarte.

Trabalho de conclusão de curso (graduação) – Universidade Federal da Fronteira Sul, curso de Ciência da Computação, CHAPECÓ, SC, 2019.

1. Aprendizado de máquina. 2. Predição. 3. Modelo. 4. Algoritmos. 5. Método de transporte. I. Duarte, Dr. Denio, orientador. II. Universidade Federal da Fronteira Sul. III. Título.

E-mail: ricardo.muller@estudante.uffs.edu.br

RICARDO AUGUSTO MÜLLER

APLICAÇÃO DE APRENDIZADO DE MÁQUINA PARA IDENTIFICAR O MEIO DE TRANSPORTE BASEADO EM LOCALIZAÇÕES DE GPS

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Dr. Denio Duarte

Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 03/07/2019.

BANCA AVALIADORA

Dr. Denio Duarte – UFFS

Prof. Dr. Guilherme Dal Bianco

Prof. Me. Adriano Sanick Padilha

RESUMO

A popularização de dispositivos controladores de geolocalização, como o GPS (*Global Position System*), criou uma nova demanda para aplicações que utilizem deste grande volume de dados, chamados dados móveis. Um exemplo é a recomendação de produtos, a partir de lugares frequentados, rastreamento ou até planejamento urbano. Estes dados podem possuir diversas informações sobre o usuário, por exemplo, o método de locomoção utilizado, a partir de sua localização, velocidade de seu trajeto e o próprio trajeto executado. A partir dessas informação é possível descobrir qual o meio de transporte utilizado pelo usuário. Este trabalho então, busca a criação de um modelo de predição classificatória de métodos de transporte, através de um conjunto de dados formado por dados móveis. Assim como a criação do modelo, este trabalho também avalia os resultados obtidos e compara com trabalhos relacionados, mesmo que tais trabalhos utilizem métricas e métodos diferentes. Os experimentos obtiveram resultados considerados bons, principalmente no modelo *Random Forest*, com uma acurácia de acerto acima de 80%. Como resultados satélites, este trabalho apresenta as classes que são mais facilmente preditas, além dos atributos mais informativos para a criação do modelo.

Palavras-chave: Aprendizado de máquina, GPS, dados móveis, conjunto de dados, métodos de transporte, árvore de decisão, *random forest*

ABSTRACT

The global use of geolocation dispositives, for example, GPS(Global Position System), has created a new demand for applications that manage the massive amount of data representing user's location, called mobile data. As an example, product and places recommendation and urban planning. The mobile data contains much information about user behavior: mean of transportation, user location, user speed, among others. This paper aims to build a classification model to predict user means of transportation. The work intends to estimate the results and compare them with the baseline, even it uses different metrics and methods. The experiments have good results, mainly in the Random Forest model, with an accuracy of over 80%. As additional results, this paper presents the more easily predicted' category, besides the more informative attributes for the creation of the model.

Keywords: Machine Learning, GPS, mobile data, dataset, transportation mode, decision tree, random forest

LISTA DE ABREVIATURAS

ALF	*Altitude Final*
ALI	*Altitude Inicial*
DAL	*Distância Altitude*
DLO	*Distância Longitude*
DLT	*Distância Latitude*
DP	*Discriminant Power*
DPQ	*Distância do primeiro quinto de pontos*
DPT	*Distância do primeiro terço de pontos*
DQQ	*Distância do quarto quinto de pontos*
DQU	*Distância do quinto quinto de pontos*
DSQ	*Distância do segundo quinto de pontos*
DST	*Distância do segundo terço de pontos*
DTM	*Distância Total em Metros*
DTP	*Diferença Tempo*
DTQ	*Distância do terceiro quinto de pontos*
DTT	*Distância do terceiro terço de pontos*
FN	*Falso Negativo*
FP	*Falso Positivo*
GPS	*Global Position System*
LOF	*Longitude Final*
LOI	*Longitude Inicial*
LTF	*Latitude Final*
LTI	*Latitude Inicial*
MAL	*Média de Altitude*

MLO	*Média de Longitude*
MLT	*Média de Latitude*
MTP	*Média de tempo*
N	*Número de segmentações da trajetória nos primeiros modelos de conjunto de dados*
TPQ	*Tempo do primeiro quinto de pontos*
TPT	*Tempo do primeiro terço de pontos*
TQQ	*Tempo do quarto quinto de pontos*
TQU	*Tempo do quinto quinto de pontos*
TSQ	*Tempo do segundo quinto de pontos*
TST	*Tempo do segundo terço de pontos*
TTQ	*Tempo do terceiro quinto de pontos*
TTT	*Tempo do terceiro terço de pontos*
VN	*Verdadeiro Negativo*
VP	*Verdadeiro Positivo*

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Com os avanços tecnológicos recentes e o aumento da capacidade de armazenamento de dados, novos ramos da tecnologia foram surgindo, entre eles, métodos de processamento e aprendizado deste grande volume de dados. A área de predição de dados é uma das áreas que vem ganhando força, utilizando técnicas avançadas de aprendizado de máquina, este volume de dados é processado, para a criação de modelos de predição. (ALPAYDIN, 2009)

Outra área que se popularizou com os avanços tecnológicos, foram as áreas de dados móveis, ou seja, dados que representem espaço e tempo. Principalmente pela grande capacidade de armazenamento de dados e redução de custo, os aparelhos GPS (*global positioning system*) tornaram-se itens comuns no cotidiano da maioria das pessoas. Encontrados principalmente integrados em smartphones e em veículos, estes aparelhos geram um grande volume de dados móveis. (MAZIMPAKA; TIMPF, 2016)

A mobilidade do usuário e o armazenamento dessa informação deu origem a uma variedade de aplicações WEB, nas quais o sistema GPS desempenha vários papéis na conexão entre estas aplicações e o usuário final. Além disso, a extração de conhecimento destes dados brutos gerados por um GPS pode fornecer informações de contexto para aplicativos geográficos e móveis (ZHENG; LIU et al., 2010).

A análise destes dados também pode ser usada em diversas áreas, como planejamento urbano, transportes, ecologia comportamental, análise de cenários esportivos, vigilância e segurança (MAZIMPAKA; TIMPF, 2016).

Com um grande volume de dados móveis, a utilização de métodos de aprendizado de máquina podem auxiliar na análise dos dados e criação de modelos para satisfazer as necessidades criadas por essas aplicações. Algumas aplicações destes modelos utilizam informações, como os métodos de transporte utilizados pelos usuários, para sugestões de rotas, informações sobre os trajetos ou até sugestões personalizadas de estabelecimentos. Alguns jogos atuais, no estilo mobile (desenvolvidos para smartphones e tablets) também utilizam destas informações, pois em uma de suas mecânicas, o meio de transporte utilizado é importante.

Um exemplo de jogo que possui mecânicas neste estilo é o *Pokemon Go*, lançado em 2016, da desenvolvedora *Niantic*, onde obtêm-se recompensas por caminhar certas distâncias com seu smartphone, porém, ao utilizar meios de transporte como carro, ônibus ou moto, a distância percorrida não é contabilizada pelo aplicativo.

Existem vários trabalhos relacionados a esta área, como os trabalhos (ZHENG; CHEN et al., 2008) e (JAHANGIRI; RAKHA, 2015), que são citados neste projeto. Mas ainda há vários projetos que buscam resultados melhores, construção de dados diferentes ou modelos mais rápidos, pois os trabalhos existentes acabam compreendendo uma parcela pequena, ou utilizam métricas válidas apenas para uma pequena área, por exemplo, projetos com uma taxa alta de acerto mas um tempo de execução muito alto ou grande custo operacional não podem ser utilizados por sistemas mobile (pelo baixo processamento quando comparado à máquinas

maiores, além de geralmente necessitar de resultados em um período muito curto de tempo).

Com a intenção de trabalhar com dados móveis, este trabalho tem como proposta a criação de um modelo de predição de meios de transporte através da utilização de técnicas de aprendizado de máquina supervisionado, mais precisamente, algoritmos classificadores. Os resultados serão comparados aos resultados do experimento proposto em (ZHENG; CHEN et al., 2008).

Utilizando dados de GPS e buscando prever o método de transporte de trajetórias, este projeto utilizará de técnicas de aprendizado de máquina supervisionado para criar dois modelos de predição, as árvores de decisão e *Random Forest*, além do uso de técnicas de criação e seleção de atributos também são aplicadas para melhorar os resultados. Com o auxílio da biblioteca *scikit-learn* da linguagem *python*, os modelos de aprendizado foram construídos, com resultados considerados bons, se comparados com o trabalho (ZHENG; CHEN et al., 2008). Acredita-se que os bons resultados são devidos a criação de novos atributos, que não foram utilizados no trabalho relacionado.

Para alcançar o objetivo, este projeto foi composto por algumas etapas: análise dos dados, construção de um conjunto de dados baseado em trajetórias, criação de dois modelos de classificação, que possuem o propósito de classificar a trajetória pelo método de transporte utilizado, utilização de ferramentas para melhorar os resultados obtidos e comparação dos resultados entre os modelos.

O trabalho está estruturado como segue. O Capítulo 2 apresenta breve definições sobre aprendizado de máquina e suas subáreas, além de algumas técnicas de aprendizado de máquina classificatória. O Capítulo 3 introduz o conceito de dados móveis e apresenta informações sobre *GPS*. O Capítulo 4 apresenta os trabalhos relacionados. Os capítulos 5 e 6 referem-se ao experimento, sendo o Capítulo 5 responsável por apresentar os conjunto de dados e métricas utilizados, além de relatar os experimentos realizados que justificam as decisões para as configurações do mesmo. O Capítulo 6 analisa os resultados obtidos nos experimentos, considerando as métricas utilizadas, também mostra a importância de cada atributo no conjunto, compara os modelos e resultados com o trabalho (ZHENG; CHEN et al., 2008). Por último, o Capítulo 7 possui as considerações finais deste trabalho.

2 APRENDIZADO DE MÁQUINA

O aprendizado de máquina é uma área que busca, através de um conjunto de dados pré-definido, gerar conhecimento às máquinas. Este processo também é conhecido como *knowledge discovery*, ou descoberta do conhecimento (MAZIMPAKA; TIMPF, 2016). Com o objetivo de auxiliar na tomada de decisões, de certa forma, inteligentes, o aprendizado de máquina busca, através de modelos matemáticos, reconhecer padrões nos conjuntos de dados e por meio destes modelos predizer resultados para novas entradas (DUARTE; STÅHL, 2019).

Basicamente, enquanto na programação tradicional, busca-se encontrar a saída correta para a entrada passada, nos métodos de aprendizado de máquina, o objetivo é criar um modelo correto para as entradas e saídas dos conjuntos de dados (DUARTE; STÅHL, 2019). Os conjuntos de dados (*Datasets*) utilizados no processo de aprendizagem de máquina consistem de atributos (*features*) de entrada (*input*) e saída (*output*), podendo esta última ser nula, para o caso de aprendizado não-supervisionado. O conjunto de dados geralmente é dividido entre o conjunto de treinamento (*training set*), ou seja, os dados utilizados para a criação do modelo (*model*) e aprendizagem do sistema e o conjunto de testes, aplicado no modelo construído para validação do mesmo, como exibido na Figura 1.

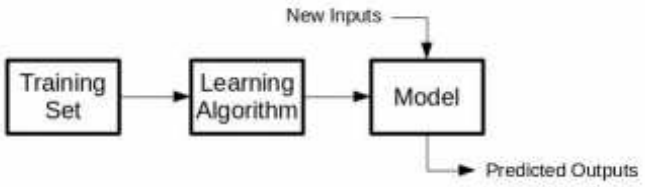


Figura 1 – Modelo processo de aprendizado de máquina

Fonte: (DUARTE; STÅHL, 2019)

2.1 APRENDIZADO SUPERVISIONADO

O método de aprendizado supervisionado aplica-se quando o conjunto de dados de treinamento possui atributos de entrada e saída correspondentes, ou seja, os dados encontram-se "rotulados". Os modelos de aprendizado de máquina supervisionados são classificados em "regressão" e "classificação".

2.1.1 Classificação

Problemas de classificação buscam, como o nome diz, criar um modelo que prevê a classificação dos dados sobre classes pré-definidas. A Figura 2 representa um exemplo de classificação, onde o modelo busca classificar ovelhas (círculos) e cabras (triângulos), com um modelo simples em duas classes, divididas por uma linha reta.

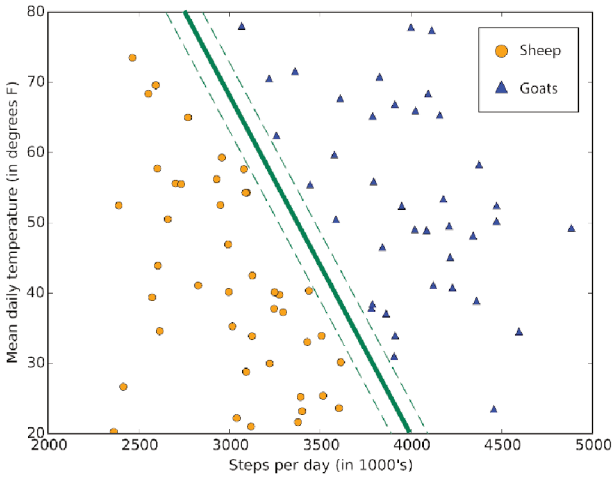


Figura 2 – Gráfico resultante do método de classificação

Fonte: (ZHANG, s.d.)

2.1.1.1 Árvore de decisão

As árvores de decisão são um método de funções de aproximação de valores discretos, onde o modelo de aprendizagem é representado por uma árvore (MICHEL, 1997).

A representação das árvores é feita começando por um nodo raiz e descendo até os nodos folha, onde encontram-se as classes definidas. Cada nodo intermediário da árvore corresponde a um teste do dado, onde cada ramo deste nodo corresponde a uma possível resposta ao teste. Basicamente, o dado inicia pelo nodo raiz e desce pelos testes da árvore, até encontrar sua classe no nodo folha. A Figura 3 mostra um exemplo de árvore de decisão que busca determinar a se deve-se ou não jogar tênis à partir do clima (*i. e.*, ensolarado, nublado, chuvoso, úmido ou com vento).

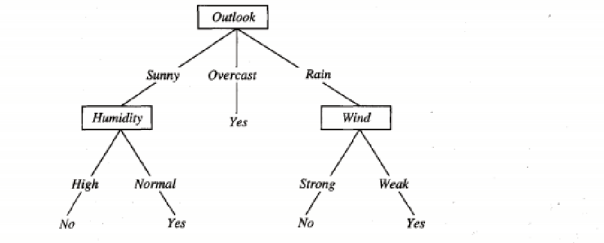


Figura 3 – Modelo árvores de decisão

Fonte: (MICHEL, 1997)

2.1.1.2 *Random Forest*

O método de *Random Forest* (ou florestas aleatórias em tradução livre) utiliza-se da combinação de árvores preditoras, onde cada árvore depende de um conjunto de valores aleatórios

independentes (BREIMAN, 2001).

Basicamente, o *Random Forest* busca melhorar os resultados da árvore de decisão à partir de uma combinação de diversas árvores, onde cada uma possui suas particularidades. Os dados de treinamento e de teste no método *Random Forest* serão então distribuídos para cada árvore de forma aleatória, buscando assim, aumentar a diversidades dos dados na floresta. Consequentemente, cada árvore tende a possuir testes e caminhos diferentes para os dados. No caso da classificação, este método define como resultado a classe com o maior número de resultados.

A Figura 4 é um exemplo de *random forest*, com três árvores de decisão. Neste exemplo, o dado passado recebeu 2 classificações diferentes, a árvore 1 classificou como classe A, enquanto as árvores 2 e 3 o classificaram como classe B. Sendo a classe B maioria, este seria o resultado final para o dado de entrada neste modelo de classificação.

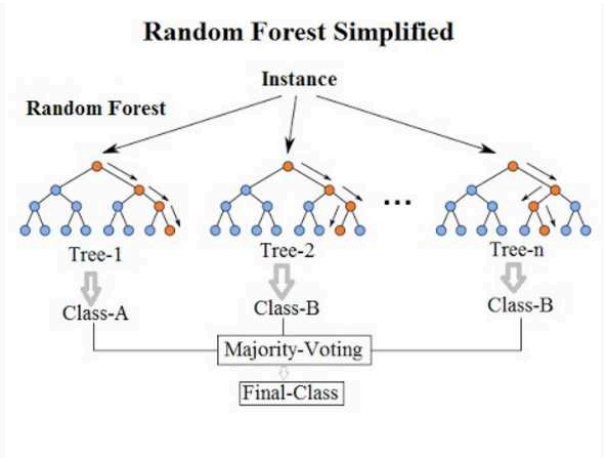


Figura 4 – Modelo *Random Forest*

Fonte: (KOEHRSEN, s.d.)

3 DADOS MÓVEIS

Dados gerados por aparelhos como GPS (*Global Positioning System*), telefones celular e sensores de rede registram a mobilidade dos indivíduos que o carregam. Por exemplo, um aparelho GPS (ou simplesmente um telefone celular com esta funcionalidade) registra dados em uma frequência definida, como por exemplo, um registro por segundo. Estes dados deixam uma "pegada", um rastro, com informações sobre um lugar no espaço e instante de tempo, por onde este dispositivo esteve. Estes dados sobre localização gerados são chamados dados móveis (BOGORNY; BRAZ, 2012).

A sequência gerada por um conjunto de dados chama-se trajetória. Com uma trajetória é possível rastrear o usuário durante o período registrado. A simples coleção de dados móveis é chamada de trajetória bruta, ou seja, a trajetória bruta é um conjunto de dados móveis, que devem minimamente possuir 4 campos, um id identificador, suas posições X e Y, além de uma variável que represente o instante de tempo da coleta do dado (BOGORNY; BRAZ, 2012).

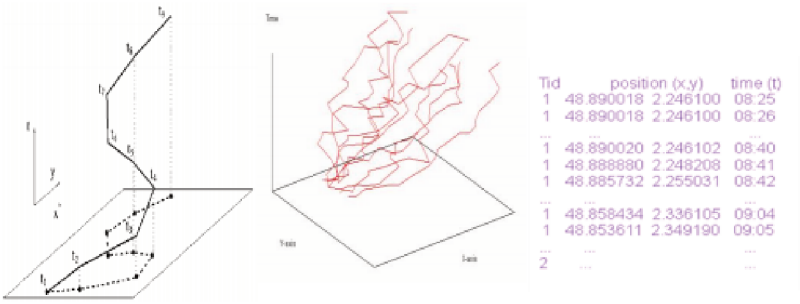


Figura 5 – Exemplo de uma trajetória, conjunto de trajetórias e conjunto de dados

Fonte: (BOGORNY; BRAZ, 2012)

Um pequeno exemplo é apresentado na Figura 5, onde a primeira imagem representa de uma trajetória bruta, gerada à partir de alguns dados móveis, enquanto a segunda imagem representa um conjunto de trajetórias semânticas, construídas à partir de trajetórias brutas e a terceira exemplo de conjunto de dados móveis, com os atributos Tid(identificador único), posições x,y e o tempo da coleta do dado.

3.1 *GPS - GLOBAL POSITIONING SYSTEM*

O GPS (*Global Positioning System* ou sistema de posicionamento global) é um sistema projetado para fornecer dados de localização instantâneos de um ponto na superfície da Terra ou próximo dela. Com projeto inicial para fins militares, durantes as últimas décadas teve seu uso estendido a aplicações civis. Hoje possui uma larga escala de uso, principalmente após a popularização dos *smartphones* que, em usa grande maioria, possuem o sistema GPS

inserido (JUNIOR, 2008).

O sistema de GPS funciona com 24 satélites, sendo 3 destes satélistes reserva, chamados *NAVISTAR* (*Navigation System with Time and Ranging*), que orbitam o planeta Terra a aproximadamente 20 mil km de altura. Os satélites basicamente funcionam com energia solar e baterias reserva para garantir seu funcionamento 24 horas por dia.

Para saber sua localização, o aparelho GPS faz uma requisição aos satélites. Ao existir uma requisição, o satélite envia ao aparelho um sinal, onde são passados dados de sua localização no espaço (posição e elevação). O aparelho então, usa estes dados e o intervalo de tempo de recepção para determinar sua distância para com o satélite. Com a informação de no mínimo 3 satélites, um aparelho GPS já é capaz de determinar sua atual posição (JUNIOR, 2008).

Os aparelhos de GPS fabricados atualmente podem ser considerados muito precisos, por utilizarem a tecnologia de multi-canais paralelos. Mas alguns fatores podem causar erros ou pequenas imprecisões na localização. Um exemplo de fator negativo é a reflexão do sinal em prédios altos ou montanhas, que pode gerar um pequeno atraso no envio ou recepção do sinal, alterando assim o resultado final do cálculo. Alguns erros mais técnicos como erros de órbita, que são dados incorretos na localização do satélite, ou uma geometria incorreta dos satélites também podem ocorrer (JUNIOR, 2008).

3.2 MÉTODOS DE TRANSPORTE

Métodos de transportes são os meios pelos quais uma pessoa, ou um grupo delas, utiliza para se locomover de um espaço físico à outro. Este agente de locomoção pode ser externo, como carros, aviões, motocicletas e etc, como partes integrantes do ser, como suas próprias pernas, que chamamos de locomoção a pé.

Os métodos de transporte podem variar de acordo com a necessidade e/ou preferência do usuário. Por exemplo, para atravessar de um continente à outro, a locomoção de motocicleta, carro ou a pé é inviável fisicamente, pois é impossível atravessar um oceano desta forma. Para ir de um continente à outro, podemos utilizar um avião, por exemplo, ou a união de mais de um método de transporte, onde, por exemplo, realiza-se a parte terrestre da locomoção com o auxílio de um ônibus e a travessia marítima através de um navio.

A velocidade de locomoção também varia entre os métodos, enquanto uma pessoa a pé, caminha em média 5 km/h, carros populares podem realizar médias acima de 100 km/h (variações ocorrem dependendo do modelo), enquanto um avião comercial move-se, durante o voo de cruzeiro, à algo próximo de 900 km/h.

Os métodos de transporte também podem ser divididos em transporte pessoais e coletivos onde, os métodos pessoais, caracterizam-se geralmente por bens pessoais, como carro próprio, motocicleta, bicicleta entre outros. Enquanto os métodos coletivos, geralmente caracterizam-se

por veículos maiores, como ônibus, navios ou aviões e na sua maioria, incluem um número maior de pessoas.

O grande volume de dados gerados pela popularização de aparelhos de GPS e o avanço nas tecnologias para armazenamento e processamento destes dados, criou uma nova área de pesquisa que tenta descobrir diversas informações sobre o perfil do usuário em termos de locomoção. Neste contexto, este projeto busca criar um modelo de classificação para identificar o método de transporte utilizado pelo usuário.

4 TRABALHOS RELACIONADOS

Este capítulo apresenta um breve resumo sobre os trabalhos relacionados que auxiliaram na construção deste projeto. As Seções 4.1 e 4.2 apresentam os dois principais trabalhos, o primeiro projeto (ZHENG; CHEN et al., 2008) trata-se de um projeto da *Microsoft* chamado *Geolife* que, a partir de dados coletados de *GPS*, busca predizer o método de transporte. O segundo projeto (JAHANGIRI; RAKHA, 2015) também busca predizer o método de transporte, a partir de um método de classificação supervisionado, com o adicional de algumas classes, quando comparado ao (ZHENG; CHEN et al., 2008).

4.1 PROJETO 1 - GEOLIFE E DADOS DE GPS

O trabalho de (ZHENG; CHEN et al., 2008) é parte de um projeto chamado *Geolife* da *Microsoft* (ZHENG; XIE; MA, 2010), que busca, através de um conjunto de dados, criar um modelo de predição de métodos de transporte, a partir do uso de aprendizado de máquina supervisionado. Este modelo tem como objetivo a classificação do método de transporte utilizado pelo usuário, a partir de dados de *GPS*.

O conjunto de dados foi criado a partir da contribuição de 45 pessoas em um período de seis meses, que carregam consigo um aparelho de *GPS* nos modelos *Magellan Explorist* 210 ou 300, além de alguns smartphones, os quais deveriam selecionar o instante inicial e final da rota, além do método de transporte. Os dados gerados por estes aparelhos foram então utilizados na construção do conjunto de dados do projeto.

Os resultados deste projeto seriam utilizados para algumas aplicações no *Geolife*, a principal delas seria a sugestão de uma trajetória alternativa de acordo com o método de transporte utilizado. Pois um usuário a pé não precisa respeitar as mesmas regras nas vias que outro utilizando um carro (ruas de mão única são um exemplo disto).

A Figura 6 mostra a diferença de sugestões de rota, considerando os diferentes métodos de transporte. Cada linha representa uma trajetória, enquanto as cores representam os diferentes usuários que registraram suas rotas. A Figura 6(a) representa as trajetórias da forma como os dados chegam do conjunto de dados, enquanto a Figura 6(b) apresenta sugestões de trajetórias de acordo com o método de transporte a ser utilizado, por exemplo, caso o usuário faça o trajeto de bicicleta, a recomendação de trajetória deveria ser a linha branca, ou, a mesma rota da Figura 6(c), enquanto, se o usuário preferir fazer a trajetória de carro, a recomendação do algoritmo deveria ser a trajetória da Figura 6(d).

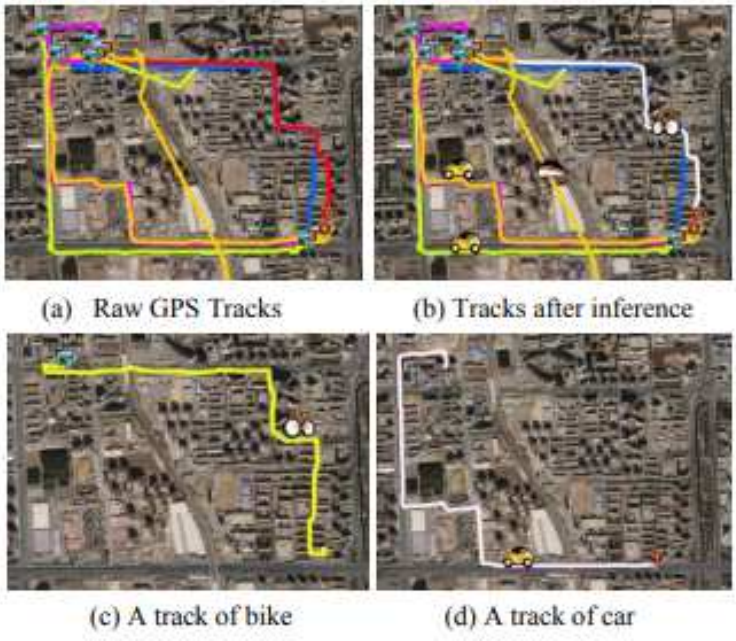


Figura 6 – Exemplo de recomendação de rotas do *Geolife* com diferença para métodos de transporte

Fonte: (ZHENG; CHEN et al., 2008)

Segundo (ZHENG; CHEN et al., 2008), ainda é possível existir diferentes sugestões, por exemplo, a de um restaurante, dependendo do meio de transporte. Isso devido à diferença de distância física que possa ser realizada, dependendo do transporte utilizado pelo usuário. Por exemplo: para um usuário que esteja caminhando, restaurantes próximos podem ser restaurantes que estejam à menos de 500 metros, enquanto um usuário que neste momento esteja dirigindo, um restaurante próximo pode significar dois ou três quilômetros.

Como todos os dados do conjunto foram criados por voluntários, (ZHENG; CHEN et al., 2008) aponta que, um problema encontrado, é que nem todos possuíam vontade de rotular suas trajetórias ou esqueciam de rotular partes das mesmas e não lembravam o exato momento em que alteraram o método de transporte, um exemplo é quando um usuário faz parte do percurso de carro, estaciona, caminha uma parte e decide pegar um ônibus.

Para resolver este problema foi preciso criar uma forma de separar as trajetórias, então o projeto inicialmente tenta prever o "ponto de mudança", que trata-se do ponto de *GPS* em que ocorre a troca do método de transporte. Essa separação cria segmentos de trajetória, onde cada segmento utiliza um método de transporte diferente. O modelo para prever os "pontos de mudança"foi criado utilizando as árvores de decisão. Alguns dados foram importantes para a definição dos "pontos de mudança", como por exemplo, velocidades iguais ou muito próximas à 0 durante a troca de método de transporte.

Neste primeiro conjunto de dados, coletado de 45 pessoas por seis meses, na maioria casos de troca de métodos de transporte, os usuários usavam o método "caminhar"entre o seu

método de transporte inicial e final, ou seja, o usuário saia, por exemplo, de um carro, caminhava pelo menos alguns metros e pegava um ônibus.

Após as segmentações das trajetórias, foram feitas as construções dos atributos a partir dos pontos de *GPS*, sendo eles a distância, velocidade média, expectativa de velocidade, alternância de velocidade, além das três maiores velocidades e três maiores acelerações para cada segmento da trajetória.

O conjunto de dados utilizado na criação do modelo foi criado a partir de dados de GPS (modelos *Magellan Explorist* 210 ou 300 e smartphones) com uma taxa de 1 dado por segundo, caso a velocidade do usuário fosse alterada, gerado por 6 meses, com o auxílio de 45 voluntários. Com um total de 20 mil quilômetros e 15 diferentes cidades. Para o treinamento, o conjunto de dados foi separado em 70% para treinamento, enquanto o resto foi utilizado para testar o modelo. Após a coleta, um novo conjunto de dados foi montado, com a junção de cada ponto, transformando em diversas trajetórias, sendo que definia-se o fim de uma trajetória e o início de outra quando a diferença entre dois pontos excedesse 20 minutos, caso o usuário não definisse o fim da mesma.

Também foram escolhidas quatro classes para o modelo, sendo elas:

- Bicicleta;
- Ônibus;
- Carro;
- Caminhar.

Foram escolhidos dois critérios para avaliar o modelo, a acurácia por distância e a acurácia por tempo. Foram também utilizados seis novos métodos de segmentação uniforme de trajetórias, três deles por tempo (60, 90 e 120 segundos), onde a trajetória era segmentada cada vez que a duração alcançava o tempo estipulado, e mais três por distância (100, 150 e 200 metros), segmentando a trajetória sempre que a distância estipulada era alcançada, possuindo assim 7 métodos de segmentação: "Ponto de Mudança", por duração de 60, 90 e 120 segundos e por distância de 100, 150 e 200 metros. Os melhores resultados foram obtidos pelo modelo de árvores de decisão, em todos os métodos de segmentação. Na segmentação por tempo, os melhores resultados encontram-se na segmentação em 120 segundos, com acurácia por distância de 68.7% e acurácia por tempo de 72.1% na classificação do método de transporte, além de uma revocação de 86.7% e precisão de 19.7% em descobrir o ponto de mudança.

Na segmentação por distância, os melhores resultados pertencem a segmentação em 100 metros, com acurácia por distância de 39.9% e acurácia por tempo de 67,4% na classificação do método de transporte, além de revocação de 86.7% e precisão de 14.8% em descobrir o ponto de mudança.

Na segmentação chamada *Change Point* (Ponto de Mudança), utilizando o modelo de árvores de decisão, a acurácia por distância foi de 67.5% e acurácia por tempo de 74.3%, além de revocação de 88.7% e precisão de 40.6% em descobrir o ponto de mudança.

Os melhores resultados encontrados no projeto (ZHENG; CHEN et al., 2008) foram encontrados utilizando as árvores de decisão, com a segmentação de "Ponto de Mudança", com as acurácias por distância e tempo de 67.5% e 74.3%, respectivamente, e, na tentativa de descobrir o ponto de mudança, a revocação foi de 88.7% e precisão de 40.6%. A tabela 1 apresenta os melhores resultados para cada modelo de segmentação, utilizando árvores de decisão.

Tabela 1 – Melhores Resultados por Segmentação

	120 seg	100 m	Ponto de Mudança
Acurácia Distância	68.7%	39.9%	67.5%
Acurácia Tempo	72.1%	67.4%	74.3%
F1-Score - Ponto Mudança	32.1%	25.28%	55.7%

4.2 PROJETO 2 - DADOS DE SMARTPHONE

Este projeto cria o conjunto de dados de uma forma diferente, utilizando sensores presente na maioria dos smartphones atuais, como o acelerômetro, giroscópio e vetor de rotação, sem utilizar os dados de *GPS*, pois, segundo (JAHANGIRI; RAKHA, 2015), os GPS possuem algumas limitações, como a falta de sinal em áreas subterrâneas ou cobertas, como túneis. Um dos problemas encontrados é que estes sensores não geram dados de localização diretamente, como o *GPS*, logo uma janela muito curta de dados precisa ser considerada para conseguir formar a trajetória, logo isto significa uma quantidade grande de dados em um período muito curto de tempo.

Inicialmente um aplicativo para smartphone foi desenvolvido, com o objetivo de armazenar os dados dos sensores acelerômetro, giroscópio e vetor de rotação, além de dados do *GPS* do aparelho, com a maior frequência possível. Posteriormente os dados de *GPS* foram ignorados, para simular momentos em que seu sinal estivesse indisponível.

Para a criação do conjunto de dados, foram distribuídos para dez empregados da *Virginia Tech Transportation Institute* (VTTI) aparelhos de smartphone com o aplicativo instalado (foram utilizados dois modelos de smartphone, o Galaxy Nexus e Nexus4). Os empregados deveriam, antes de iniciar um trajeto, escolher o método de transporte à ser utilizado, após isso deveriam iniciar e parar os trajetos no aplicativo, conforme o fizessem em seu dia a dia. Foi pedido aos empregados que também variassem a forma como carregam os smartphones, como em seu bolso, na palma da mão, mochila e etc.

As coletas foram realizadas durante o período de trabalho dos empregados, ou seja, de segunda à sexta, das 8:00 da manhã até 18:00 da tarde, em diferentes tipos de rodovias e linhas

de ônibus, para tentar tornar o conjunto o mais natural possível.

Um total de 5 classes foram selecionadas para este experimento, se compararmos com o projeto da Seção 4.1, houve o aumento de uma classe (correr), sendo assim, as classes são:

- Carro;
- Bicicleta;
- Caminhar;
- Correr;
- Ônibus.

Cada um dos voluntários armazenou 30 minutos de dados em cada um dos métodos de transporte, totalizando assim um total de 25 horas de dados armazenados. Destes dados 70% foram utilizados para o treinamento do modelo e 30% para teste.

As métricas utilizadas para este projeto foram a acurácia, *F1-Score*, o índice de Youden e o poder discriminante (DP, do inglês *Discriminant Power*).A acurácia é dada pelo número de acertos, dividido pelo número de dados testados. O *F1-Score* é uma combinação do *Recall* e da precisão. O índice de Youden é uma medida para avaliar a capacidade de evitar falhas de um modelo e o DP mede a eficácia em dividir classes do modelo.

Comparando todas estas métricas, os melhores resultados gerais encontrados pertenceram ao modelo *Random Forest*, com uma acurácia de 95.1%, *F1-Score* de 95.12%, poder discrimante com uma taxa de erro entre 4 e 6% e o índice de Youden de aproximadamente 93%.

4.3 RELEVÂNCIA DOS TRABALHOS RELACIONADOS

Este trabalho visa, assim como os trabalhos relacionados apresentados nas Seções 4.1 e 4.2, criar um modelo de classificação de método de transporte, utilizando os modelos com melhores resultados nos dois projetos, que são as árvores de decisão para o projeto (ZHENG; CHEN et al., 2008) e *random forest* para o projeto (JAHANGIRI; RAKHA, 2015).

A relevância do projeto (ZHENG; CHEN et al., 2008) neste projeto é ainda maior pois utilizará o mesmo conjunto de dados inicial utilizado no projeto (ZHENG; CHEN et al., 2008), devido ao fato de seus dados serem construídos a partir de dados de *GPS*. Uma observação importante é que no momento da criação destes novos modelos, o conjunto de dados esta maior e possuindo mais classes do que as descritas na Seção 4.1. As informações sobre o conjunto de dados deste projeto encontra-se na Seção 5.1 do Capítulo 5.

5 PROJETOS DE EXPERIMENTOS

Utilizando como base o conjunto de dados *Geolife GPS Trajectory* (ZHENG; LIU et al., 2011), este trabalho iniciou a construção do conjunto de dados, transformando as trajetórias brutas (conjunto de pontos) em um conjunto de trajetória semânticas, a partir dos dados móveis disponibilizados no conjunto de dados original. Foram criados dois conjuntos de dados para este projeto, descritos nas Seções 5.3 e 5.4. Ambos os conjuntos de dados transformavam cada arquivos de pontos em uma trajetória, mas com algumas variações em seus dados, para então posteriormente verificar qual obteve melhores resultados. A próxima seção descreve brevemente o conjunto de dados original.

5.1 CONJUNTO DE DADOS *GEOLIFE GPS TRAJECTORY*

O conjunto de dados *Geolife GPS Trajectory* (ZHENG; LIU et al., 2010; ZHENG; XIE; MA, 2010) foi construído com dados de 182 usuários, utilizando dados recolhidos de aparelhos de *GPS*, em um período de mais de cinco anos (entre abril de 2007 e agosto de 2012), totalizando 17.621 trajetórias. A maior parte dos dados foi coletada na China, principalmente na cidade de Beijing, mas ainda assim, existem dados de trinta cidades diferentes na China, além de algumas coletas nos Estados Unidos da América e Europa.

O arquivo é separado por pastas, cada pasta corresponde a um usuário que possui de 1 à 2.153 arquivos. Cada arquivo corresponde a uma trajetória, possuindo como conteúdo diversos pontos de *GPS* obedecendo o seguinte esquema:

- Latitude em graus decimais;
- Longitude em graus decimais;
- 0 (*este dado não possui utilidade no dataset*);
- Altitude em pés;
- Número de dias desde 30/12/1899 (com as partes decimais);
- Data;
- Tempo.

Abaixo, como exemplo, a primeira linha de dados, do primeiro arquivo, do usuário 000:

"39.984702,116.318417,0,492,39744.1201851852,2008-10-23,02:53:04"

Os arquivos de pontos também possuem seis linhas de cabeçalho, apenas para identificação, todas estas linhas foram ignoradas posteriormente na construção dos conjuntos de dados.

A construção das trajetórias neste conjunto de dados é dada pela junção de todos os pontos encontrados em cada arquivo, ou seja, a trajetória basicamente é um conjunto de pontos. O arquivo que armazena os pontos não possuem rótulos pois estes encontram-se em um arquivo externo chamado *labels.txt*, encontrado dentro da pasta de casa usuário. Este arquivo possui três campos, *Data e hora do início da rota*, *Data e hora do fim da rota*, *Método de transporte (rótulo)*.

No total, existem onze rótulos no conjunto de dados, todos descrevendo o método de transporte utilizado: *airplane (avião)*, *boat (barco)*, *bike (bicicleta)*, *walk (caminhar)*, *car (carro)*, *run (correr)*, *subway (metrô)*, *motorcycle (motocicleta)*, *bus (ônibus)*, *taxi (táxi)*, *train (trem)*.

A linha abaixo apresenta como exemplo o primeiro registro do arquivo *labels.txt* do usuário 010:

"2007/06/26 11:32:29 2007/06/26 11:40:29 bus"

Os dois primeiros dados correspondem respectivamente a data e hora de início e fim da trajetória, enquanto o último dado trata-se do rótulo.

Assim, para rotular os dados, é preciso analisar os pontos, criar as trajetórias e então, a partir da data de início e fim da trajetória encontrar o rótulo correspondente no arquivo *labels.txt*. Existem diversas formas de montar essas trajetórias, este trabalho então utilizou de dois métodos, ambos descritos nas Seções 5.3 e 5.4.

O conjunto de dados possui o total de 18.655 arquivos de pontos, porém nem todos os usuários registravam seus rótulos. Como este projeto trabalha apenas com dados rotulados, utilizou-se então apenas os dados de 69 usuários (todos que rotularam seus dados), com um total de 10.906 arquivos de pontos.

5.2 ROTULANDO OS DADOS

No momento da criação das trajetórias, ou seja, o agrupamento dos pontos existentes em cada arquivo, a data e hora de início e fim da trajetória são utilizadas para agrupar as trajetórias. O período da trajetória é então comparado com os dados presentes no arquivo *labels.txt* para encontrar o rótulo referente a esta trajetória . Como os arquivos de pontos e o arquivo *labels.txt* são separados por usuário, não houve preocupações com uma trajetória (arquivo com lista de pontos) cruzar o tempo de outra, pois como os dados foram criados por usuários reais de *GPS*, não é fisicamente possível mais de uma trajetória ser realizada pelo mesmo usuário no mesmo instante de tempo.

Ao final da criação das trajetórias, 2.100 trajetórias rotuladas foram criadas e estas trajetórias serão utilizadas na criação dos conjuntos deste trabalho.

Para rotular as trajetórias, atribuiu-se um valor numérico para cada classe existente, mas antes de enumerá-los, este projeto adotou a sugestão descrita no manual do conjunto de dados e os rótulos *carro* e *táxi* foram agrupados, assim como os rótulos *trem* e *metrô*. Sendo assim, o conjunto possui nove classes, enumeradas de zero a oito, conforme Tabela 2.

Tabela 2 – Rótulos numéricos

Rótulo Original	Rótulo Numérico
0	Caminhar
1	Bicicleta
2	Ônibus
3	Carro e Táxi
4	Trem e Metrô
5	Avião
6	Barco
7	Correr
8	Motocicleta

5.3 PRIMEIRO CONJUNTO DE DADOS - TRAJETÓRIAS SEGMENTADAS

Buscando construir um conjunto de dados com um número de atributos suficiente para obter resultados aceitáveis, criou-se inicialmente dois experimentos. O primeiro experimento divide a trajetória (formada a partir dos pontos) em **N** partes, a distância e o tempo, como se houvessem **N** rotas em uma só.

Os valores de **N** utilizados foram 3 e 5, sendo assim, quando **N** for igual a 3 e a trajetória possuir 30 pontos, a soma das distâncias ponto a ponto dos pontos 1 até 10 formarão a primeira trajetória, assim como a diferença de tempo dos pontos 11 até 20 formará a segunda trajetória, seguindo desta forma até utilizar todos os pontos. Então, para **N** igual a três tem-se um total de seis atributos, conforme apresentado na Tabela 3.

Tabela 3 – Atributos Conjunto Dados com N_3

Sigla	Atributo
DPT	Distância do primeiro terço de pontos
TPT	Tempo do primeiro terço de pontos
DST	Distância do segundo terço de pontos
TST	Tempo do segundo terço de pontos
DTT	Distância do terceiro terço de pontos
TTT	Tempo do terceiro terço de pontos

Os atributos formados para **N** igual a cinco assemelha-se aos atributos mostrados anteriormente, mas a divisão da quantidade de pontos será feita por cinco, assim como o número de atributos será aumentado para dez.

Tabela 4 – Atributos Conjunto Dados com N_5

Sigla	Atributo
DPQ	Distância do primeiro quinto de pontos
TPQ	Tempo do primeiro quinto de pontos
DSQ	Distância do segundo quinto de pontos
TSQ	Tempo do segundo quinto de pontos
DTQ	Distância do terceiro quinto de pontos
TTQ	Tempo do terceiro quinto de pontos
DQQ	Distância do quarto quinto de pontos
TQQ	Tempo do quarto quinto de pontos
DQU	Distância do quinto quinto de pontos
TQU	Tempo do quinto quinto de pontos

Conforme a Tabela 4, caso a trajetória possua 100 pontos, o atributo **DPQ** é soma da distâncias ponto a ponto dos pontos 1 a 20, enquanto o atributo **TPQ** trata-se da diferença de tempo entre os pontos 1 a 20. Totalizando, para **N** igual a cinco, 10 atributos no conjunto de dados.

A quantidade de atributos dos conjuntos de dados pode ser expressada por **N** * 2, pois, para cada parcela da divisão existem dois dados, a soma das distâncias e a diferença de tempo.

Quando a divisão do número de pontos não é exata, exemplo: 56 pontos, com **N** igual à 5, a quantidade de pontos por parcela é dada pelo chão do resultado da divisão, enquanto os pontos restantes serão adicionados à ultima parcela (neste caso à quinta parcela, nos pontos **DQU e TQU**).

Os atributos destes conjuntos de dados são separados pelo caractere ';' e ainda adiciona-se ao fim da linha o rótulo, conforme Tabela 7.

As Tabelas 5 e 6 apresentam valores possíveis para cada atributo, enquanto as linhas a seguir apresentam, respectivamente, exemplos de dados dos conjuntos com **N** igual a 3 e 5.

N_3: *"4.2275290538550125;652.0;8.770782251161693;594.0;12.837014294972809; 594.0; 4"*

N_5: *"66.95710521617062;7735.0;129.1595539709344;8522.0;209.13465002538535; 7750.0;260.79779918530477;7720.0;338.32566234667206;9279.0;4"*

Tabela 5 – Exemplo de dados para conjunto N_3

Sigla	Valor
DPT	4.2275290538550125
TPT	652.0
DST	8.770782251161693
TST	594.0
DTT	12.837014294972809
TTT	594.0
Rótulo	4

Tabela 6 – Exemplo de dados para conjunto N_5

Sigla	Valor
DPT	66.95710521617062
TPT	7735.0
DST	129.1595539709344
TST	8522.0
DTT	209.13465002538535
TTT	7750.0
DTT	260.79779918530477
TTT	7720.0
DTT	338.32566234667206
TTT	9279.0
Rótulo	4

Utilizando os métodos de *Árvores de Decisão* e *Random Forest* e divisão do conjunto em 70% para treino e 30% para treino, obteve-se resultados de 62% e 67% de acurácia para o conjunto de dados N_5, respectivamente, e 64% e 69% de acurácia para o conjunto com N_3, respectivamente.

Os resultados obtidos para os conjuntos de dados N_5 e N_3 não foram considerados bons, principalmente por, apesar de trabalhar de forma diferente com o conjunto de dados, em seu trabalho, (ZHENG; CHEN et al., 2008) possuir acurácias de 72% e 69%, para as predições por tempo de duração e distância da trajetória, respectivamente.

5.4 SEGUNDO CONJUNTO DE DADOS - DIFERENÇAS POR DADO

Com o intuito de melhorar os resultados obtidos pelo primeiro conjunto de dados apresentado na seção 5.3, um novo conjunto de dados foi proposto

Com um total de 15 atributos criados a partir dos atributos originais do conjunto de dados original, o novo conjunto de dados proposto não divide as trajetórias como o anterior, mas mantém alguns dados originais, como a posição inicial e final da trajetória. A distância de cada atributo também é calculada (Latitude, Longitude e Altitude). Para tentar compensar a diferença de quantidade de pontos que existiam nas trajetórias, este conjunto de dados também possui médias de latitude, longitude, altitude e tempo. Para completar os quinze atributos, este conjunto possui o tempo total da trajetória e a distância total em quilômetros. A Tabela 7 lista os atributos criado e uma sigla para facilitar futuras explicações.

Tabela 7 – Atributos Conjunto Dados 2

Sigla	Atributo
LTI	Latitude Inicial
LOI	Longitude Inicial
ALI	Altitude Inicial
LTF	Latitude Final
LOF	Longitude Final
ALF	Altitude Final
DLT	Distância de Latitude
DLO	Distância de Longitude
DAL	Distância de Altitude
DTP	Diferença de Tempo
MLT	Média de Latitude
MLO	Média de Longitude
MAL	Média de Altitude
MTP	Média de Tempo
DTM	Distância Total em Metros

Para a construção dos atributos, foram utilizados os dados LTI (*Latitude Inicial*), LOI (*Longitude Inicial*), ALI (*Altitude Inicial*), LTF (*Latitude Final*), LOF (*Longitude Final*) e ALF (*Altitude Final*) dos pontos originais. Estes postos foram escolhidos dos pontos de início e fim da trajetória. Já os dados *Distância de Latitude*, *Distância de Longitude* e *Distância de Altitude* foram calculadas usando soma das distância euclidiana ponto a ponto.

Considerando **nPonto** como o número de pontos existentes na trajetória, as Equações 5.1, 5.2 e 5.3 apresentam os cálculos de distância utilizados. A fórmula utilizada é a soma das distâncias euclidianas de ponto a ponto, para cada um dos atributos que definem localização.

$$Distância\ de\ Latitude = \sum_{i=1}^{nPonto-1} \sqrt{(latitude_i - latitude_{i+1})^2} \tag{5.1}$$

$$Distância\ de\ Longitude = \sum_{i=1}^{nPonto-1} \sqrt{(longitude_i - longitude_{i+1})^2} \tag{5.2}$$

$$Distância\ de\ Altitude = \sum_{i=1}^{nPonto-1} \sqrt{(altitudei - altitude_{i+1})^2} \tag{5.3}$$

O atributo DTP (*Diferença de Tempo*) armazena a diferença de tempo entre a maior data e hora e a menor data e hora encontrada no arquivo. Os atributos MLT (*Média de Latitude*), MLO (*Média de Longitude*), MAL (*Média de Altitude*), MTP (*Média de Tempo*) e DTM (*Distância Total em Metros*) são a divisão dos valores calculados nos pontos DLT, DLO, DAL e DTP respectivamente, pela quantidades de pontos existentes no arquivo.

O atributo DTM (*Distância Total em Metros*) é calculado à partir da soma das distâncias euclidianas de ponto a ponto conforme Equação 5.4. Considerando novamente **nPonto** o número de pontos existentes na trajetória.

$$DTM = \sum_{i=1}^{nPonto-1} \sqrt{(lat_i - lat_{i+1})^2 + (long_i - long_{i+1})^2 + (alt_i - alt_{i+1})^2} \tag{5.4}$$

O atributo *DTM* é então multiplicado por 111.32 para converter o dado de graus decimais para quilômetros ($1^o \approx 111.32km$, segundo (O' DANIEL; HUSSIEN; ABDULLA, 2016)).

Após a criação dos quinze atributos do conjunto de dados, verifica-se a data e hora inicial e final do conjunto, com o arquivo *labels.txt* para rotular a trajetória, conforme Tabela 7.

Assim como no conjunto de dados apresentado na seção 5.3, utilizou-se os métodos de árvores de decisão e *random Forest* para o novo conjunto de dados, além de novamente 70% do conjunto para treino e 30% para teste. Os resultados obtidos foram então mais animadores, com a acurácia aproximada de 72% e 80%, respectivamente. Ambos os conjuntos de dados possuem um total de 2.100 trajetórias, mas a distribuição de classes não é equivalente, como a Tabela 8 apresenta.

Tabela 8 – Quantidade de trajetórias por classe

Classe	Trajetórias
0	606
1	402
2	309
3	589
4	187
5	4
6	2
7	1
8	0
Total	2.100

A classe 8 (motocicleta) não possui nenhum registro de rota compatível no conjunto de dados, por isso a mesma foi desconsidera. As classes 5 (avião), 6 (barco) e 7 (correr) também foram desconsideradas em todos os experimentos, pois possuíam uma quantidade muito pequena de dados se comparadas às demais classes, somadas estas três classes representavam menos de 0.35% do conjunto de dados. Sendo assim o conjuntos de dados realmente possui quatro classes: 0 (caminhar), 1 (bicicleta), 2 (ônibus), 3 (carro e táxi) e 4 (trem e metrô).

5.5 AVALIAÇÃO DE RESULTADOS

Para avaliar os resultados obtidos pelos algoritmos a serem executados, duas métricas foram utilizadas, a acurácia, já utilizada na escolha do conjunto de dados, e o *f1-score*.

As definições de acurácia e *f1-score* são baseadas na matriz de confusão, esta matriz mostra as frequências de acerto de uma classe no modelo, utilizando-se de quatro valores: verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos(FN). Os valores de verdadeiro positivo e falso positivo, representam os acertos e erros na classe determinada positiva, enquanto VN e FN, representam os acertos e erros na classe determinada negativa.

A matriz de confusão é binária, ou seja, aplica-se aos valores falsos ou verdadeiros de uma classe, para aplicar à um modelo de predição com N classes, a matriz de confusão é criada classe a classe e o resultado final é a média das classes para cada um dos quatro valores.

A acurácia é o resultado das predições corretas, divido pela quantidade de predições, conforme a fórmula apresentada na Equação 5.5.

$$acuracia = \frac{VP + VN}{VP + VN + FP + FN} \tag{5.5}$$

A acurácia geralmente é utilizada quando o conjunto de dados possui um número de dados por classe balanceado. Devido à isto, este projeto utilizará também a métrica apresentada na equação 5.6, que é calculada à partir da precisão e revocação.

$$f1 - score = 2 * \frac{precision * recall}{precision + recall} \tag{5.6}$$

Torne-se necessário também, calcular a precisão e a revocação, sendo a precisão o número de acertos para as previsões verdadeiras do modelo, enquanto a revocação trata-se da relação de acertos entre as classes verdadeiras. As métricas são apresentadas nas Equações 5.7 e 5.8.

$$precision = \frac{VP}{VP + FP} \tag{5.7}$$

$$recall = \frac{VP}{VP + FN} \tag{5.8}$$

Dadas as métricas, este trabalho utilizou da acurácia e *f1-score* para avaliar os resultados obtidos nos experimentos.

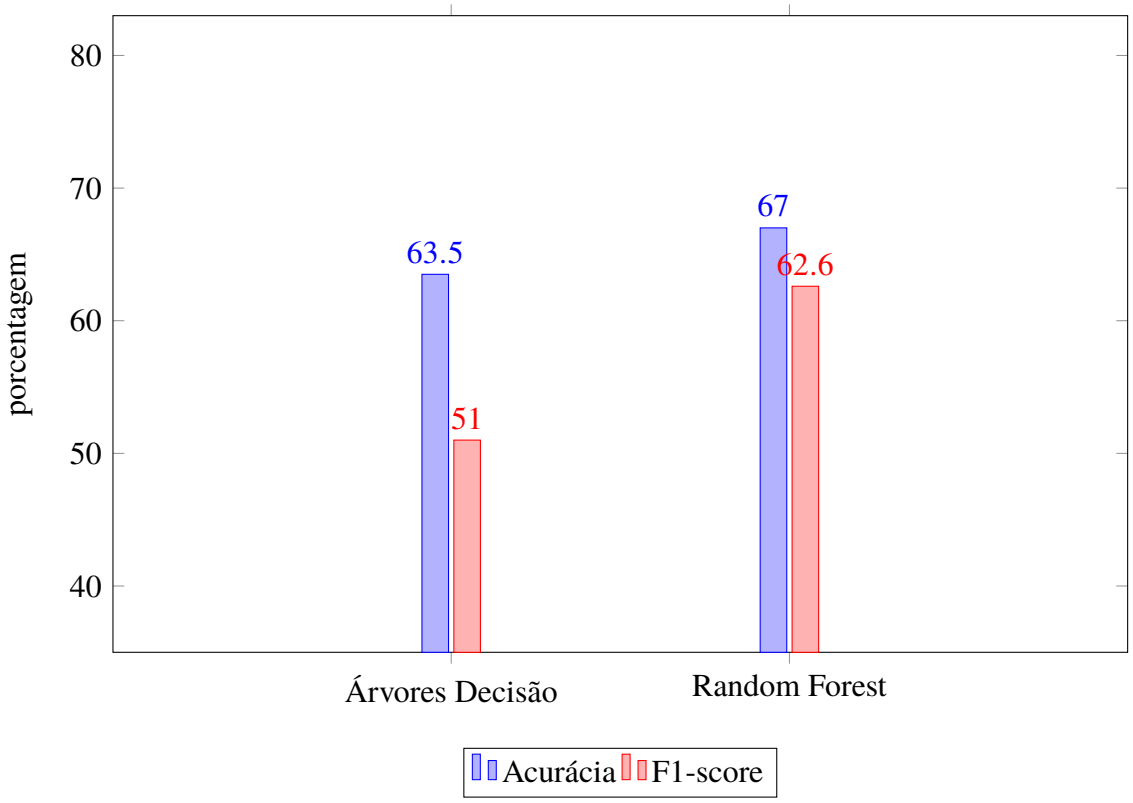
5.6 SELEÇÃO DO CONJUNTO DE DADOS

Com a criação de três conjuntos de dados, três experimentos foram realizados, utilizando, para cada experimento, os classificadores de árvores de decisão e *random forest*, para definir o conjunto que será utilizado no decorrer do projeto. A biblioteca *scikit-learn* da linguagem *Python* foi utilizada para a implementação dos algoritmos em todos os experimentos.

Este projeto adotou como padrão para todos os experimentos, o tamanho do conjunto de teste e treino, para os casos de *split*, sendo estes 70% e 30%, respectivamente.

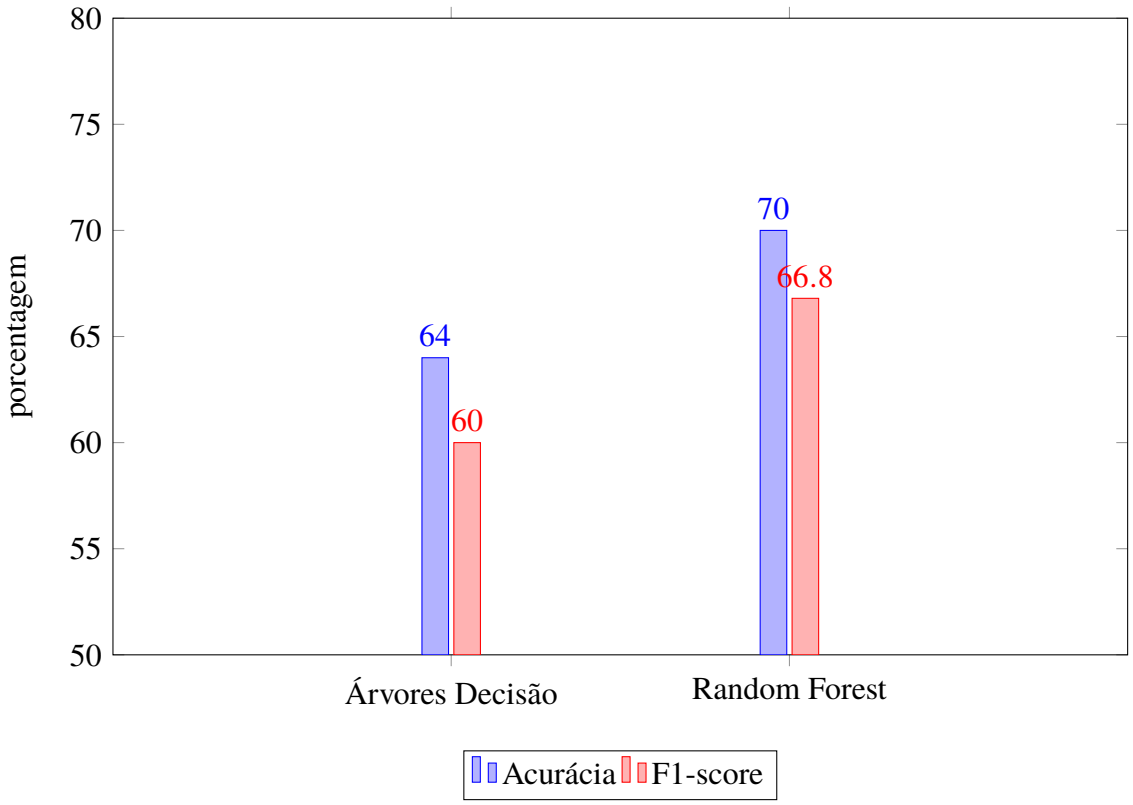
O primeiro experimento realizado utilizou o conjunto de dados descrito na seção 5.3, com **N**(número de segmentações da trajetória) igual à 5. Utilizando os parâmetros padrões para ambos os algoritmos classificadores, os resultados obtidos são apresentados na Figura 7. Neste experimento, o modelo *random forest* obteve os melhores resultados em ambas as métricas, com 67% de acurácia e 62.6% de *f1-score*, enquanto o modelo de árvores de decisão, resultou em uma acurácia de 63.5% e 51% de *f1-score*.

Figura 7 – Conjunto de dados N_5



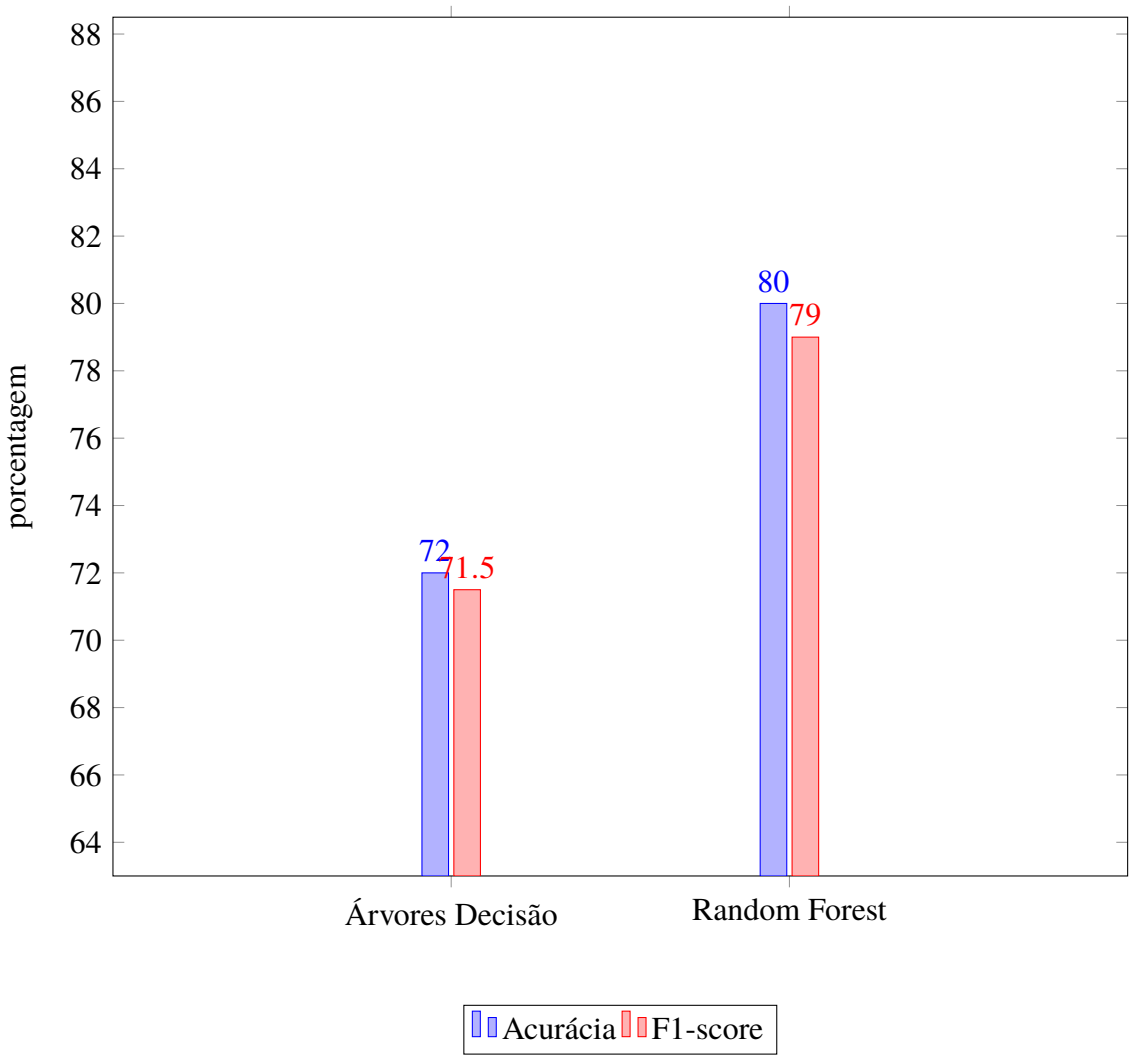
O segundo experimento utilizou o conjunto de dados similar, mas com **N** igual à 3. Assim como no experimento representado na Figura 7, este experimento obteve melhores resultados com o modelo *random forest* em ambas as métricas, sendo a acurácia de 70% e *f1-score* de 66.8%, contra 64% de acurácia e 60% de *f1-score* para o modelo de árvores de decisão, conforme apresenta a Figura 8.

Figura 8 – Conjunto de Dados N_3



Motivados pelos resultados obtidos nos experimentos anteriores, um terceiro experimento foi realizado, utilizando o conjunto de dados descrito na seção 5.4. Este experimento também utilizou os algoritmos classificadores de árvores de decisão e *random forest*, com os parâmetros padrões de ambos os métodos. A Figura 9 apresenta os resultados obtidos, ou seja, acurácia de 72% e *f1-score* de 71.5% para o modelo de árvores de decisão e acurácia de 80% e *f1-score* de 79% para o modelo *random forest*.

Figura 9 – Conjunto de Dados Final



O terceiro conjunto de dados obteve melhores resultado em acurácia e *f1-score* e devido a isto, todos os experimentos posteriores utilizam este conjunto de dados. Os novos experimentos utilizam métodos diferentes de separação de dados, combinação de hiperparâmetros e seleção de melhores atributos, como descrito na próxima seção.

5.7 CONFIGURAÇÃO DOS ALGORITMOS E CONJUNTO DE DADOS FINAL

Após a escolha do conjunto de dados para o experimento, este projeto utilizou algumas ferramentas do *scikit-learn* para buscar melhorar os resultados, entre elas está o método *GridSearchCV*.

O *GridSearchCV* busca, entre um conjunto de hiperparâmetros passado, as melhores combinações entre eles para sintonizar o modelo. Este método foi utilizado para ambos os algoritmos e os hiperparâmetros utilizados encontram-se no Apêndice A.

Disponível também na biblioteca *scikit-learn*, o método *SelectKBest* foi utilizado para selecionar os melhores atributos do conjunto. Como o conjunto possui quinze atributos, uma

iteração entre um e quinze foi realizada para descobrir qual o melhor **K** para o método (**K** é o número de atributos que o método *SelectKBest* irá selecionar).

Alguns métodos de dividir os dados foram testados para este experimentos, sendo eles o *train_test_split*, *KFold* e *StratifiedKFold*.

Os métodos *KFold* e *StratifiedKFold* utilizaram os parâmetros padrões e *shuffle* com o valor *True*, enquanto o método *train_test_split* realizou a divisão do conjunto em 70% para treino e 30% para teste.

Para este projeto foram utilizados os métodos *train_test_split*, *KFold* e *StratifiedKFold* para separação dos dados de treino e teste, as Figuras 10 e 11 apresentam as diferenças de resultado para cada método.

Figura 10 – Árvore de Decisão

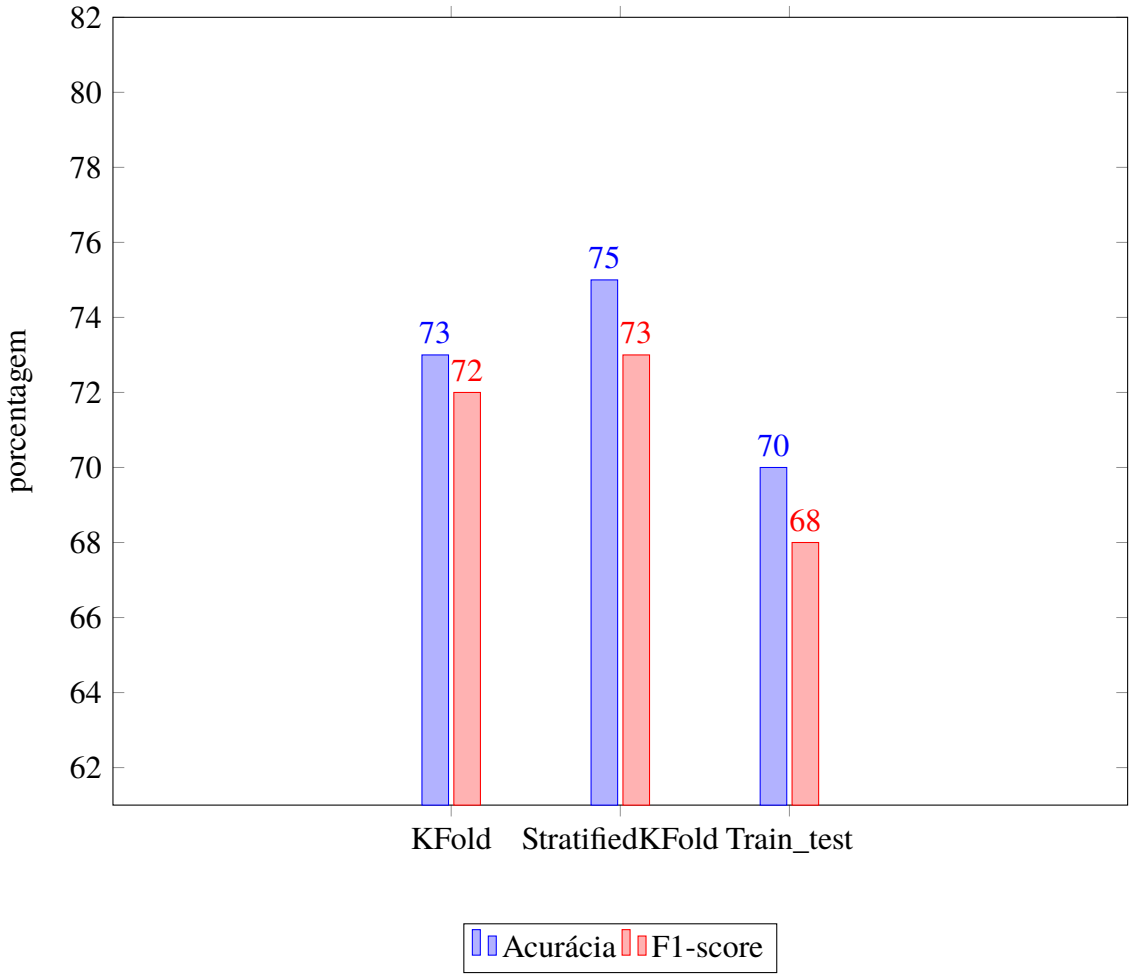
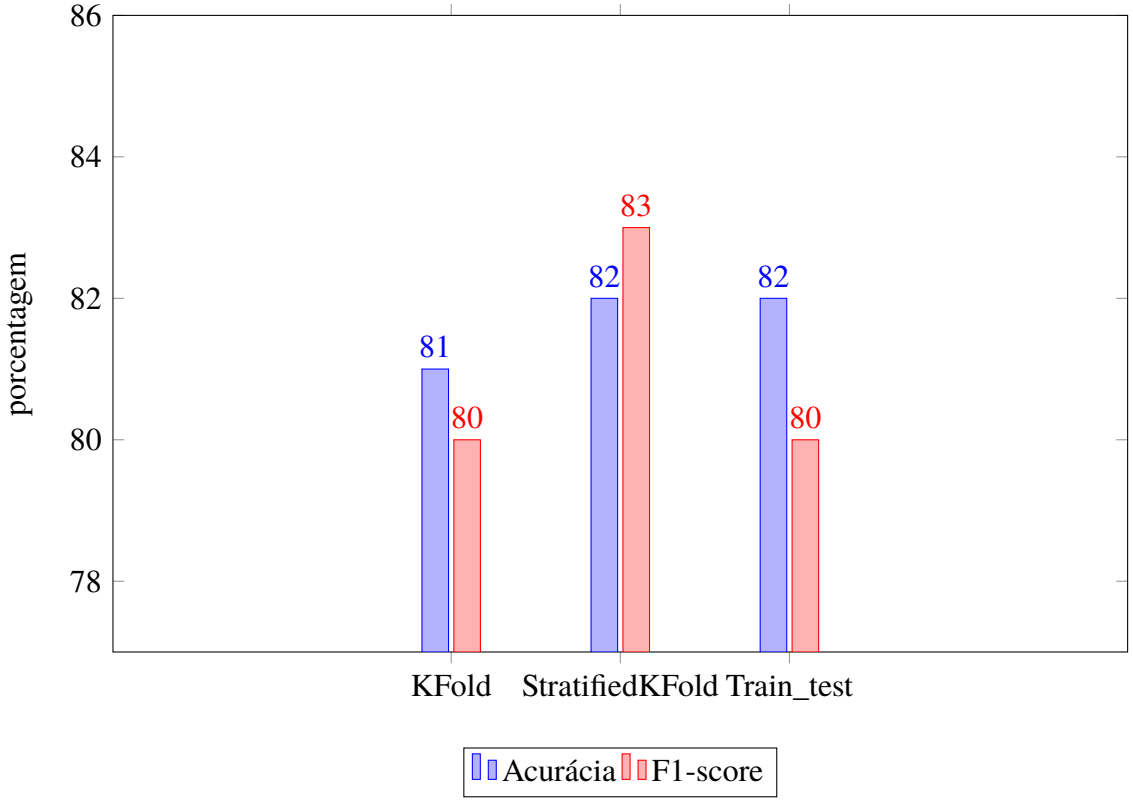


Figura 11 – Random Forest



Como as Figuras 10 e 11 apresentam, o método de separação com *StratifiedKFold* obteve acurácia e *f1-score* de 75% e 73%, respectivamente, para o modelo de árvores de decisão, enquanto para o modelo *random forest*, a acurácia obtida foi de 82% e *f1-score* de 83%.

Para o método *KFold*, os resultados para o modelo de árvores de decisão foram de 73% de acurácia e *f1-score* de 72%. No modelo *random forest*, os resultados alcançaram 81% de acurácia e 80% de *f1-score*.

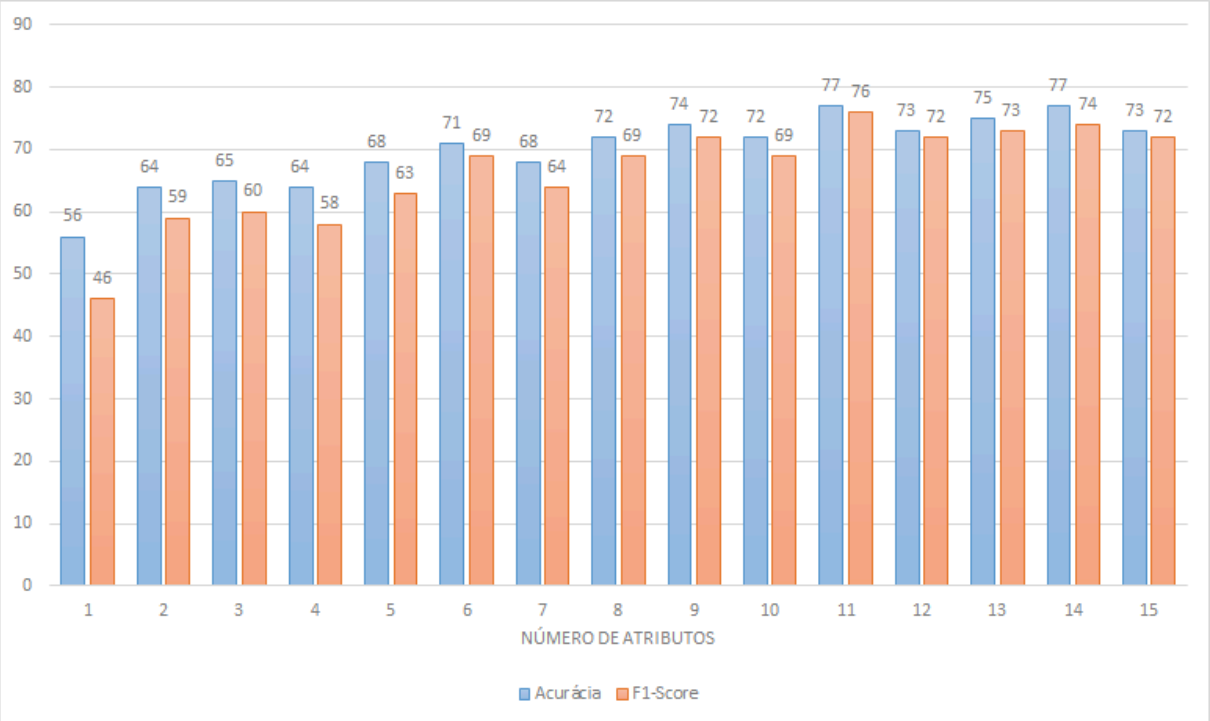
O último método, chamado *train_test_split* obteve 70% de acurácia e 68% de *f1-score* para as árvores de decisão e, no modelo *random forest*, 82% de acurácia e *f1-score* de 80%.

Os resultados com menor valor variaram de modelo, no modelo de árvores de decisão, o menor resultado ficou com o método de *train_test_split*, enquanto no *random forest*, o menor resultado foi obtido com o método *KFold* (na *f1-score* os resultados ficaram muito próximos, mas, considerando também a acurácia, o *KFold* possui o pior resultado).

Devido à esta superioridade nos resultados, através das métricas avaliadas, o método de separação *StratifiedKFold* foi utilizado para a continuação do experimento e todos os demais resultados, apresentados neste capítulo, utilizaram este método.

Utilizando o método *SelectKBest*, este projeto selecionou a quantidade de atributos com melhores resultados. Como este conjunto de dados possui 15 atributos os valores de k variaram entre 1 e 15 para K (sendo K o número de atributos a serem escolhidos). A Figura 12 compara os resultados para o modelo de árvores de decisão.

Figura 12 – KBest para o modelo de árvores de decisão

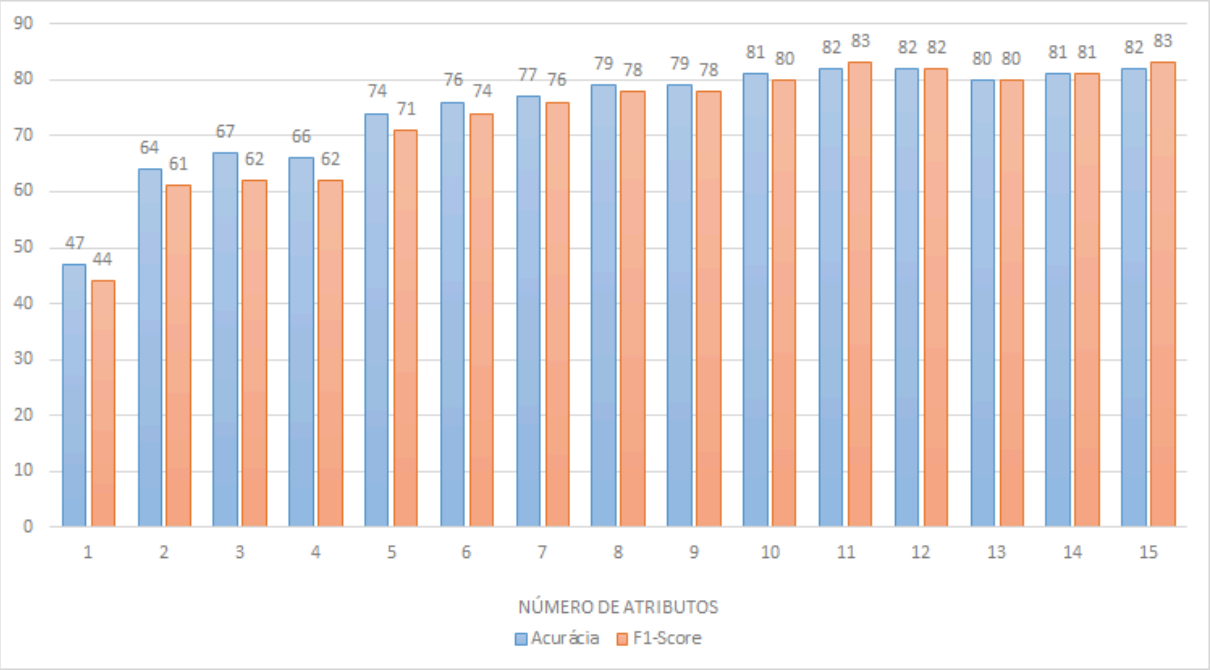


Conforme apresentado na Figura 12, os melhores resultados de k em acurácia foram 11 (77%) e 14 (77%), enquanto os piores resultados estão nos valores de k igual à 1 (56%), 4 (64%) e 2 (64%).

Para *f1-score*, os melhores resultados também encontram-se quando k é igual à 11 (76%) e 14 (74%), enquanto os piores resultados estão em k igual à 1 (46%) e 4 (58%). Para definir o melhor k para o modelo de árvore de decisão, considera-se ambas as métricas, sendo assim, com acurácia de 77% e *f1-score* de 76%, o k escolhido para este modelo é o 11.

Os melhores k atributos também foram selecionados para o modelo *random forest*, com os resultados apresentados na Figura 13. A melhor acurácia para o modelo *random forest* foi encontrado em k igual à 11(82%), 12(82%) e 15(82%), enquanto as piores acurácias encontram-se com k igual a 1 (47%) e 4 (66%). Para a métrica *f1-score*, os melhores resultados também encontram-se nos atributos 11 (83%), 12 (82%) e 15 (83%). Para a escolha do k para o modelo *random forest*, ambas as métricas foram consideradas e ainda assim, os valores de k 11 e k 15 foram iguais, tanto em acurácia (82%), quanto em *f1-score* (83%).

Figura 13 – KBest para o modelo *Random Forest*



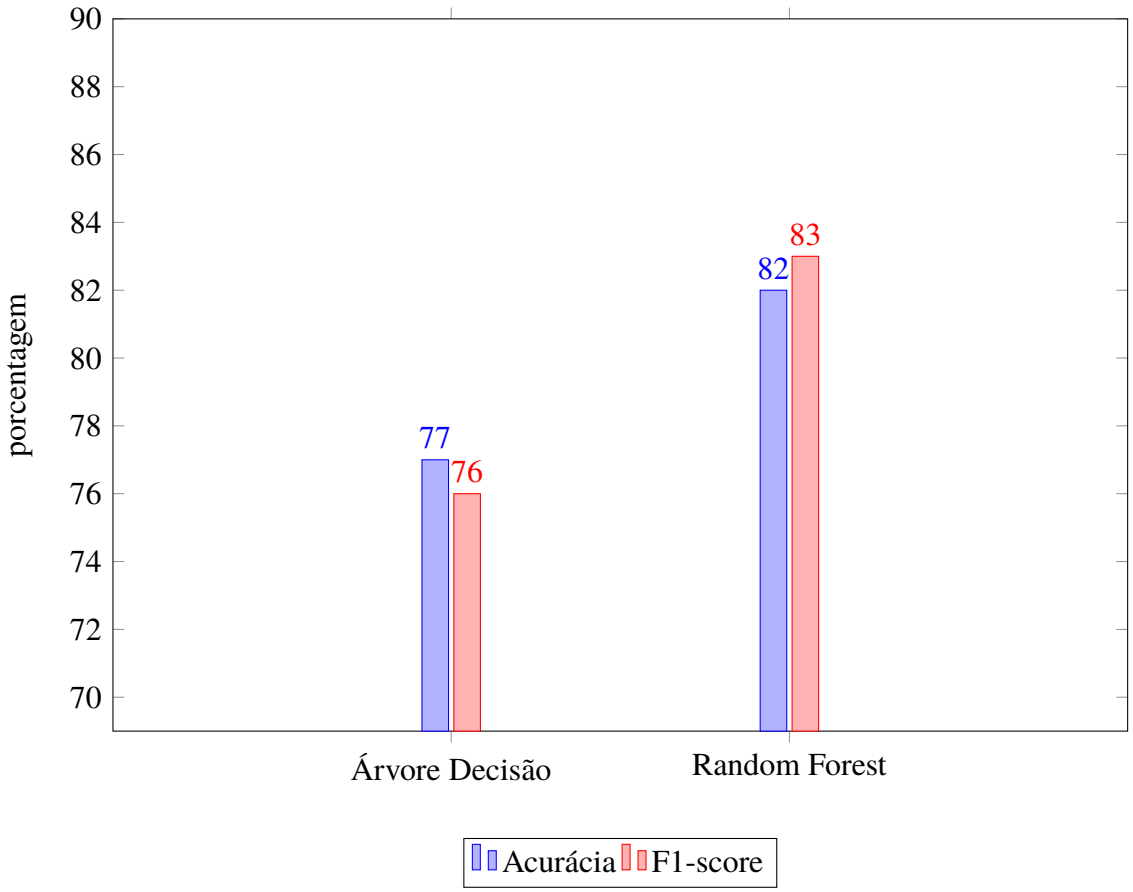
O próximo capítulo apresenta a análise dos resultados obtidos no experimento final, fazendo uma breve comparação entre os algoritmos, assim como, com os diversos métodos utilizados.

6 ANÁLISE DE RESULTADOS

Com os resultados obtidos, este projeto utilizou o método de *StratifiedKfold* para dividir os conjuntos de dados em teste e treino e utilizou os 11 atributos para o método de Árvores de Decisão devido aos resultados obtidos: acurácia de 77% e *f1-score* de 76%. Já para o modelo *random forest*, a acurácia e *f1-score* para o k igual 11 e 15 foram iguais(acurácia de 82% e *f1-score* de 83%). Como ambos obtiveram resultados similares, o número de atributos escolhido foi 11.

Os resultados obtidos na *random forest* foram superiores aos obtidos pela árvore de decisão, tanto em acurácia, com 82% para o modelo *random forest* e 77% na árvore de decisão, como na *f1-score*, sendo 83% para a *random forest* e 76% para a árvore de decisão, conforme Figura 14, sendo a diferença de *f1-score* de 7%, enquanto a acurácia diferencia-se em 5%.

Figura 14 – Comparativo entre algoritmos



Apesar de obter resultados acima de 70% de *f1-score* para a árvore de decisão e acima de 80% para a *random forest*, este valor é uma média geral e não corresponde aos valores de cada classe. As Tabelas 9 e 10 apresentam a precisão, revocação e *f1-score* para cada uma das classe dos modelos.

Para o modelo de árvores de decisão, a Classe 3 (Carro e Táxi) possui o melhor *f1-score*, com 77.11%, além do melhor percentual de precisão, com 83%, enquanto a melhor revocação

para este modelo pertence à Classe 1(bicicleta).

No modelo de *random forest*, o melhor *f1-score* pertence para a Classe 4 (Trem e Metrô), com 89.99%, esta classe também possui a maior revocação para o modelo, com 91%, enquanto a melhor precisão pertence a Classe 3(Carro e Táxi), com 90%.

A menor taxa, nas três métricas, pertence a Classe 2 (Ônibus), com *f1-score* de 53.65%, precisão de 49% e revocação de 58%, no modelo de árvores de decisão. No modelo *random forest*, a menor taxa nas três métricas também pertence a classe 2 (Ônibus), mas com resultados superiores, com *f1-score* de 74.5%, precisão de 75% e revocação de 74%.

Tabela 9 – Precisão, Revocação e F1-score por classe para árvore de decisão

Classe	Precisão(%)	Recall(%)	f1-score(%)
0	83	72	77.11
1	76	80	77.95
2	49	58	53.65
3	80	79	79.50
4	69	71	69.99

Tabela 10 – Precisão, Revocação e F1-score por classe para Random Forest

Classe	Precisão(%)	Recall(%)	f1-score(%)
0	81	83	81.99
1	78	81	80.43
2	75	74	74.50
3	90	84	86.90
4	89	91	89.99

Por utilizar o mesmo conjunto de dados original, torna-se indispensável uma comparação dos resultados obtidos neste trabalho, com os resultados de (ZHENG; CHEN et al., 2008), mesmo que, o conjunto final utilizado nos experimentos não sejam os mesmos.

O modelo de árvores de decisão foi utilizado em ambos os projetos, apesar de o trabalho de (ZHENG; CHEN et al., 2008) possuir dois valores para acurácia (por duração e distância), os resultados obtidos para este método (que inclusive obteve os melhores resultados) foram de 72.1% de acurácia para o método por distância e 68.7% para o método por duração, enquanto este projeto, que considera tempo e distância na criação dos atributos, obteve acurácia de 77% no modelo de árvores de decisão, mas resultados melhores também foram alcançados utilizando um método diferente, a *random forest*, com 82% de acurácia.

Considerando a outra métrica, *f1-score*, o projeto de (ZHENG; CHEN et al., 2008) calcula utilizando os "pontos de mudança", com precisão de 40.6% e revocação de 88,7%, possuindo um *f1-score* de 55.70%, utilizando o modelo de árvores de decisão. Este projeto por sua vez, no modelo de árvores de decisão, obteve *f1-score* de 76%, enquanto o modelo *random forest* obteve 83% de *f1-score*.

Apesar de possuir diferenças na execução dos projetos e a comparação ser difícil de ser realizada de forma direta (principalmente se considerar a forma como a segmentação das trajetórias é feita em ambos os projetos), os resultados desse projeto podem ser considerados satisfatórias ao realizar a comparação.

Após todos os experimentos e comparações realizadas neste projeto, o modelo utilizando o algoritmo *random forest* e o conjunto de dados com 15 atributos, utilizando o método de separação de dados *StratifiedKFold*, a seleção dos onze melhores atributos, além de utilizar os hiperarâmetros: *Bootstrap* como verdadeiro, *Criterion* como *Entropy*, a profundidade máxima da árvore como 10, o número máximo de atributos como automático (pois a seleção é feita anteriormente com o método *SelectKBest*), sem número fixo máximo de nós folhas, valor mínimo de diminuição de impurezas como 0, número mínimo de folhas e separações como 2, valor mínimo de peso das folhas como 0, além de 20 estimadores, obteve os melhores resultados.

Este modelo obteve uma acurácia geral de 82% e *f1-score* de 83%. Acredita-se que estes resultados foram obtidos principalmente devido à construção do novo conjunto de dados, que considera separadamente a diferença de cada um dos dados (latitude, longitude e altitude), além de dados de tempo e médias de latitude, longitude e altitude por ponto.

7 CONCLUSÃO

O objetivo principal deste projeto é a predição de métodos de transporte à partir de dados reais de *GPS*. Com base em um conjunto de dados já utilizado no projeto (ZHENG; CHEN et al., 2008), este projeto diferenciou-se principalmente no conjunto de dados, que foi reconstruído, apresentando as trajetórias de forma diferente e na utilização de um modelo diferente, o *random forest*.

Todos os atributos do projeto foram criados, a partir do conjunto de dados dos projetos (ZHENG; CHEN et al., 2008; ZHENG; LIU et al., 2010) que possuem dados reais retirados de aparelhos de *GPS*. Alguns pontos importantes puderam ser observados com este projeto, como as diferenças de longitude e latitude foram mais relevantes para a predição do que propriamente a distância total, assim como a média de tempo decorrido entre cada ponto de coleta foi mais relevante que o tempo total da trajetória.

Em relação aos resultados, eles foram considerados satisfatórios, considerando que o modelo de árvores de decisão obteve acurácia de 77% e *F1-Score* de 76%, enquanto o modelo de *Random Forest*, obteve 82% e 83%, principalmente considerando que existem 4 classes no conjunto de dados, onde a probabilidade geral é de 20%. As métricas utilizadas na avaliação também demonstraram que ambos os métodos possuíram dificuldade maior em preditar a classe 2 (Ônibus), enquanto os melhores resultados foram obtidos para a classe 3 (Carro e Táxi) para o modelo de árvores de decisão e a classe 4 (Trem e Metro) para o modelo *Random Forest*.

Comparando os resultados, o modelo *Random Forest* obteve melhores resultados em ambas as métricas, chegando em uma diferença de 7% para a *F1-Score*, que é a métrica que melhor descreve o modelo, considerando que a quantidade de dados por classe não é balanceada.

Como possíveis trabalhos futuros a este projeto pode-se pensar em uma possibilidade de aplicação em algum sistema de geolocalização, como *GPS*, para alterar a sugestão de rotas com os dados gerados (como foi utilizado o projeto (ZHENG; CHEN et al., 2008)), além da criação de um modelo em tempo real para utilizar nestes dispositivos. Outra possibilidade de uso pode ser na criação de padrões de usuários, que podem ser utilizados, por exemplo, em recomendações de produtos ou serviços personalizados por usuários. Ainda é possível usar este projeto como base para criação de modelos com outros algoritmos ou ferramentas, que busquem melhorar os resultados obtidos.

REFERÊNCIAS

ALPAYDIN, Ethem. **Introduction to Machine Learning**. Second. [S.l.]: MIT Press, 2009.

BOGORNY, Vania; BRAZ, Fernando José. **Introdução a trajetórias de objetos móveis: conceitos, armazenamento e análise de dados**. [S.l.]: Univille, 2012.

BREIMAN, Leo. Random Forests, 2001.

DUARTE, Denio; STÅHL, Niclas. Machine learning: a concise overview. In: DATA Science in Practice. [S.l.]: Springer, 2019. p. 27–58.

JAHANGIRI, Arash; RAKHA, Hesham. Applying Machine Learning Techniques to Transportation Mode Recognition Using Mobile Phone Sensor Data. IEEE Transactions on Intelligent Transportation Systems, 2015.

JUNIOR, Caio Schamber. GPS - Sistema de Posicionamento Global. UNIPAR - Universidade Paranaense, 2008.

KOEHRSEN, William. **Random Forest Simple Explanation**. Disponível em: <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>. (accessed: 28.11.2018).

MAZIMPAKA, Jean Damascène; TIMPF, Sabine. Trajectory data mining: A review of methods and applications. **pp-62-69**, 2016.

MICHEL, Tom M. **Machine Learning**. [S.l.]: McGraw-Hill Science/Engineering/Math; 1997.

O' DANIEL, Thomas; HUSSIEN, Mohammed; ABDULLA, Raed. Localization using GPS Coordinates in IPv6 Addresses of Wireless Sensor Network Nodes. **Indian Journal of Science and Technology**, v. 9, mar. 2016. DOI: 10.17485/ijst/2016/v9i10/88985.

ZHANG, Xiao. **Como escolher algoritmos do Azure Machine Learning Studio**. Disponível em: <https://docs.microsoft.com/pt-br/azure/machine-learning/studio/algorithm-choice>. (accessed: 27.11.2018).

ZHENG, Yu; CHEN, Yunkun et al. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. Microsoft Research Asia, 2008.

ZHENG, Yu; LIU, Like et al. **Geolife GPS trajectory dataset**. 2011. Disponível em: <https://www.microsoft.com/en-us/research/publication/geolife-gps-trajectory-dataset-user-guide/>. (accessed: 08.06.2019).

_____. Understanding Transportation Modes Based on GPS Data for Web Applications. Microsoft Research Asia, 2010.

ZHENG, Yu; XIE, Xing; MA, Wei-Ying. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. Microsoft Research Asia, 2010.

APÊNDICE A – PARÂMETROS

Abaixo encontra-se a lista de hiperparâmetros utilizados no *GridSearchCV* para cada os modelos de árvores de decisão e *Random Forest*.

- ***Decision Tree*:**
  - $criterion$ - Define uma função que mede a qualidade da divisão em cada nó. Parâmetros utilizados no modelo: *Gini* e *Entropy*;
  - max_depth - Define a profundidade máxima da árvore. Parâmetros utilizados no modelo: 5, 10, 30 e 50;
  - $min\_samples\_split$ - Define o número mínimo de amostras para poder dividir um nó. Parâmetros utilizados no modelo: 0.5, 1.0, 2 e 3;
  - $min\_samples\_leaf$ - Define o número mínimo de amostras que define um nó folha. Parâmetros utilizados no modelo: 0.5, 1 e 2;
  - $min\_weight\_fraction\_leaf$ - Define a fração ponderada mínima da soma total dos pesos que define um nó folha. Parâmetros utilizados no modelo: 0, 0.3 e 0.5;
  - $max\_features$ - Define o número máximo de atributos a serem utilizados para definir a melhor divisão.Parâmetros utilizados no modelo: *Auto*, *Sqrt* e *Log2*;
  - max_leaf_nodes - Define o número máximo de nós folhas no modelo. Parâmetros utilizados no modelo: 10, 50, 100 e 200;
  - $min\_impurity\_decrease$ - Define o valor mínimo de impureza no nó. Parâmetros utilizados no modelo: 0.0, 0.3 e 0.5;
  - $class\_weight$ - Define os pesos associados as classes. Parâmetros utilizados no modelo: *Balanced*, [0:1, 1:1] e [0:1, 1:5].

- ***Hiperparâmetros escolhidos Decision Tree*:**
  - $criterion$ - *Entropy*;
  - max_depth - 10;
  - $min\_samples\_split$ - 2;
  - $min\_samples\_leaf$ - 1;
  - $min\_weight\_fraction\_leaf$ - 0;
  - $max\_features$ - *Auto*;
  - max_leaf_nodes - 100;
  - $min\_impurity\_decrease$ - 0.0;

  - *class_weight* - Define os pesos associados as classes. Parâmetros utilizados no modelo: [0:1, 1:1].

- ***Random Forest*:**

  - *criterion* - Define uma função que mede a qualidade da divisão em cada nó. Parâmetros utilizados no modelo: *Gini* e *Entropy*;
  - *max_depth* - Define a profundidade máxima da árvore. Parâmetros utilizados no modelo: 5, 10, 30 e 50;
  - *min_samples_split* - Define o número mínimo de amostras para poder dividir um nó. Parâmetros utilizados no modelo: 0.5, 1.0, 2 e 3;
  - *min_samples_leaf* - Define o número mínimo de amostras que define um nó folha. Parâmetros utilizados no modelo: 0.5, 1 e 2;
  - *min_weight_fraction_leaf* - Define a fração ponderada mínima da soma total dos pesos que define um nó folha. Parâmetros utilizados no modelo: 0, 0.3 e 0.5;
  - *max_features* - Define o número máximo de atributos a serem utilizados para definir a melhor divisão.Parâmetros utilizados no modelo: *Auto*, *Sqrt*, *Log2* e *None*;
  - *max_leaf_nodes* - Define o número máximo de nós folhas no modelo. Parâmetros utilizados no modelo: 10, 50, 100, 200 e *None*;
  - *min_impurity_decrease* - Define o valor mínimo de impureza no nó. Parâmetros utilizados no modelo: 0.0, 0.5 e 1.0;
  - *bootstrap* - Define se as amostras *bootstrap* serão usadas ao construir as árvores ou se todo o conjunto de dados será usado. Parâmetros utilizados no modelo: *True* e *False*;
  - *n_estimators* - Define o número de árvores da floresta. Parâmetros utilizados no modelo: 5, 10 e 20.

- ***Hiperparâmetros escolhidos Random Forest*:**

  - *criterion* - *Entropy*;
  - *max_depth* - 10;
  - *min_samples_split* - 2;
  - *min_samples_leaf* - 2;
  - *min_weight_fraction_leaf* - 0;
  - *max_features* - *Auto*;
  - *max_leaf_nodes* - *None*;
  - *min_impurity_decrease* - 0.0;

- *bootstrap* - *True*;
- *n_estimators* - 20.