INSTITUTO FEDERAL DE MATO GROSSO DO SUL

CAMPUS CORUMBÁ

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA A CLASSIFICAÇÃO DE CAVALOS PANTANEIROS E ESTUDO COMPARATIVO DO DESEMPENHO DOS ALGORITMOS NAIVE BAYES, KNN E C4.5 NESTA CLASSIFICAÇÃO

SOUMAYA OUNKHIR

ORIENTADOR: PROF. ME. MARCEL JOSÉ SOLEIRA GRASSI

COORIENTADOR: PROF. DR. DIEGO SAQUI

Corumbá – MS

2020

DESENVOLVIMENTO DE UMA APLICAÇÃO PARA A CLASSIFICAÇÃO DE CAVALOS PANTANEIROS E ESTUDO COMPARATIVO DO DESEMPENHO DOS ALGORITMOS NAIVE BAYES, KNN E C4.5 NESTA CLASSIFICAÇÃO

Soumaya Ounkhir

Trabalho de Conclusão de Curso apresentado ao curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, do instituto Federal de Mato Grosso do Sul - Campus Corumbá, como requisito para obtenção do título de Tecnóloga em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Marcel José Soleira Grassi

Coorientador: Prof. Dr. Diego Saqui

Corumbá – MS

2020

Resumo

O Cavalo Pantaneiro é uma raça que conseguiu se adaptadar à região do Pantanal, que possui características climáticas particulares, sendo este animal de extrema importância no manejo do gado, lazer e esportes como corrida e salto. Ao longo dos anos, essa raça tornou-se bastante valorizada. O núcleo de pesquisa da Empresa Brasileira de Pesquisa Agropecuária – EMBRAPA do Pantanl trabalha em parceria com Associação Brasileira de criadores de Cavalo Pantaneiro (ABCCP) para conservar esta raça e estabelecer um padrão racial, efetuando registros de 15 medidas morfométricas e outras características físicas dos cavalos pantaneiros. A fim de auxiliar os diversos usuários como técnicos, investidores e interessados nos processos de tomada de decisão e seleção animal, a presente pesquisa possui como objetivo desenvolver uma aplicação capaz de classificar os cavalos pantaneiros em alta, média e baixa qualidade. Para tanto, utilizou-se técnicas de aprendizado de máquina, comparando o desempenho dos algoritmos K-Nearest Neighbor (KNN), Naive Bayes (NB) e Árvore de Decisão C4.5 na classificação dos cavalos. Os resultados mostram que o KNN teve as melhores taxas de acerto. Além disso, com o método de seleção características, foi possível reduzir o número das medidas morfométricas em oito medidas para classificar os cavalos.

Palavras-chave: Cavalo Pantaneiro, padrão racial, aprendizado de máquina.

Abstract

The Pantaneiro Horse is a breed that can adapt to the Pantanal region, which has particular features and this animal is of extreme importance in cattle management, leisure and sports such as running and jumping. Over the years, this breed has become highly valued. The Brazilian Agricultural Research Corporation EMBRAPA works in partnership with the Brazilian Association of Pantaneiro Horse Breeders ABCCP to conserve this breed and establish a racial pattern, making records of 15 morphometric measurements and other physical features of the Pantaneiro horses. In order to assist the several users such as technicians, investors and interested in the decision making and animal selection processes, this research aims to create an application capable of classifying Pantaneiro horses in high, medium and low quality. For this purpose, machine learning techniques were used, comparing the performance of the K-Nearest Neighbor (KNN), Naive Bayes (NB) and Decision Tree C4.5 algorithms in the classification of horses. The results show that the KNN had the best hit rates. In addition, with the feature selection method used, it was possible to reduce the number of morphometric measurements in eight measures to classify the horses.

Keywords: Pantaneiro horse, racial pattern, machine learning.

Lista de Abreviaturas e siglas

ABCCP	Associação Brasleira de Criadores de Cavalos Pantaneiros
AM	Aprendizado de Máquina
CPAP	Centro de Pesquisa de Agropecuária do Pantanal
CSS	Cascading Style Sheet
CSV	Comma Separated Values
CV	Cross Validation
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
HTML	Hyper Text Markup Language
IA	Inteligência Artificial
IBGE	Instituto Brasileiro de Geografia e Estatística
JS	JavaScript
KNN	K-Nearest Neighbor
NB	Naive Bayes
TI	Tecnologia da Informação
UFMS	Universidade Federal de Mato Grosso do Sul
UFMT	Universidade Federal de Mato Grosso
W3C	Consórcio World Wide Web

Lista de Figuras

Figura 1 - Medidas lineares laterais do cavalo 16
Figura 2 - Medidas lineares da cabeça e pescoço do cavalo 17
Figura 3 - Medidas lineares do peito, anca e canela do cavalo 17
Figura 4 - Exemplo de construção de algoritmo de aprendizado supervisionado 21
Figura 5 - Gráfico dos objetos e suas características 24
Figura 6 - Gráfico de objetos e suas características em situação K = 5 25
Figura 7 - Exemplo do jogo de Tênis em um modelo de árvore de decisão 29
Figura 8 - Ilustração do funcionamento do algoritmo Random Forest 30
Figura 9 - Objetos com classe desconhecida 32
Figura 10 - Objetos após Clustering 32
Figura 11 - Dados não rotulados com três centroides definidos 33
Figura 12 - Três clusters com seus centroides finais 34
Figura 13 - Iterações Cross-Validation para k-folds = 5 36
Figura 14 - Método ShuffleSplit usando 5 folds e 10% de cada grupo para teste 36
Figura 15 - Queda de desempenho do classificador com o aumento de características 37
Figura 16 - Fluxo do trabalho completo 45
Figura 17 - Base de dados em arquivo .csv 47
Figura 18 - Planilha da base de dados em arquivo .xlsx 47
Figura 19 - Caso de uso do aplicativo 49
Figura 20 - Gráfico do grau de importância de cada característica em ordem 51
Figura 21 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 2 folds 54
Figura 22 - Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 2 folds 55
Figura 23 - Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 5 folds 56
Figura 24 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 10 folds 57
Figura 25 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 7 medidas com o aumento de folds de 2 a 10 57
Figura 26 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 8 folds 58
Figura 27 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 9 folds 58
Figura 28 - Interface principal do aplicativo para inserir as medidas do cavalo 60
Figura 29 - Interface do aplicativo mostrando a classificação do cavalo 61

Lista de Tabelas

Tabela 1 - Número de indivíduos, média e desvio padrão das medidas do cavalo Pantaneiro de ambos os sexos definidos pela ABCCP adaptado pelo autor baseado em (Santos et al., 1995) 18
Tabela 2 - Exemplo de base de dados para Aprendizado Supervisionado 22
Tabela 3 - Tabela com os atributos e classe de cada objeto 26
Tabela 4 - Resultado da classe da instância 11 usando k = 5 27
Tabela 5- Grau de importância de cada característica 50
Tabela 6 - Matriz de confusão da classificação do algoritmo KNN 59
Tabela 7 - Resultado da configuração obtida para classificar os cavalos pantaneiros ... 59

Sumário

1. Introdução 9

1.1. Objetivos 12

1.1.1. Objetivo geral 12

1.1.2. Objetivos específicos 12

2. Referencial teórico 13

2.1. Cavalo Pantaneiro 13

2.1.1. História do Cavalo Pantaneiro 13

2.1.2. Conservação do cavalo Pantaneiro 14

2.1.3. Características morfológicas do cavalo Pantaneiro 15

2.2. Aprendizado de Máquina 20

2.2.1. Aprendizado Supervisionado 20

2.2.1.1. Naive Bayes 22

2.2.1.2. K-Nearest Neighbor (KNN) 24

2.2.1.3. Árvore de Decisão C4.5 28

2.2.1.4. Random Forest 30

2.2.2. Aprendizado Não Supervisionado 31

2.2.2.1. K-Means 33

2.2.3. Avaliação e Testes de Classificadores 35

2.2.3.1. Holdout 35

2.2.3.2. Validação Cruzada (*Cross Validation*) 35

2.2.4. Maldição de Dimensionalidade 37

2.2.4.1. Seleção de características 38

2.3. Ferramentas utilizadas 39

2.3.1. Linguagens de Programação 39

2.3.1.1. Python 40

2.3.1.2. JavaScript 41

2.3.1.3. HTML e CSS 41

2.3.2. Bibliotecas 42

2.3.2.1. Pandas 42

2.3.2.2. Scikit-Learn 42

2.3.3. Ionic Framework 43

2.3.4. Anaconda Navigator 43

3. Materiais e Métodos 45

3.1. Conjunto de dados utilizados 46

3.2. Procedimentos 48

4. Experimentos e Resultados 50

5. Considerações Finais 62

Referências 64

ANEXOS 68

ANEXO A - Número de indivíduos, média, desvio padrão e coeficiente de variação das características de conformação, índices zootécnicos e relações corporais de equinos machos e fêmeas, durante os períodos de 26/07/72 a 29/09/91. 68

ANEXO B – Número de indivíduos, média, desvio padrão e coeficiente de variação das características de conformação, índices zootécnicos e relações corporais de equinos machos e fêmeas, durante os períodos de 26/07/72 a 29/09/91. (Continuação) 68

APÊNDICES 69

APÊNDICE A - Gráficos da taxa de acerto do KNN, para k vizinhos de 1 a 10, para os conjunto de 4 a 15 medidas, usando de 2 a 10 folds 69

APÊNDICE B - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando de 2 a 10 folds 122

1. Introdução

A equideocultura - ramo que abrange a criação de cavalos, asnos, jumentos, burros e jegues - exerce um importante papel na formação do Brasil, tanto nos aspectos econômicos e políticos, quanto sociais do país. Na óptica econômica, o cavalo desempenha diferentes funções como de sela (montaria), transporte de carga e tração (trabalhos pesados). Além disso, vê-se seu amplo uso na área de lazer e esportes como corrida e salto (Lima et al., 2015). Na agropecuária, somente o mercado de cavalos faturou mais de 16 bilhões de reais no ano de 2015 e avança 12% a cada ano que passa (IBGE, 2015), gerando mais de 3 milhões de empregos em todo o país, sendo destes 610 mil diretos e 2.430 mil indiretos (Lima et al., 2015).

O total do rebanho no país é de 5,55 milhões de cabeças em 2015, configurando um aumento de 1,8% em relação ao ano anterior. Destes 20,9% são do Centro-Oeste, terceiro maior rebanho equino do Brasil. Em nível municipal, destaca-se a cidade de Corumbá (MS), região pantaneira, com o maior efetivo de equinos de todo o país, ocupando, desta maneira, a primeira colocação (IBGE, 2015).

Com relação às raças, existem inúmeras em todo o território nacional, com diferentes características, entretanto, será dado o foco, neste trabalho, para a raça Pantaneira pela sua importância social e econômica para a região. A Empresa Brasileira de Pesquisa Agropecuária - Embrapa e a Associação Brasileira dos Criadores de Cavalo Pantaneiro – ABCCP realiza estudos com o objetivo de conservar esta raça (Santos et al., 2005).

O cavalo Pantaneiro é considerado o melhor do brasil no que se refere à capacidade de resistir na região pantaneira. Pois este espaço apresenta diversas características complexas para a adaptação de outros tipos de equinos. Esta região é, periodicamente, inundável e apresenta grande variabilidade espacial e temporal (Santos et al., 2005). Ora é muita água, ora é muita seca, além de outros fatores como altas temperaturas, insetos, predadores, entre outros (Santos et al., 2016).

A principal atividade econômica local é constituída pela exploração extensiva do gado de corte, em que o cavalo Pantaneiro é ator principal no manejo do rebanho. Além de ser usado, fortemente, no transporte local. Apesar de essa raça possuir uma população

ainda, relativamente, baixa (cerca de 2.400 cabeças, no ano de 2003) que é cadastrada na ABCCP e se enquadra no padrão racial definido por esta (Santos et al., 2005), percebe-se, atualmente, um grande aumento (cerca de 5.000 cabeças) e interesse por estes animais, principalmente pela alta qualidade dos cavalos expostos em feiras e em leilões, assim como também pelo seu desempenho no campo esportivo (Santos et al., 2016).

Para determinar se o cavalo é puramente da raça Pantaneira e de alta qualidade, a ABCCP estabelece alguns critérios para o seu padrão racial. Entre eles, constam 15 medidas lineares tomadas no registro do animal, além de outras características físicas (Santos et al., 2016). Já que a maior parte da população dos cavalos que existe na região pantaneira são incompatíveis com o padrão estabelecido por esta associação e mestiços resultantes de cruzamentos com outros tipos de raças desconhecidas (Santos et al., 1995). Desta maneira, apenas especialistas são capazes de realizar esta análise.

Para os criadores, investidores e pesquisadores dessa raça, portanto, é de suma importância manter viva essa raça tão valiosa. Sendo assim, o uso da Tecnologia de Informação (TI) mostra-se como fator poderoso para ajudar a manter a qualidade dessa raça, automatizando a coleta e fornecimento dos dados para o correto controle.

Dessa forma, alguns desses métodos dentro da área de TI, pode ser o aprendizado de máquina (AM), ramo que vem ganhando cada vez mais espaço. AM é caracterizado como modelo computacional com alta capacidade de aprendizagem, capaz de realizar predição de informações e padrões de comportamento (MITCHELL, 1997), tornando-se um bom recurso no fornecimento de uma solução satisfatória.

Diante do contexto apresentado, o presente trabalho tem como objetivo o desenvolvimento de uma aplicação capaz de determinar a qualidade destes animais e auxiliar na tomada de decisão e seleção dos dados da raça do cavalo pantaneiro por meio de técnicas de aprendizado de máquina.

O trabalho encontra-se organizado da seguinte maneira: o Capítulo 1 apresenta a contextualização e objetivos do trabalho e no Capítulo 2, são apresentados os principais conceitos sobre o cavalo pantaneiro, aprendizado de máquina e ferramentas de aprendizado que auxiliam no desenvolvimento do trabalho, além de classificadores que fazem parte dos conhecimentos necessários para o bom entendimento deste trabalho.

Já no Capítulo 3, é feita, de uma forma geral, a descrição do método e aplicativo

abordado e desenvolvido no presente trabalho.

A análise dos resultados obtidos com os experimentos está descrita no Capítulo 4 e por fim, no Capítulo 5 são apresentadas as considerações finais sobre o desenvolvimento do trabalho, suas contribuições e sugestões para trabalhos futuros.

1.1. Objetivos

O presente capítulo aborda todos os objetivos do presente trabalho, tanto o objetivo geral quanto os específicos.

1.1.1. Objetivo geral

Desenvolver uma aplicação capaz de classificar a qualidade do cavalo da raça pantaneira em: alta, média e baixa qualidade.

1.1.2. Objetivos específicos

- Avaliar o desempenho dos algoritmos Naive Bayes, KNN (K Nearest Neighbor) e C4.5 (Árvore de Decisão) para classificação do cavalo pantaneiro;
- Realizar a seleção de caraterísticas, diminuindo o número de medidas necessárias para avaliação da qualidade do cavalo Pantaneiro;
- Tornar este conhecimento mais acessível ao público interessado e
- Auxiliar o usuário na tomada de decisão.

2. Referencial teórico

2.1. Cavalo Pantaneiro

Este capítulo aborda os principais conceitos, informações históricas e os métodos adotados para conservar o cavalo pantaneiro.

2.1.1. História do Cavalo Pantaneiro

Os primeiros cavalos que chegaram à região pantaneira, há mais de dois séculos, foram capazes de se adaptarem às condições bioclimáticas da região. Com a formação das fazendas no Pantanal, o cavalo Pantaneiro tornou-se um fator de importância econômica e social no Pantanal Mato-Grossense, por ser essencial no manejo de gado. Diariamente, os peões percorrem as invernadas para verificarem o estado do gado. Desta forma, estes animais são capazes de suportar marchas por períodos prolongados em áreas alagadas, contribuindo para a integração e fixação do homem à terra (Balieiro, 1971; Corrêa Filho, 1973; Beck.1985).

A raça surgiu originalmente na parte central da América do Sul, entretanto, não há uma data exata da sua chegada à região. Portanto, ainda há a necessidade de um conhecimento mais preciso sobre a origem do cavalo Pantaneiro. Segundo pesquisadores esta raça é, provavelmente, oriunda de cruzamentos de equinos de origem lusitana (Céltico, Barba e Andaluz), do Acabe e do Crioulo Argentino, sob pressão da seleção natural (Domingues,1957; Corrêa Filho, 1973).

Segundo Balieiro (1971) os municípios em que houve a formação da raça dos cavalos Pantaneiro são os listados a seguir: Santo Antônio do Leverger, Barão de Melgaço, Nossa Senhora do Livramento, Poconé, Cáceres, Corumbá e Aquidauana. Porém, por volta do ano de 1900, iniciou-se a influência de outras raças, tais como Acabe e Puro - Sangue - Inglês a fim de melhorar a raça e elevar o porte, porém sem critérios de cruzamento e orientação técnica, causando diminuição do rebanho equino pantaneiro.

2.1.2. Conservação do cavalo Pantaneiro

O primeiro estudo sobre o cavalo Pantaneiro foi realizado por Domingues (1957), que pesquisou a população equina dos pantanais do norte (Poconé, Cáceres, Santo Antônio do Leverger e Barão de Melgaço). Ele, juntamente com os criadores de Poconé foram responsáveis por criarem um padrão provisório da raça pantaneira.

Diante dos fatores que estavam contribuindo para a extinção do cavalo Pantaneiro (cruzamento sem orientação técnica e doenças), foi fundada em 1972 a Associação Brasileira dos Criadores de Cavalo Pantaneiro (ABCCP). Esta teve como principais objetivos congregar os criadores, organizar e manter o registro genealógico da raça, fomentar a criação e estudar todos os assuntos referentes à raça (Santos et al., 1995).

Ainda segundo Santos et al. (1995), na década de 80, alguns órgãos governamentais, como a Secretaria da Agricultura de Mato Grosso do Sul (SECAP-MS), Universidade de Mato Grosso (UFMT), Universidade de Mato Grosso do Sul (UFMS) e o Centro de Pesquisa Agropecuária do Pantanal (CPAP-EMBRAPA), iniciaram trabalhos de conservação e melhoramento da raça do cavalo Pantaneiro. Para tanto, o CPAP mantém um núcleo de criação de Cavalos Pantaneiros na fazenda Nhumirim, na sub-região da Nhecolândia, que, permite conservar a raça, realizar pesquisas em reprodução, nutrição, genética, seleção e melhoramento, parasitologia, virologia, dentre outras.

Além disso, o CPAP, juntamente à ABCCP, realiza o levantamento dos núcleos de criação do Cavalo Pantaneiro nas diversas sub-regiões do Pantanal, caracterizando genética e fenotipicamente as populações equinas existentes e identificando os sistemas de criação adotados pelos criadores em cada local. O CPAP também se preocupa em estudar a origem do Cavalo Pantaneiro. Portanto, os técnicos possuem, além disso, a função de pesquisar o acervo, não somente das principais bibliotecas do país, mas também do exterior (Santos et al., 1995).

Como foi explanado, anteriormente, o cavalo Pantaneiro possui diversos usos e é extremamente valorizado. Sem dúvida o seu principal uso é na lida do gado. São inúmeras as pesquisas e estudos feitos acerca do cavalo Pantaneiro, mostrando seu valor, capacidade e resistência.

Por exemplo, a pesquisa de da Silva et al. (2005) teve como objetivo avaliar a frequência cardíaca e respiratória (indicadores de adaptação) de cavalos Pantaneiros

usados na lida diária do gado, na sub-região da Nhecolândia. Para tanto, no período de 5 a 8 de julho de 2003, foram registradas a frequência respiratória e frequência cardíaca antes, imediatamente depois e 30 minutos após o trabalho com gado de 12 cavalos, sendo 7 testados duas vezes, num total de 19 observações. Após todos os testes, os pesquisadores concluíram que os cavalos Pantaneiros usados rotineiramente no trabalho de gado são adaptados ao estresse do exercício.

Além disso, segundo a pesquisa de Santos et al. (2001), o cavalo Pantaneiro é extremamente resistente a longas caminhadas. Já o estudo de McManus (2001), mostra que esta raça, além de suportar as longas distâncias, este animal, por possuir um bom desenvolvimento torácico, é capaz de atingir altas velocidades para manejar o gado.

2.1.3. Características morfológicas do cavalo Pantaneiro

Segundo Langlois et. al. (1983), existem duas categorias de seleção do animal: a seleção direta, que analisa a habilidade do animal na prática. O segundo tipo é a seleção indireta, que mede uma característica do animal em relação a outras. Porém os critérios utilizados para medir a habilidade do animal nem sempre são objetivos.

De acordo com Santos et al. (1995), um dos instrumentos de seleção baseia-se na avaliação fenotípica, ou seja, se refere a características externas que podem ser observadas visualmente e medidas. Além disso, eles incluem performance e taxa de conformação.

A conformação corporal está relacionada à performance, e por isso, deverá ser incluída em todos os processos de seleção. Um sistema esquelético correto, especialmente pés e pernas torna-se essencial para a boa performance (Santos et al., 1995). Segundo Giannoni (1988) a afirmação “forma prediz função" é um tanto genérica, visto que o desempenho dos equinos é influenciado por outros fatores, tais como morfológicos, fisiológicos, psíquicos e de ambiente. Portanto, a forma é apenas um indicativo da aptidão zootécnica do animal.

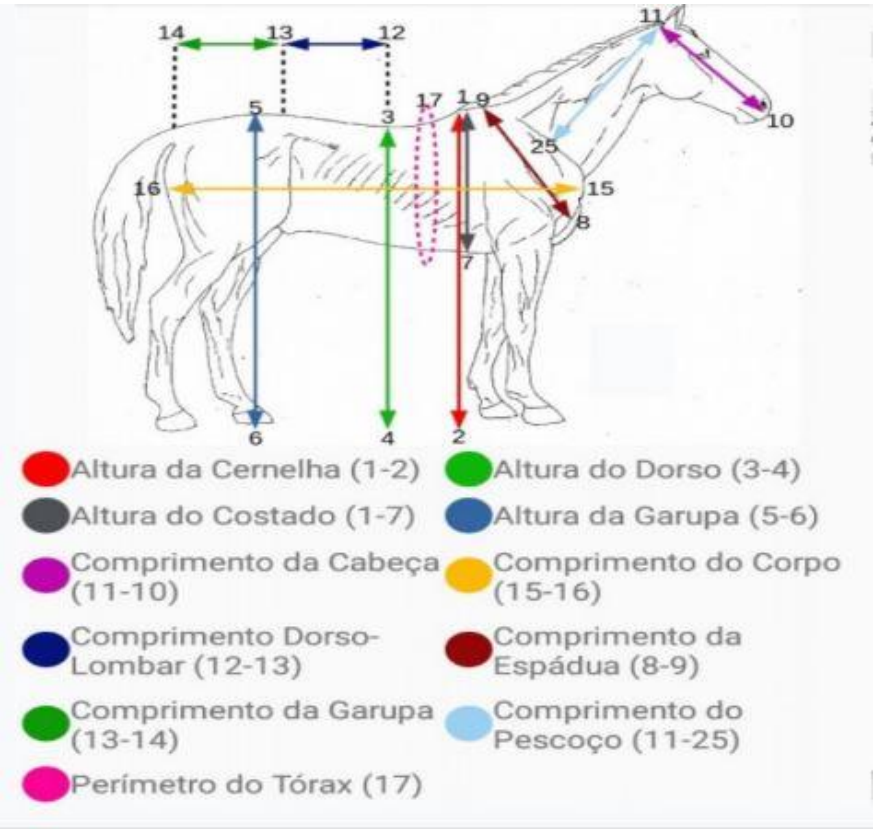
A conformação corporal do cavalo pode ser estudada e analisada por meio do uso de medidas corporais, que abrangem proporções, dimensões e relações entre as diversas partes. No estudo das proporções, consideram-se as lineares, angulares, as de

continuidade, as gerais, as variações e as compensações (Santos et al., 1995). Na presente pesquisa, o foco é dado exclusivamente às medidas lineares e algumas relações corporais.

A fim de avaliar as medidas e relações corporais e outras características do cavalo pantaneiro, a ABCCP realizou um estudo dos equinos pantaneiros registrados. Foram utilizados para avaliação dados contidos nas fichas de registro definitivo na ABCCP durante os períodos de 26 de julho de 1972 a 29 de setembro de 1991. Estas fichas possuíam as seguintes informações sobre as medidas corporais: altura da cernelha, altura do dorso, altura da garupa, altura dos costados, comprimento da cabeça, comprimento do pescoço, comprimento dorso-lombo, comprimento da garupa, comprimento da espádua, comprimento do corpo, largura da cabeça, largura do peito, largura das ancas, perímetro de tórax e da canela, somando um total de 1.215 registros, naquela data (Santos et al., 1995).

As Figuras 1, 2 e 3 ilustram todas as 15 medidas corporais citadas.

Figura 1 - Medidas lineares laterais do cavalo



A Figura 1 mostra todas as medidas corporais lineares laterais do cavalo pantaneiro. Por meio desta imagem é possível verificar as alturas da cernelha, costados,

garupa, dorso, comprimentos da cabeça, do corpo, dorso-lombar, espádua, garupa, pescoço e o perímetro do tórax. Já a Figura 2 traz os comprimento e largura da cabeça e o comprimento do pescoço.

Figura 2 - Medidas lineares da cabeça e pescoço do cavalo

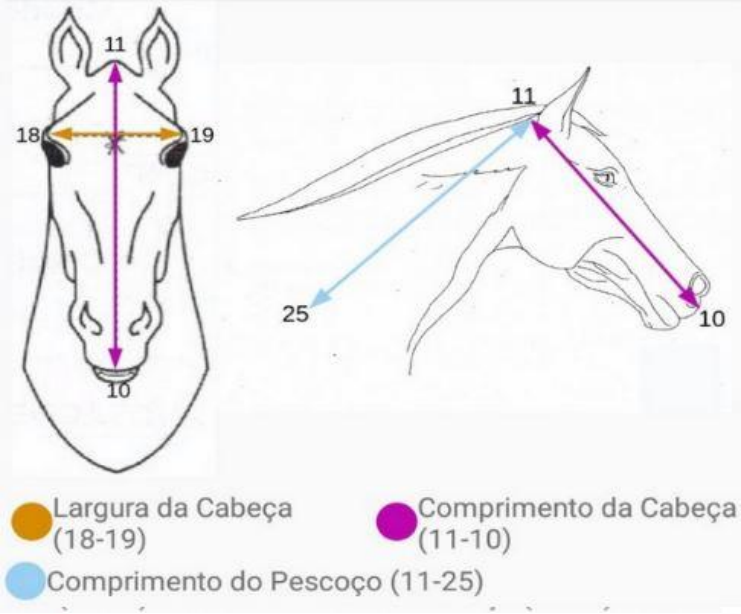
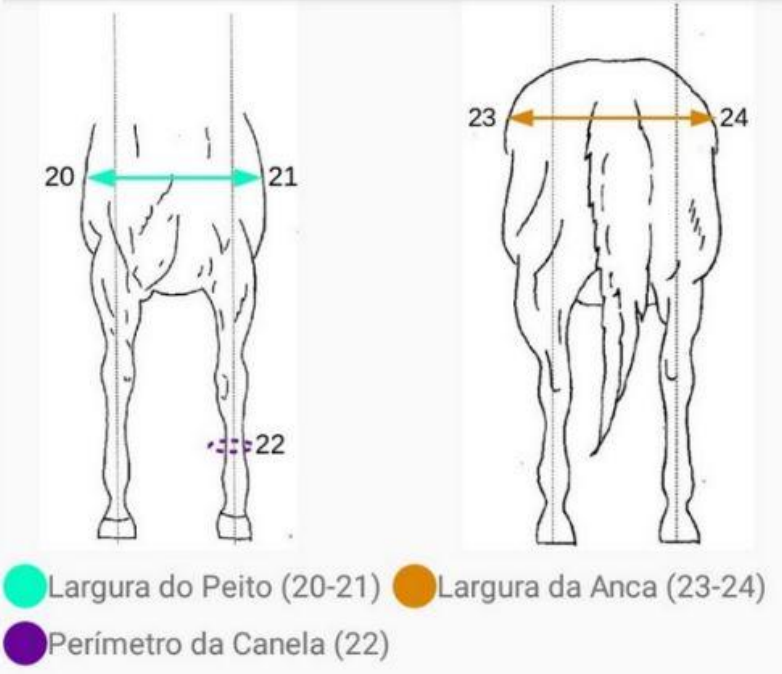


Figura 3 - Medidas lineares do peito, anca e canela do cavalo



Na Figura 3, verifica-se as larguras do peito e anca e também o perímetro da canela.

Alguns dos resultados obtidos pela pesquisa da ABCCP estão representados na tabela 1. Nela, pode ser visto o número de indivíduos, a média obtida para cada medida linear. Ou seja, a média da medida que o cavalo pantaneiro deve possuir para ser classificado e registrado como Pantaneiro. Também pode ser visto o desvio padrão dos resultados das medidas de ambos os sexos.

Tabela 1 - Número de indivíduos, média e desvio padrão das medidas do cavalo Pantaneiro de ambos os sexos definidos pela ABCCP adaptado pelo autor baseado em (Santos et al., 1995)

	Machos			Fêmeas		
Variáveis observadas	**Número de indivíduos**	**Média (cm)**	**Desvio padrão**	**Número de indivíduos**	**Média (cm)**	**Desvio padrão**
Altura da cernelha	180	141,7	1,7	1035	136,9	2,2
Altura do dorso	180	138,2	2,6	1032	134,5	2,7
Altura da garupa	180	141,8	2,2	1034	137,9	2,5
Altura do costado	180	54,4	2,5	1031	58,5	3,5
Comprimento da cabeça	180	54,4	2,5	1034	53,6	2,5
Comprimento do pescoço	180	57,7	4,9	1034	57,5	6
Comprimento do dorso	147	50,4	8,1	914	42,6	9,6
Comprimento da garupa	180	45,2	2	1034	44,9	2,8
Comprimento da espádua	180	47,6	3,6	1034	46,2	3,4
Comprimento do corpo	180	146,2	22,4	1033	141,3	5,1
Largura da cabeça	180	20,9	2	1033	20,5	2,5
Largura do peito	180	34,1	3,1	1034	31,6	3
Largura das ancas	180	43,5	2,7	1034	43,5	3

Todos os resultados completos obtidos mais detalhadamente pela ABCCP se encontram no Anexo A e B.

Desta forma, todos os cavalos que se encaixam nos critérios descritos pela ABCCP podem ser considerados da raça pantaneira e registrados na associação. Vale destacar que o registro do animal não significa que ele seja de alta qualidade. Podendo ser enquadrado na classe de média qualidade e dentro dos padrões estabelecidos. Também é importante salientar que há uma rigorosa proporcionalidade entre as medidas que deve ser atendida para o animal se encaixar nos padrões.

Além disso, segundo o estudo de Miserani et al. (2001) que faz uma avaliação dos Fatores que influenciam as medidas lineares do cavalo Pantaneiro, a melhor época para avaliar as medidas morfométricas dos animais é a partir dos 4 anos de idade, pois os animais entre 2 e 3 anos ainda se encontram em fase de crescimento.

Outro fator relevante em relação às regras estabelecidas é que, embora todas as medidas atinjam o valor desejado, isso não garante que o cavalo seja de alta qualidade e vença, por exemplo, em um concurso de leilões de cavalos pantaneiros. A avaliação do animal durante este tipo de eventos depende de um juiz experiente que realiza a avliação do animal visualmente, ou seja, sem medir o cavalo, portanto a avaliação é bastante subjetiva.

Por estas razões, decidiu-se utilizar o aprendizado de máquina para determinar a qualidade dos cavalos Pantaneiros, já que este método é capaz de aprender a subjetividade e os padrões por trás dos dados fornecidos. Em outras palavras, a máquina consegue imitar o homem para classificar os cavalos da maneira de um juiz, por exemplo. Para isso, basta apresentar ao algoritmo dados de cavalos de alta, média e baixa qualidade, que foram classificados por humanos, para aprender as características e diferencial de cada uma destas classes. A grosso modo, uma vez “aprendido” como é o cavalo de cada uma das classes, o algoritmo é capaz de prever a qual classe pertence um cavalo novo nunca analisado por ele antes.

O seguinte capítulo aborda os principais conceitos do aprendizado de máquina, a suas categorias, alguns algoritmos de classificação empregados na presente pesquisa e exemplos didáticos.

2.2. Aprendizado de Máquina

O Aprendizado de Máquina (AM) é uma subárea da Inteligência Artificial (IA) que tem como objetivos desenvolver técnicas computacionais, capazes de simular o processo de aprendizado e a construção de sistemas que podem adquirir conhecimento automaticamente (Mitchell, 1997). Os algoritmos de Aprendizado de Máquina (AM) utilizam experiências anteriores, denominadas casos ou exemplos, para auxiliar o processo de tomada de decisão e melhorar seu desempenho.

Segundo Mitchell (1997), existem três componentes básicos no Aprendizado de Máquina (AM). O primeiro é a tarefa a ser realizada e o segundo é a fonte de experiência (banco de dados). De forma geral, quanto mais dados disponíveis, melhor é o desempenho e aprendizado do programa.

Já o terceiro e último componente é a medida de desempenho que serve para mostrar se o desempenho está melhorando ou piorando conforme esteja usando a experiência disponível.

Os três componentes do presente trabalho são exemplificados a seguir:

- Tarefa: classificação dos cavalos pantaneiros em alta, média e baixa qualidade.
- Fonte de experiência: base de dados de cavalos pantaneiros com a sua respectiva classificação.
- Medida de desempenho: porcentagem de cavalos classificados corretamente.

No estudo do AM, há dois tipos clássicos de aprendizado: Aprendizado supervisionado e Aprendizado não supervisionado que são descritos abaixo, brevemente.

2.2.1. Aprendizado Supervisionado

A principal diferença entre o Aprendizado Supervisionado e Não-Supervisionado é a presença ou não do atributo classe, que rotula os exemplos do conjunto de dados fornecido ao algoritmo, denominado conjunto de treinamento. Dessa maneira, no Aprendizado Supervisionado, sabe-se a classe de cada dado deste conjunto que foi

previamente determinado por um humano (Chapelle et al., 2009).

De forma geral, neste modo de aprendizado, os dados já classificados são fornecidos ao algoritmo (classificador) a fim de este ser estruturado, adquirindo a experiência e aprendizado necessários, para identificar e classificar, automaticamente, novos dados não rotulados.

A Figura 4 ilustra um fluxograma básico da construção de um novo classificador de aprendizado supervisionado. Primeiro são definidos os dados que irão ser utilizados para treino do classificador. A partir destes dados, o classificador é moldado e construído. Por fim, tem-se como resultado o classificador final pronto para ser testado com novos dados não rotulados.

Figura 4 - Exemplo de construção de algoritmo de aprendizado supervisionado

Um exemplo clássico de uma base de dados para o aprendizado supervisionado é apresentado na Tabela 2. Por meio desta, é possível observar as características, também denominadas atributos (Dia, Aspecto, Temperatura, Umidade e Vento). A última coluna se refere à classe (Decisão) de cada amostra do conjunto de dados, havendo apenas duas classes: “Sim” e “Não” que já estão pré-definidas. Dessa forma, o algoritmo, por meio destes dados, é capaz de determinar se a pessoa pode ou não jogar Tênis.

O seguinte capítulo apresenta o primeiro algoritmo de classificação usado nesta pesquisa que é o Naive Bayes. O capítulo traz uma definição básica do classificador e um exemplo de como o mesmo funciona.

Tabela 2 - Exemplo de base de dados para Aprendizado Supervisionado

Aspecto	Temperatura	Umidade	Vento	Decisão
Sol	Quente	Alta	Fraco	Não
Sol	Quente	Alta	Forte	Não
Nublado	Quente	Alta	Fraco	Sim
Chuva	Agradável	Alta	Fraco	Sim
Chuva	Fria	Normal	Fraco	Sim
Chuva	Fria	Normal	Forte	Não
Nublado	Fria	Normal	Forte	Sim
Sol	Agradável	Alta	Fraco	Não
Sol	Fria	Normal	Fraco	Sim
Chuva	Agradável	Normal	Fraco	Sim
Sol	Agradável	Normal	Forte	Sim
Nublado	Agradável	Alta	Forte	Sim
Nublado	Quente	Normal	Fraco	Sim
Chuva	Agradável	Alta	Forte	Não

2.2.1.1. Naive Bayes

O Naive Bayes (NB) é um algoritmo baseado no Teorema de Bayes de probabilidade condicional. A probabilidade condicional é a medida de probabilidade acontecer, dado que outro evento específico tenha acontecido (JOHN; LANGLEY, 1995).

Partindo do teorema de Bayes, a função é como descrita na equação 1 abaixo:

$$P(C|D) = \frac{P(D|C) * P(C)}{P(D)} \quad (1)$$

Em que P é a probabilidade; C é a classe; D é o vetor de dados; P(C|D) é a probabilidade da classe C dado o vetor D e é chamada de probabilidade a posteriori de C; P(D|C) se refere à probabilidade do vetor D dada a classe C e é chamado de probabilidade condicional de D; P(C) é a probabilidade de C ocorrer (ou a priori da classe c) e por último P(D) é a probabilidade a priori do vetor de treinamento D.

A seguir, tem-se um exemplo prático de como calcular a probabilidade de ocorrência de meningite em pacientes. Sabe-se que a meningite causa torcicolo em 50% dos casos. Além disso, a probabilidade a priori de qualquer paciente ter meningite é 1/50000 e a probabilidade a priori de qualquer paciente ter rigidez de nuca é de 1/20. Portanto, deseja-se saber qual é a probabilidade a posteriori de um paciente ter meningite, se ele tem rigidez de nuca.

Dado que M é meningite e R a rigidez no pescoço, tem-se a resolução a seguir na equação 2:

$$P(M|R) = \frac{P(R|M) * P(M)}{P(R)} \quad (2)$$

$$P(M|R) = \frac{0{,}5 * (1/50000)}{1/20}$$

$$P(R) = 0{,}0002$$

Dessa maneira, tem-se que a probabilidade de ocorrer meningite dada a ocorrência de rigidez na nuca igual a 0,0002.

O teorema de Bayes nos permite calcular a probabilidade a posteriori para chegar o mais perto possível da priori. A priori é a probabilidade dada sem conhecimento de qualquer outro evento, seria a classificação perfeita. A posteriori é a probabilidade condicional que é atribuída quando um evento relevante é considerado para uma determinada classe, conforme mostra a equação 3.

$$ProbPosteriori = [ProbPriori * DistProb] / Evidência \quad (3)$$

A principal característica deste algoritmo é o fato de não levar em consideração a correlação entre as variáveis, ou seja, ele as trata de forma independente. Por isso, ele recebe o nome de Naive, que em inglês significa ingênuo.

O próximo capítulo traz outro algoritmo de classificação, também usado na presente pesquisa.

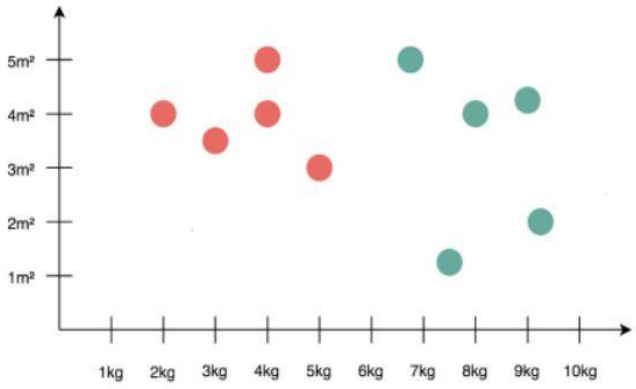
2.2.1.2. K-Nearest Neighbor (KNN)

Uma forma de predizer a classe de um novo dado não rotulado consiste em comparar esse dado com outros cuja classe é conhecida e atribuir a classe do caso mais próximo. Por exemplo, um médico, por meio de um conjunto de sintomas que descrevem o estado de saúde de um paciente, pode se basear em fichas médicas de pacientes já diagnosticados com sintomas similares a este, no intuito de auxiliar no diagnóstico de determinada doença (Ferrero 2009).

O algoritmo K-Nearest Neighbor (KNN) ou K-Vizinhos Mais Próximos é um algoritmo de aprendizado supervisionado. A ideia geral desse algoritmo consiste em encontrar os k dados rotulados mais próximos do dado não classificado e, com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa à classe do exemplo não rotulado (Ferrero 2009).

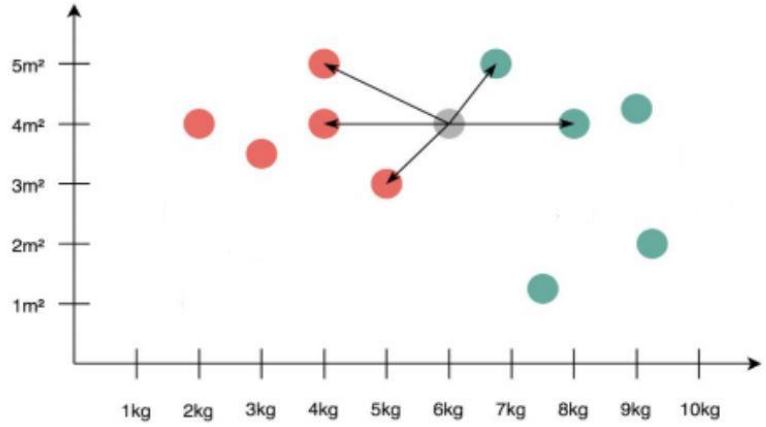
A Figura 5 mostra um exemplo de objetos baseado nas duas características: massa e área de cada um dos objetos, em um espaço euclidiano bidimensional. A cor dos objetos indica o rótulo de cada um: a cor vermelha indica a classe A e a azul a classe B.

Figura 5 - Gráfico dos objetos e suas características



Para classificar um novo objeto ou dado, escolhe-se primeiro o K, que se refere ao número de vizinhos com os quais o novo objeto será comparado. A Figura 6 ilustra um exemplo com K = 5. Após escolhido o melhor K, o algoritmo calcula a distância euclidiana entre o novo objeto (ilustrado na cor cinza na Figura 6) e os seus K objetos vizinhos mais próximos. Por meio da Figura, é possível ver que três objetos são da classe vermelha e dois da azul, portanto, o KNN define o objeto não classificado (cinza) pertence à classe vermelha.

Figura 6 - Gráfico de objetos e suas características em situação K = 5



Para calcular a distância euclidiana, utiliza-se a fórmula demostrada na equação abaixo, em que d_E é a distância euclidiana; x, y os objetos e x_i, y_i os atributos dos objetos.

$$d_E(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{4}$$

A fim de demonstrar o cálculo da distância euclidiana de cada instância em relação ao objeto de interesse (objeto cinza) e, assim, saber quais se encontram mais próximos dele para determinar a sua classe, organizaram-se os dados do gráfico da Figura 6 conforme a Tabela 3.

Tabela 3 - Tabela com os atributos e classe de cada objeto

#	kg	m²	Classe
1	2	4	Vermelho
2	3	3.5	Vermelho
3	4	4	Vermelho
4	4	5	Vermelho
5	5	3	Vermelho
6	6.5	5	Azul
7	7.5	1	Azul
8	8	4	Azul
9	9	4.5	Azul
10	9.5	2	Azul
11	6	4	Desconhecido

A instância 11 (última linha) se refere ao objeto de interesse e, como foi explicado acima, a distância euclidiana será calculada em relação a ele para descobrir a qual classe ele pertence. Para exemplificar, é demonstrada a distância euclidiana da instância 1 à instância 11 (objeto de interesse), utilizando os seus atributos kg e m² de cada objeto, que equivalem, respectivamente, a 2kg, 4m² e 6kg, 4m² para o objeto número 1 e 11. Logo:

$$d_E(x,y) = \sqrt{\left(kg(x) - kg(y)\right)^2 + \left(m^2(x) - m^2(y)\right)^2} \quad (5)$$

$$d_E(1,11) = \sqrt{(2-6)^2 + (4-4)^2}$$

$$d_E(1,11) = \sqrt{(-4)^2 + (0)^2}$$

$$d_E(1,11) = \sqrt{(16)^2 + 0}$$

$$d_E(1,11) = 4$$

Por meio da equação 5, tem-se que a distância do objeto 1 em relação ao objeto 11 é igual a 4. Realizando o mesmo procedimento para o restante dos objetos, tem-se os valores conforme a Tabela 4. E considerando k = 5, ou seja, considerando as 5 instâncias mais próximos, o objeto desconhecido passa a pertencer à classe Vermelha.

Tabela 4 - Resultado da classe da instância 11 usando k = 5

#	kg	m²	Classe	Distância à instância #11
1	2	4	Vermelho	4
2	3	3.5	Vermelho	3.04
3	4	4	Vermelho	2
4	4	5	Vermelho	2.23
5	5	3	Vermelho	1.41
6	6.5	5	Azul	1.11
7	7.5	1	Azul	3.35
8	8	4	Azul	2
9	9	4.5	Azul	3.08
10	9.5	2	Azul	4.03
11	6	4	Vermelho	-

Para otimizar e aumentar a taxa de acerto do KNN, é importante escolher o melhor K. Porém esta não é uma tarefa fácil e não é possível estabelecer um K fixo para todas as amostras de dados, ainda que definido por especialistas (Zhang et al., 2017).

Quando o K é muito pequeno, a classificação se torna sensível a pontos de ruído. Por outro lado, quando é muito grande, a vizinhança poderá incluir dados de outras classes. Além disso, é preferível que o K seja um número ímpar, para diminuir a ocorrência de impossibilidade da classificação devido a empates na quantidade de objetos

de vizinhos próximos (SOUSA, 2013; DUDA; HART; STORK, 2001). Portanto, na maioria das vezes, são feitos diversos testes até encontrar o melhor K para a amostra de dados disponível.

O capítulo a seguir descreve o terceiro algoritmo de classificação, empregado no presente trabalho, conhecido como Árvore de Decisão.

2.2.1.3. Árvore de Decisão C4.5

A Árvore de Decisão C4.5 faz parte da família das árvores de decisão que são estruturas de dados que podem ser representadas como conjuntos de regras if-then. Esses métodos de aprendizado estão entre os algoritmos de inferência indutiva mais populares e foram aplicados com sucesso a uma ampla gama de tarefas, desde o aprendizado até o diagnóstico de casos médicos (MITCHELL, 1997).

Vale ressaltar que as árvores de decisão são as únicas a tratar os resultados de forma hierárquica (com priorização). Dessa maneira, nelas, o atributo mais relevante é destacado na árvore como o primeiro nó (nó raiz), e os atributos menos importantes são mostrados nos nós ou nodos subsequentes. Ao representar os atributos em ordem de importância, a Árvore permite compreender e analisar quais fatores mais influenciam no trabalho do usuário (Crepaldi et al., 2011).

A fim de determinar o grau de importância de cada atributo, C4.5 utiliza a Entropia que, baseada na Teoria da Informação, é uma grandeza que mede a desordem, tanto de objetos físicos quanto de informações. Dessa maneira, quanto maior o grau da entropia maior é a desordem, logo, quanto menor o grau da entropia melhor a organização. Seguindo este raciocínio, o melhor grau de entropia é aquele que possui valor mais baixo (Arariboia 1989; Monard et al. 1997). Para calcular a entropia de um determinado atributo da base de dados, utiliza-se a equação 6:

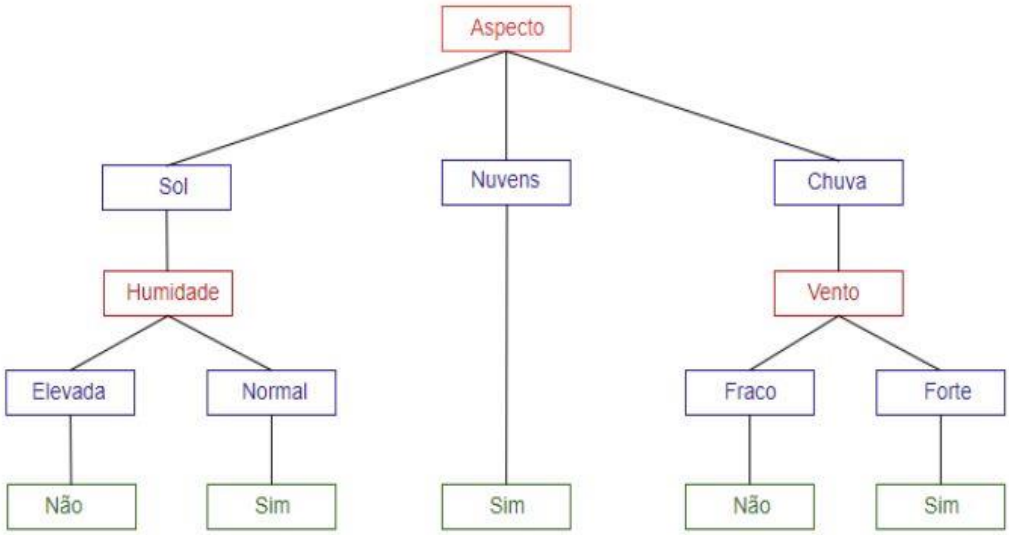
$$Entropia(S) = -(p^{+} * log_2 p^{+}) - (p^{-} * log_2 p^{-}) \quad (6)$$

Na equação da Entropia tem-se p^{+} que se refere à proporção de instâncias (ocorrência) positivas; p^{-} corresponde às proporções de instâncias negativas e S corresponde à soma de todas as instâncias positivas e negativas. Já o $log_2 p^{+}$ é o logaritmo na base de 2 pela quantidade de instâncias positivas e $log_2 p^{-}$ é o logaritmo na

base de 2 pela quantidade de instâncias negativas.

Retomando o exemplo da classificação dos dias adequados para jogar tênis, ao realizar o cálculo da entropia de todas as características (atributos), uma estrutura de árvore é formada, como pode ser visto na Figura 7.

Figura 7 - Exemplo do jogo de Tênis em um modelo de árvore de decisão



Conforme a Figura 7, é possível verificar que a característica de menor Entropia é colocada no topo da árvore, neste caso, o atributo Aspecto. Depois disso, o restante dos atributos são ramificados, conforme seu grau de importância, nos nós e a última folha se refere à classe de cada instância. Por exemplo, a instância:

(Aspecto = Sol, Temperatura = Quente, Umidade = Alta, Vento = Forte)

seria classificada como uma instância negativa, ou seja, a árvore prevê que Jogar Tênis = não (MITCHELL, 1997).

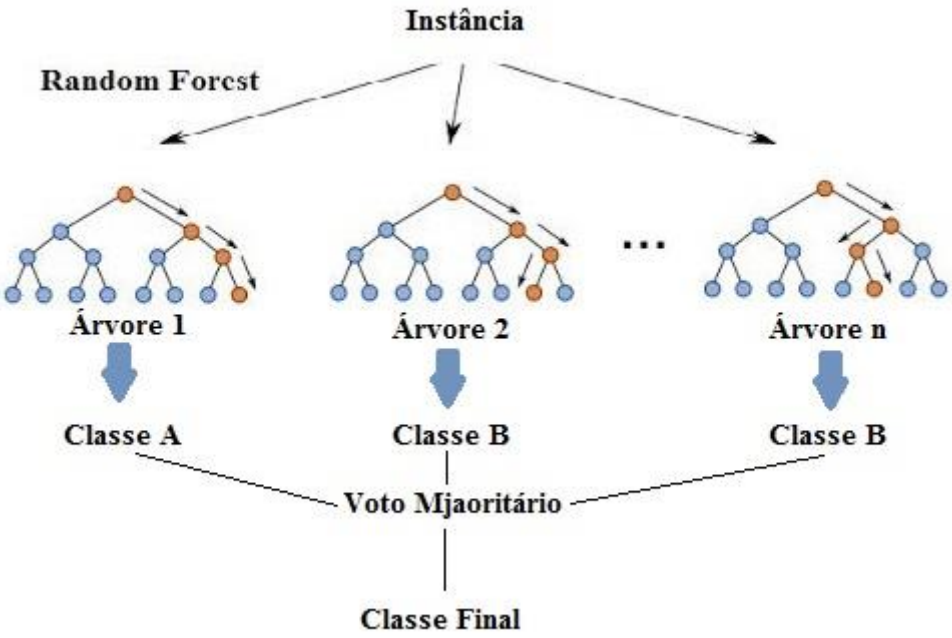
A seguir, no capítulo seguinte, é apresentado o algoritmo Random Forest e suas características.

2.2.1.4.Random Forest

Dentro do grupo de aprendizado supervisionado, há uma subcategoria conhecida como aprendizado em Ensemble. Trata-se de modelos que combinam múltiplos algoritmos simples de aprendizado de máquina. Usando este conceito, é possível obter um desempenho similar, ou até mesmo superior, ao desempenho de algoritmos de classificação mais complexos (KÜMMEL, 2017).

O algoritmo Random Forest (Floresta Aleatória) é um método de aprendizado em Ensemble e utiliza múltiplas Árvores de Decisão durante o processo de treinamento. Como as árvores de decisão usadas no Random Forest possuem todas o mesmo peso, este classificador retorna a moda dos resultados das árvores integradas para a classificação, ou seja, a saída da classificação é dada pela maioria dos votos dados pelas árvores (Friedman et al., 2001). A Figura 8 mostra este procedimento.

Figura 8 - Ilustração do funcionamento do algoritmo Random Forest



Segundo a Figura 8, cada árvore de decisão passa pelo processo de classificação, ou seja, treino e teste. Após todas retornarem a classe de determinada instância, a classe final é decidida pelo voto majoritário, ou seja, a classe que se repetir mais. E é assim para todas as instâncias.

Uma das maiores vantagens da floresta aleatória é o poder de lidar com dados em

grandes volumes e com muitas dimensões. Este algoritmo é capaz de lidar com milhares de variáveis de entrada e identificar as variáveis ou características mais importantes usando seus próprios métodos, sendo por isso considerado um dos métodos de redução de dimensões e instâncias necessárias para o aprendizado do modelo de classificação. O capítulo 2.2.4 traz mais informações e conceitos importantes sobre a redução da dimensionalidade e sua importância no aprendizado da máquina.

Por outro lado, o modelo de Random Forest construído ocupa uma grande quantidade de memória, dependendo da quantidade de árvores usadas no processo. Além disso, um modelo treinado funciona um pouco mais lento do que outros algoritmos. Por exemplo, se houver 100 árvores em um modelo, é necessário iterar sobre todos eles para obter o resultado.

O próximo capítulo aborda o segundo tipo de aprendizado de máquina: o aprendizado não supervisionado.

2.2.2. Aprendizado Não Supervisionado

Como foi citado, anteriormente, o aprendizado não supervisionado, também conhecido como aprendizado por observação e descoberta, supõe que o conjunto de exemplos (dados) não está rotulado. Portanto, este modo de aprendizado permite abordar problemas com pouca ou nenhuma ideia do resultado, fornecendo variáveis sem informar a classe correspondente. Desta forma, os algoritmos de aprendizado não supervisionado são responsáveis para encontrar estruturas significativas e interessantes nos dados, como padrões, perfis, itens semelhantes e relações importantes em uma amostra de exemplos (Decker & Focardi,1995;McCallum, Nigam, & Ungar,2000).

Para tal, geralmente, é utilizada a técnica de *clustering* (agrupamento), que consiste em, literalmente, agrupar uma coleção de dados, em *clusters* (grupos), segundo algum padrão de similaridade que o próprio algoritmo consegue identificar (Everitt, 1993). As Figuras 9 e 10 ilustram, de forma simples, este conceito.

Figura 9 - Objetos com classe desconhecida

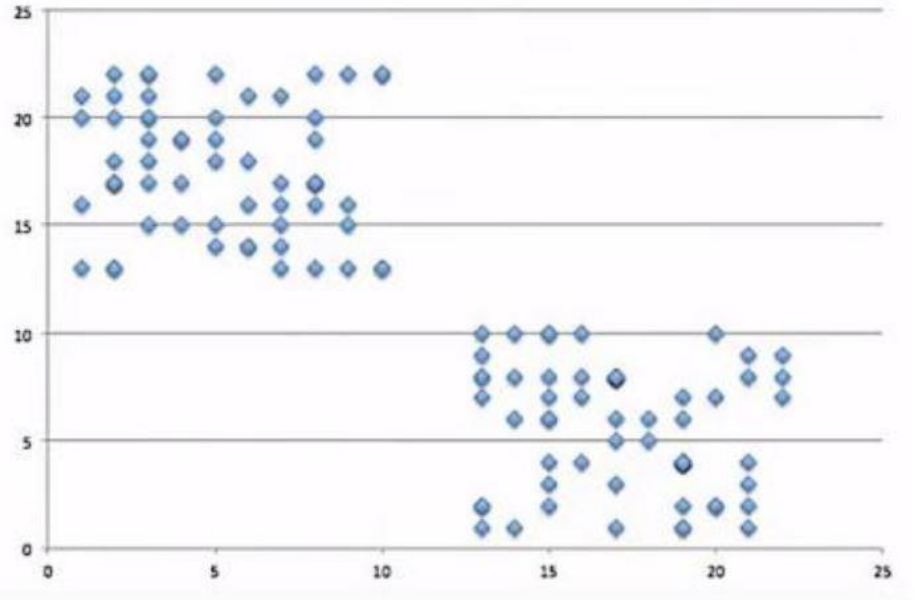
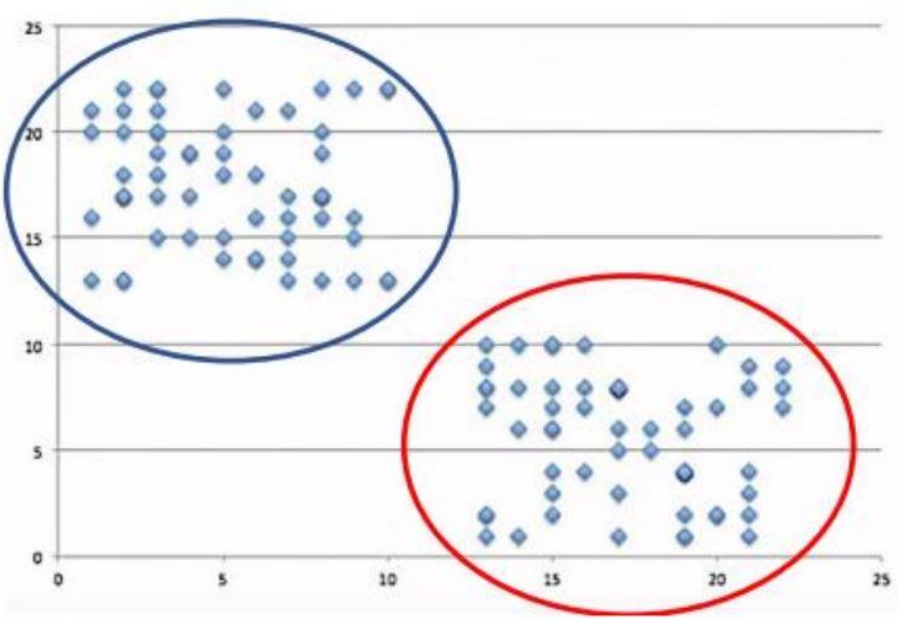


Figura 10 - Objetos após Clustering



A Figura 9 mostra dados quaisquer sem rótulo nenhum e após o clustering, conforme a Figura 10, uma vez que os dados são agrupados segundo a sua similaridade, o usuário, por observação e análise consegue descobrir o que cada conjunto representa para os objetos reais.

O próximo capítulo aborda outro algoritmo de aprendizado de máquina do tipo não-supervisionado chamado K-means.

2.2.2.1. K-Means

Um algoritmo de aprendizado não supervisionado bastante conhecido, para aprendizado de máquina, é o K-Means (também chamado de K-Médias) que é um método de *Clustering* e tem como ideia principal fornecer uma classificação de informações de acordo com os próprios dados (Malheiros, 2017).

Ainda, segundo Malheiros (2017), primeiramente, o usuário fornece ao algoritmo a quantidade de clusters que ele deseja. Este número é conhecido como k e por isso o nome do algoritmo: K-Means. Uma vez escolhido k, o algoritmo vai analisar todos os dados e criar k agrupamentos. Para tanto, este classificador compara a distância entre cada valor de cada instância, utilizando, geralmente, a distância euclidiana. Os pontos dos centróides são atribuídos para os centros mais próximos.

Depois o K-Means começa a iteração, e a cada rodada, o valor dos centróides é atualizado pela média dos valores de cada atributo de cada instância que pertence a este centróide, já que o valor do centro pode mudar dependendo da medida de proximidade dos dados. O algoritmo termina quando os valores dos centróides ficam estáveis, ou seja, eles param de variar. Sendo assim, o algoritmo gera k clusters bem definidos. As Figuras abaixo ilustram o antes e depois da classificação usando K-Means (Malheiros, 2017)

Figura 11 - Dados não rotulados com três centroides definidos

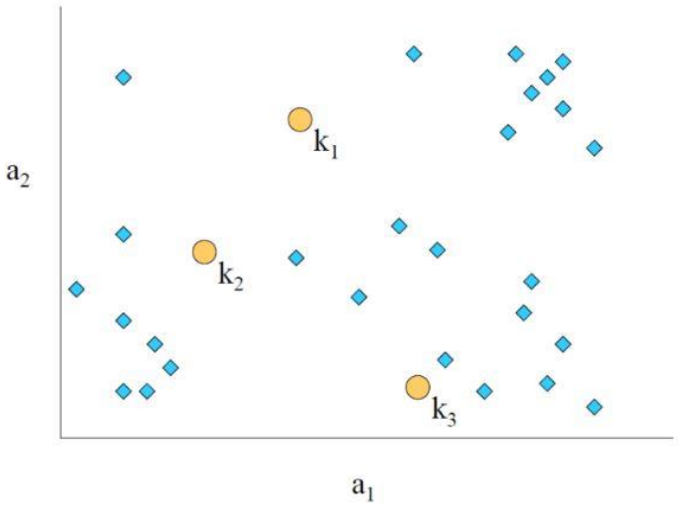
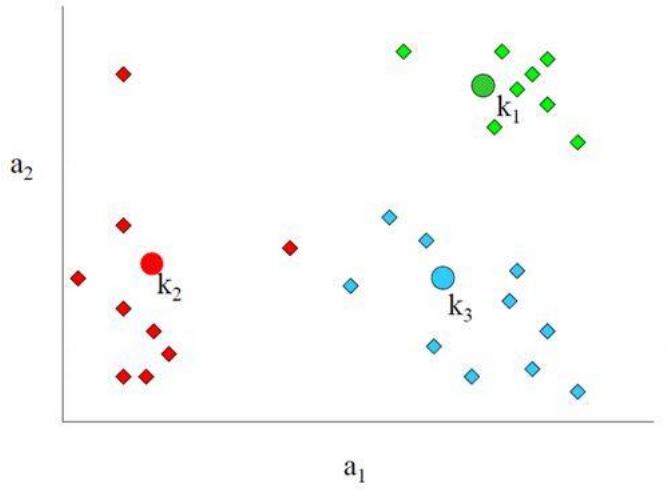


Figura 12 - Três clusters com seus centroides finais



De maneira simplificada, o K-Means funciona como descrito nos 4 passos a seguir:

1) O usuário define o número de k de centróides;
2) K-means escolhe aleatoriamente um objeto para ser o centroide inicial;
3) K-Means define o centróide mais próximo para cada instância e define seu respectivo cluster;
4) K-Means atualiza os valores dos centróides pela média de cada cluster;
5) Os passos 2 e 3 são repetidos até os valores dos centróides pararem de mudar.

É possível que os cluster entrem em um loop infinito e não consigam convergir nunca. Neste caso, o usuário é obrigado a decidir o momento da parada e escolher os melhores resultados.

A seguir, é feita uma introdução aos métodos de avaliação e testes dos classificadores usados neste trabalho.

2.2.3. Avaliação e Testes de Classificadores

Para avaliar os algoritmos classificadores se faz necessário o uso de métodos de avaliação para evitar o teste de um modelo no mesmo conjunto de dados no qual ele foi treinado. A seguir são apresentados dois métodos de avaliação dos algoritmos empregados na presente pesquisa.

2.2.3.1.Holdout

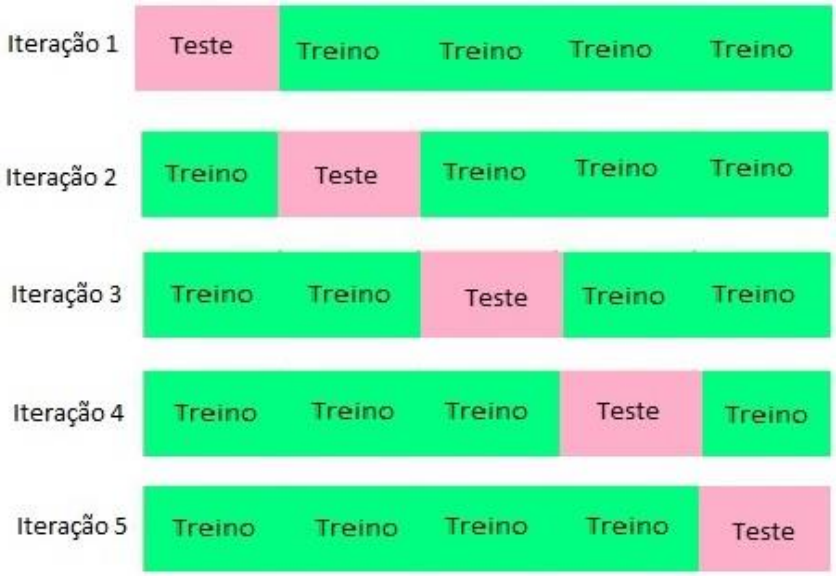
A técnica de teste Holdout, consiste em dividir a amostra de dados disponível em duas porcentagens: uma parte do conjunto de treinamento e outra para os testes. Nessa técnica, os dados de treinamento são usados para construção do modelo (treino), e os de teste são, exclusivamente, para medir a taxa de erro do classificador. Esta divisão, geralmente, é feita em proporções como 80%/20%, 75%/25%, 60%/40% e etc.

2.2.3.2. Validação Cruzada (*Cross Validation*)

A técnica de validação cruzada, também conhecido como k-fold cross-validation, em que k representa o número de folds (grupos) gerados a partir da amostra de exemplos, serve para treinar e testar o modelo. A cada iteração, um fold diferente é usado para testar o sistema de aprendizado e todos os outros restantes são usadas para treinar o sistema (Batista, 2003). Por exemplo, no caso de 5 folds, todo o conjunto de exemplos é dividido em 5 partes diferentes, porém de tamanhos iguais. Dessa forma, são realizadas 5 iterações, e a cada iteração, uma partição é usada para testar e as outras 4 restantes para treinar o modelo. A Figura 13 representa o exemplo com k-folds = 5.

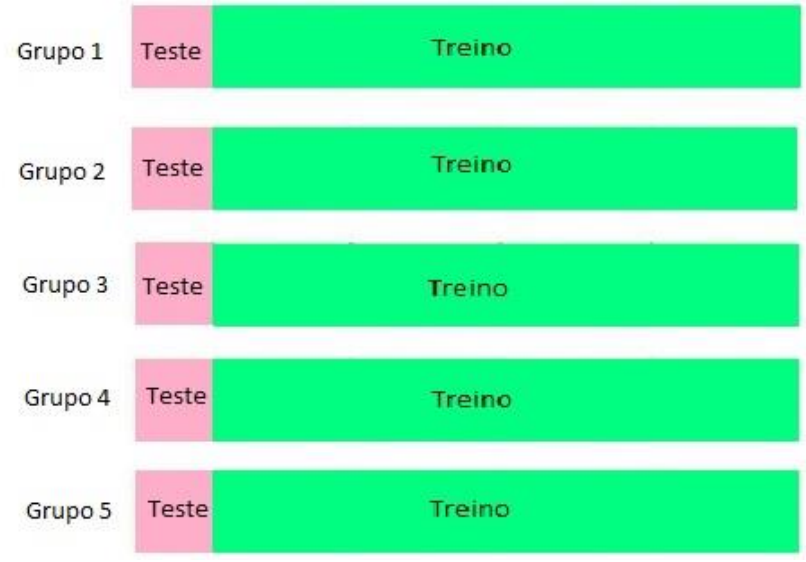
A técnica CV é de suma importância, visto que ela evita o *Overfitting*. Ou seja, quando ocorre o vício dos algoritmos classificadores, durante o teste, ao usar os mesmos dados com os quais foi treinado. Isto, portanto, resulta em uma taxa de desempenho enganosa, que não condiz com a realidade.

Figura 13 - Iterações Cross-Validation para k-folds = 5



O ShuffleSplit é um tipo de Validador cruzado de permutação aleatória. Assim como na validação cruzada comum, explicada anteriormente, ele divide o conjunto de dados em número escolhido de folds. Os dados de cada iteração são divididos na porcentagem escolhida de conjunto de treino e teste. É importante destacar que os dados de cada iteração são diferentes uma da outra. Por padrão, o conjunto de teste usado no ShuffleSplit, quando não especificado, é de 10% e o restante dos 90% de dados são usados para treino. A Figura 14 ilustra um exemplo de ShuffleSplit usando 5 folds e 10% de dados para teste.

Figura 14 - Método ShuffleSplit usando 5 folds e 10% de cada grupo para teste.



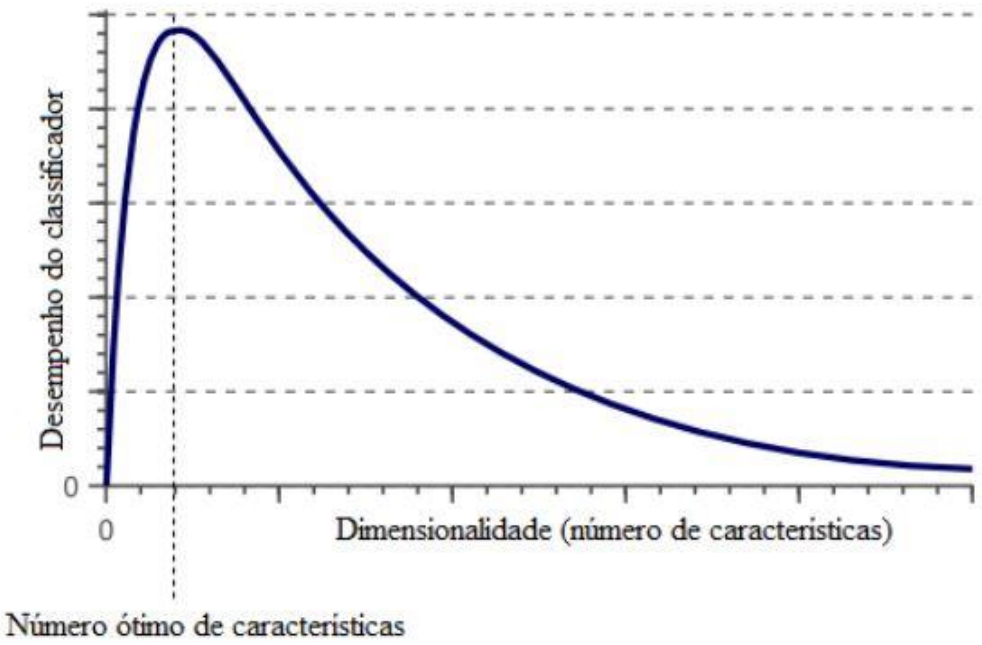
O capítulo seguinte apresenta o conceito da maldição de dimensionalidade, suas características e importância para o aprendizado dos modelos computacionais de classificação.

2.2.4. Maldição de Dimensionalidade

O conceito de Maldição da Dimensionalidade declara que ao aumentar, sem critério, o número de características (atributos) para um classificador, sem aumentar o número de exemplares de treinamento, o desempenho do classificador aumenta apenas nos dados de treinamento. Portanto, ele se torna pior, de maneira geral, em amostras de dados novos (KÜMMEL, 2017).

Conforme ilustra o gráfico da Figura 15, ao aumentar sem critério o número de características, o desempenho do classificador tende a diminuir.

Figura 15 - Queda de desempenho do classificador com o aumento de características



Por isso, é importante equilibrar o número de características empregadas para criar um modelo de classificação. Sendo que o ideal é usar o menor número possível de

características a fim de alcançar a precisão mais adequada nos dados de teste (KÜMMEL, 2017).

Reduzir a dimensionalidade da amostra de dados possui algumas vantagens como a redução do tempo e custo de treinamento do modelo de classificação, além de facilitar a compreensão e análise do modelo gerado (JAMES, 2013). Uma das estratégias para reduzir o número de características de uma coleção de dados é a seleção de características, descrita no seguinte capítulo.

2.2.4.1. Seleção de características

A seleção de características é uma estratégia de redução de dimensionalidade que busca selecionar um subconjunto das características mais relevantes para a construção de um modelo de classificação (DASH, 1997), visto que este conceito considera que alguns dos atributos podem ser redundantes ou irrelevantes. Portanto, estas podem ser removidas sem causar prejuízo à acurácia do modelo de classificação.

Assim, existem três categorias de algoritmos de seleção de características: *wrappers*, métodos de filtragem e métodos integrados ou embedded (GUYON et al., 2003).

A categoria de *wrappers* são métodos de busca nos quais diferentes subconjuntos de características são usados para treinar um classificador. Após isso, algumas inferências são extraídas dos resultados e depois o algoritmo decide por adicionar ou retirar uma ou mais características de um subconjunto de características até descobrir o subconjunto com o melhor desempenho. Os métodos wrappers são considerados algoritmos de busca e possuem como desvantagem um alto custo computacional.

Já os métodos de filtragem, diferentemente dos métodos *wrappers*, ocorrem como um pré-processamento antes do treinamento e de forma independente do classificador. A seleção de características é feita baseada em pontuações originadas de testes estatísticos aplicados à amostra de dados, medindo a correlação entre cada característica e a classe resposta conhecida dos dados de treinamento.

Normalmente, os métodos de filtragem resultam em subconjuntos de características com baixo desempenho, quando comparados com os subconjuntos

resultantes por meio dos métodos *wrappers*. Porém, a coleção de características gerada por meio de métodos de filtragem não depende de um modelo de predição específico e, por isso, torna mais evidentes as relações entre características.

Por último, os métodos integrados ou embedded realizam a combinação dos dois métodos anteriores: métodos de filtragem e métodos wrappers. Esses métodos integrados estão implementados dentro de alguns algoritmos de aprendizado de máquina que possuem um mecanismo próprio de seleção de características durante a construção do modelo de classificação com os dados de treinamento. Um exemplo de algoritmo de aprendizado de máquina conhecido por possuir métodos de seleção de características integrados é o Random Forest.

Basicamente, Random Forest usa o método *feature_importances* para calcular a importância da característica usando o grau de impureza do nó em cada uma das árvores de decisão empregadas, ou seja, utiliza a entropia que foi explicada anteriormente no capítulo 2.2.1.3. De maneira geral, a entropia mede o grau de desordem das informações ou objetos. Desta maneira, quanto menos desordem ou maior o grau de pureza, menor o grau de entropia e quanto maior a desordem, maior o grau de entropia.

Portanto, o melhor grau de entropia é aquele cujo valor for o mais baixo, por isso o nó com entropia mais baixa (maior grau de importância) fica sempre no topo de cada árvore do Random Forest. Como o algoritmo de Random Forest utiliza diversas árvores de decisão, a importância da característica final é a média de toda a importância da característica das árvores de decisão empregadas.

O capítulo seguinte aborda todas as ferramentas usadas no presente trabalho, como as linguagens de programação, bibliotecas e algumas plataformas.

2.3. Ferramentas utilizadas

Neste capítulo são apresentadas todas as ferramentas utilizadas durante este trabalho, como as linguagens de programação, bibliotecas e plataformas diversas.

2.3.1. Linguagens de Programação

Existem algumas linguagens de programação que possuem estruturas e bibliotecas

que facilitam a implementação de algoritmos de Inteligência Artificial (IA). Além disso, elas possuem características indispensáveis como bom desempenho em tempo de execução, bom suporte a ferramentas e uma grande e ativa comunidade de programadores para o apoio. As linguagens utilizadas para essa finalidade, neste trabalho, são descritas a seguir.

2.3.1.1. Python

Python é uma linguagem de programação mantida pela organização Python Software Foundation (PSF) sem fins lucrativos. Python, é aplicada para o desenvolvimento de sites e desktops, no ensino, aplicações com números e principalmente aplicações científicas (PYTHON, 2019).

A linguagem Python foi criada por Guido Van Rossum, em 1991. Ela é considerada uma linguagem de programação de alto nível e possui facilidade de leitura de código com uma sintaxe clara, simples e direta, porém com grande poder. A sua principal característica é seu uso de identação como delimitador de blocos, sendo ela a única entre as linguagens de programação com esta peculiaridade (Ribeiro, 2011).

A linguagem Python proporciona facilidade de programação, versatilidade e eficiência. Somando-se essas características, Python se mostra promissor para a computação científica.

É importante notar que o Python também é usado em diversas áreas fora do meio acadêmico. É usado como uma linguagem de propósito geral em aplicações como: administração de sistemas, sites da internet, sistemas distribuídos, engenharia de software, interfaces gráficas, motores de busca (Google), rede de computadores, educação, e até aplicações empresariais de larga escala (Ribeiro, 2011).

Esta linguagem foi usada durante todo o processo de classificação com os algoritmos de aprendizado de máquina por possuir diversos recursos e bibliotecas poderosos para esta área.

Outra linguagem de programação presente nesta pesquisa é apresentada no capítulo a seguir.

2.3.1.2. JavaScript

A linguagem de programação JavaScript foi criada em 1995 por Brandan Eich, um especialista em sistemas para computadores da Netscape (Ingalls, 2008). Também conhecida apenas como JS, JavaScript é uma linguagem de programação usada para desenvolver aplicações, sistemas e serviços de alta complexidade.

Com ela, é possível criar páginas web dinâmicas, animações, mapas interativos, aplicativos para dispositivos móveis, além de jogos. O sucesso do JavaScript é evidente e como apontam dados, JS é usado por 97 dos 100 sites mais populares da web (Ingalls, 2008).

A linguagem JS foi empregada no desenvolvimento do aplicativo para classificar os cavalos pantaneiros, usando a plataforma Ionic descrita no capítulo 2.3.3. A próxima linguagem apresentada é o HTML e CSS.

2.3.1.3. HTML e CSS

HTML é a abreviação em inglês para *Hyper Text Markup Language*, que em português, significa linguagem para marcação de hipertexto.

O HTML foi inventado por Tim Berners-Lee, e ao longo dos anos, esta linguagem de web evoluiu por sete versões. Atualmente, a última versão é o HTML5. De maneira simplificada, hipertexto pode ser considerado como todo o conteúdo inserido em um documento para a web e que tem como principal característica a possibilidade de se conectar a outros documentos da web. A construção de hipertextos é feita por meio de links, presentes nas páginas dos sites de internet (Silva, 2008).

Para a apresentação e o visual de uma página web é importante empregar as folhas de estilo em cascata que é a tradução para o termo em inglês *Cascading Style Sheet (*CSS). O Consórcio World Wide Web (W3C), que é o responsável por desenvolver padrões para a Web, define CSS como uma maneira simples para empregar estilos como fontes, cores e espaçamentos aos documentos web (W3C, 2019).

Assim como o JS, HTML e CSS foram usadas durante a implementação do aplicativo com a plataforma Ionic. O próximo capítulo aborda as bibliotecas que foram

usadas no presente trabalho.

2.3.2. Bibliotecas

Assim como as Linguagens de Programação, as bibliotecas também são um recurso importante para a implementação e desenvolvimento de sistemas voltados à área de Inteligência Artificial (IA), pois facilitam a criação desses sistemas.

Uma Biblioteca é nada mais que um conjunto de funções pré-escritas por outros programadores que já resolvem determinados problemas sem que haja a necessidade de uma nova implementação para cada aplicação. As bibliotecas usadas neste trabalho são descritas a seguir.

2.3.2.1.Pandas

Pandas é uma biblioteca open source (código aberto), que oferece alto desempenho, estruturas de dados fáceis de usar e ferramentas de análise de dados para a linguagem de programação Python. (PANDAS, 2019).

Basicamente esta biblioteca foi usada para a manipulação dos dados e sua análise, usando os próprios métodos oferecidos pelo Pandas para agrupar, filtrar e combinar dados.

2.3.2.2. Scikit-Learn

Scikit-Learn é uma biblioteca de código aberto para Python com ferramentas simples e eficientes para análise preditiva de dados, acessível a todos e reutilizável em vários contextos como mineração de dados, reconhecimento de imagem, resposta a medicamentos, preços de ações, agrupamento de resultados de experimentos entre vários outros (SCIKIT-LEARN, 2019).

Esta biblioteca foi usada para importar os algoritmos de classificação KNN, K-Means, C4.5 e Naive Bayes, assim como os inúmeros métodos prontos de aprendizado de máquina. Além disso, foi desta biblioteca que foram importados os métodos de

avaliação dos classificadores como o ShuffleSplit e o Holdout.

O seguinte capítulo traz a plataforma usada para desenvolvimento do aplicativo.

2.3.3. Ionic Framework

O Ionic Framework é um kit de ferramentas de interface do usuário de código aberto para a criação de aplicativos móveis e de desktop de alta qualidade e desempenho, usando HTML, CSS e JavaScript. Por meio desta plataforma, é possível criar e implantar aplicativos que funcionam em várias plataformas, como iOS nativo, Android, desktop (Ionic Framework, 2019).

Esta plataforma foi usada para o desenvolvimento do aplicativo que realiza a classificação porque há várias bibliotecas e métodos de aprendizado de máquina para a linguagem Javascript e não há plataformas confiáveis para desenvolvimento de aplicativos para a linguagem Python. E por outro lado utilizou-se a linguagem Python pela vasta gama de ferramentas e bibliotecas de aprendizado de máquina que ela disponibiliza. Portanto, a pesquisa inteira foi feita em Python e o aplicativo em Javascript usando Ionic.

2.3.4. Anaconda Navigator

O Anaconda Navigator é uma interface gráfica do usuário da área de trabalho que permite iniciar aplicativos e gerenciar facilmente pacotes, ambientes e canais de anaconda sem usar linha de comando. Esta plataforma está disponível para Windows, macOS e Linux. Além disso, o navegador é uma maneira fácil de apontar e clicar para trabalhar com pacotes e ambientes sem a necessidade de digitar comandos conda em uma janela do terminal. Ademais, ao usá-lo é possível encontrar os pacotes desejados, instalá-los em um ambiente, executar os pacotes e atualizá-los dentro do próprio navegador. Uma das aplicações web disponíveis no Anaconda Navigator é o Jupyter Notebook (Anaconda, 2019).

O Jupyter Notebook é um aplicativo da web, também de código aberto, que permite criar e compartilhar documentos que contêm código ativo, equações,

visualizações e texto narrativo. As principais funções desta aplicação são: limpeza de dados e transformação, simulação numérica, modelagem estatística, visualização de dados, aprendizado de máquina, entre outras (Jupyter, 2019).

Desta forma, o Jupyter Notebook foi utilizado na execução dos códigos de programação da linguagem Python. Ao utilizar o Jupyter, é possível visualizar a saída (*output)* imediata do código, sem que haja a necessidade de compilar ou executar todo o documento.

O próximo capítulo é sobre a metodologia empregada no presente trabalho e todos os materiais e procedimentos adotados.

3. Materiais e Métodos

Neste capítulo, são apresentados o banco de dados utilizado, o método realizado para a seleção de características e o procedimento para a escolha do classificador mais eficiente para os dados disponíveis. A pesquisa deste trabalho está representada, de forma simplificada, no fluxograma da Figura 16.

Figura 16 - Fluxo do trabalho completo

O primeiro passo do fluxograma consiste no tratamento da base, ou seja, filtro dos dados válidos, remoção de instâncias com dados faltantes, conversão dos mesmos em dados válidos usando o Pandas. No segundo passo, foram rotulados os dados dos cavalos com a sua respectiva classe usando métodos descritos no capítulo seguinte.

A terceira etapa se trata da seleção das características mais importantes para a classificação. Ou seja, em vez de usar todas as 15 medidas para avaliação do cavalo pantaneiro, foram usados métodos de aprendizado de máquina para diminuição destas medidas. Depois disso, na quarta etapa, foram treinados e testados os algoritmos KNN, Naive Bayes e C4.5. Após isso, no quinto passo, foi escolhido o algoritmo mais adequado e com maior taxa de acerto para a classificação dos cavalos. No sexto passo, foi

implementado o aplicativo e no sétimo passo foi implementado o algoritmo escolhido para a classificação na linguagem Javascript para ser combinado com o aplicativo e, por último, o aplicativo foi testado em campo.

O capítulo seguinte traz todas as informações sobre a base de dados e descreve os passos 1 e 2 do fluxograma.

3.1.Conjunto de dados utilizados

O conjunto de dados utilizado nesta pesquisa foi disponibilizado pela EMBRAPA Pantanal em forma de três planilhas de arquivo .xlsx, manipulável no aplicativo Excel da Microsoft. A primeira planilha é composta por 3522 cavalos pantaneiros cadastrados na ABCCP e cavalos premiados nos últimos 10 anos. Portanto, estes configuram os cavalos de alta e média qualidade. O segundo arquivo contém 116 cavalos que não foram aprovados pela ABCCP e são de baixa qualidade. A terceira planilha é composta por 3 cavalos pantaneiros premiados no ano de 2018, os quais foram usados como referência (alta qualidade).

Nos três arquivos, são retratadas as 15 medidas lineares que já foram citadas no capítulo 2.3.1. Além destas, encontram-se, na base, as seguintes características: sexo, pelagem e as seguintes relações: largura do peito/espádua; altura do dorso/garupa; altura da cernelha/garupa; costados/vazio subesternal; altura da cernelha/corpo; comprimento da cabeça/pescoço e por último a relação das ancas/dorso.

Posteriormente, estes dados foram armazenados em arquivos .csv ou comma-separated values que, como o próprio nome indica, possui os seus valores separados por vírgulas (mas também é possível usar ponto e vírgula como separador). A conversão foi necessária para que os dados sejam lidos corretamente pela linguagem de programação e para que sejam manipulados na biblioteca Pandas.

O arquivo .csv é manipulável tanto no aplicativo Excel da Microsoft, quanto por linguagens de programação. A Figura 17 representa uma parte da estrutura de dados no formato .csv e a Figura 18 ilustra a mesma base de dados na extensão .xlsx.

Figura 17 - Base de dados em arquivo .csv

```
1  Sexo;Pelagem;Cabeca;Pescoco;Corpo;Dorso;Peito;Costados;
2  Macho;Tord, Negro;0,53;0,6;1,37;1,37;0,27;0,57;1,38;0,4
3  Macho;Baia;0,56;0,62;1,38;1,36;0,27;0,59;1,4;0,48;0,17;
4  Macho;Tordilha;0,49;0,68;1,38;1,33;0,28;0,69;1,37;0,46;
5  Macho;Baia;0,5;0,52;1,39;1,36;0,29;0,59;1,38;0,45;0,17;
6  Macho;Tordilha;0,54;0,51;1,38;1,35;0,31;0,56;1,38;0,44;
7  Macho;Tordilha;0,48;0,54;1,35;1,29;0,3;0,54;1,35;0,43;0
8  Macho;Tordilha;0,5;0,55;1,31;1,34;0,27;0,57;1,37;0,45;0
```

Figura 18 - Planilha da base de dados em arquivo .xlsx

Sexo	Pelagem	Cabeca	Pescoco	Corpo	Dorso	Peito	Costados
Macho	Tord, Negro	0,53	0,6	1,37	1,37	0,27	0,57
Macho	Baia	0,56	0,62	1,38	1,36	0,27	0,59
Macho	Tordilha	0,49	0,68	1,38	1,33	0,28	0,69
Macho	Baia	0,5	0,52	1,39	1,36	0,29	0,59
Macho	Tordilha	0,54	0,51	1,38	1,35	0,31	0,56
Macho	Tordilha	0,48	0,54	1,35	1,29	0,3	0,54

Os três datasets passaram por um processo de filtragem para escolha do conjunto adequado de treinamento e do conjunto de teste. Dessa forma, foram removidos dados com valores nulos e valores textuais foram transformados todos em valores numéricos. Este processo também é responsável por remover dados muito diferentes do resto da amostra de dados, também conhecidos como *outliers*, já que estes podem prejudicar o desempenho de um modelo de classificação.

Em seguida ao processo de filtragem, fez-se necessária a adição da coluna classe (rótulo) na base de dados. Escolheu-se, arbitrariamente, 0,1 e 2 para baixa, média e alta qualidade respectivamente. Dessa forma, para o dataset dos cavalos não aprovados na ABCCP que são bem separados e definidos, foi apenas necessário adicionar uma coluna no final da estrutura de dados denominada classe com os valores igual a 0 que corresponde à qualidade baixa.

Já para a base de dados, em que se encontram, misturados, dados de cavalos de alta e média qualidade, fez-se necessário, primeiramente, descobrir o rótulo de cada instância. Para tanto, usou-se o algoritmo K-means.

Sabe-se que a classe do dataset dos três cavalos premiados é de alta qualidade, ou

seja, com valores iguais a 2. Estes dados, com o rótulo conhecido, foram adicionados ao dataset dos cavalos com os de alta e média qualidade. Estes dados foram usados no modelo do K-means, escolhendo dois clusters. No final, obteve-se dois grupos distintos de dados, porém sem rótulos ainda. Este algoritmo foi utilizado por possuir a capacidade de lidar com dados sem rótulo e agrupá-los conforme sua similaridade, analisando seu padrão.

A fim de saber o rótulo de cada grupo, realizou-se a busca pelas instâncias premiadas (alta qualidade) nos dois grupos gerados. Desta forma, o grupo em que foram encontrados os cavalos premiados foram todos rotulados como de alta qualidade, ou seja, classe com valores igual a 2. E o outro grupo de dados ganhou uma coluna classe com os valores igual a 1, ou seja, pertencentes à classe de média qualidade.

Uma vez com todos os dados rotulados, foi necessário balancear todos os dados, visto que há quinze vezes mais dados da classe média e alta do que de baixa qualidade, fato que pode prejudicar o aprendizado do modelo. No final, todos os datasets foram armazenados em um único arquivo .csv.

O próximo capítulo descreve os procedimentos adotados de maneira geral durante a pesquisa.

3.2. Procedimentos

Com a base de dados pronta, fez-se necessário realizar a seleção de características a fim de selecionar as informações mais importantes para a classificação. Para tanto, usou-se o próprio método do classificador Random Forest, o *feature_importances*, descrito anteriormente no capítulo 2.2.4.1, que consegue calcular o grau de importância de cada atributo baseado na teoria da informação. Portanto, este método retorna o quão informativo cada característica é. Os valores retornados variam entre 0 e 1, sendo que quanto maior o número é, mais informativo o atributo se torna.

Após a obtenção do grau de importância de cada atributo, foram formados diferentes conjuntos de características de acordo com o grau de importância de cada. Em seguida, os conjuntos passaram por um processo de treinamento e teste, usando os três algoritmos escolhidos (KNN, árvore de decisão (C4.5) e o Naive Bayes (NB), a fim de

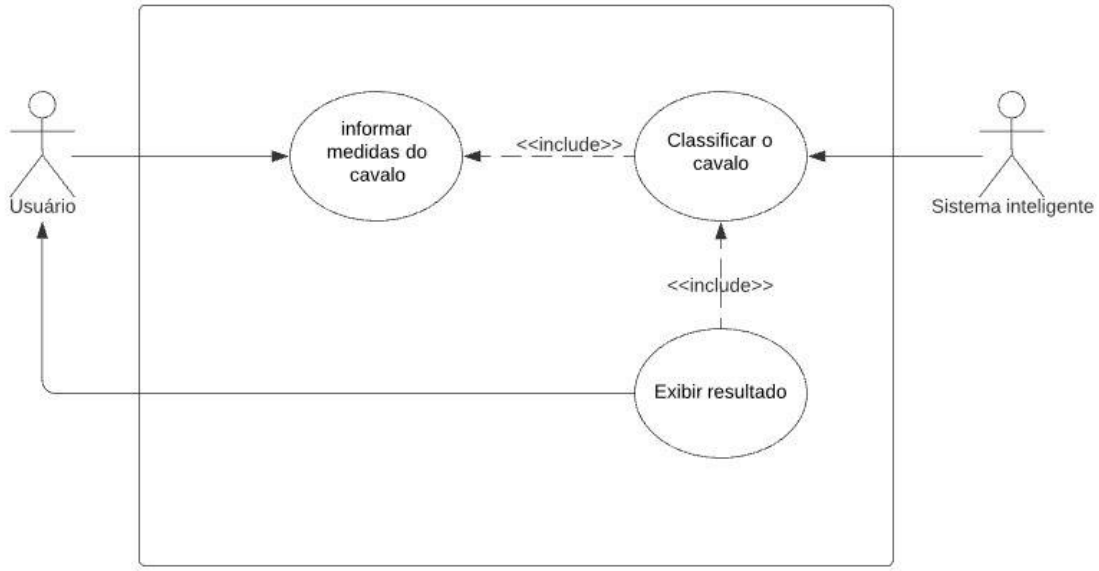
selecionar o conjunto mais apropriado e de maior taxa de acerto.

Para o aprendizado dos classificadores, estes foram treinados e passados por testes que utilizam a técnica de ShuffleSplit. Para tanto, escolheram-se os números de 2 a 10 folds (grupos) de forma aleatória para fins de testes.

A fim de encontrar o melhor número de vizinhos para o algoritmo KNN, foram testados números de 1 a 10, de forma aleatória também, em todos os conjuntos formados para todos os folds escolhidos.

Ao final foi desenvolvido um aplicativo capaz de classificar um novo dado de um cavalo com rótulo desconhecido. O caso de uso da Figura 19 apresenta o funcionamento básico da aplicação. Primeiro, o usuário informa as medidas do cavalo que foram selecionadas, e então, o modelo classificador escolhido realiza a classificação do novo dado e, por fim, retorna a classe à qual o cavalo pertence, que pode ser de alta, média ou baixa qualidade.

Figura 19 - Caso de uso do aplicativo



O próximo capitulo traz todos os experimentos e resultados obtidos nesta pesquisa.

4. Experimentos e Resultados

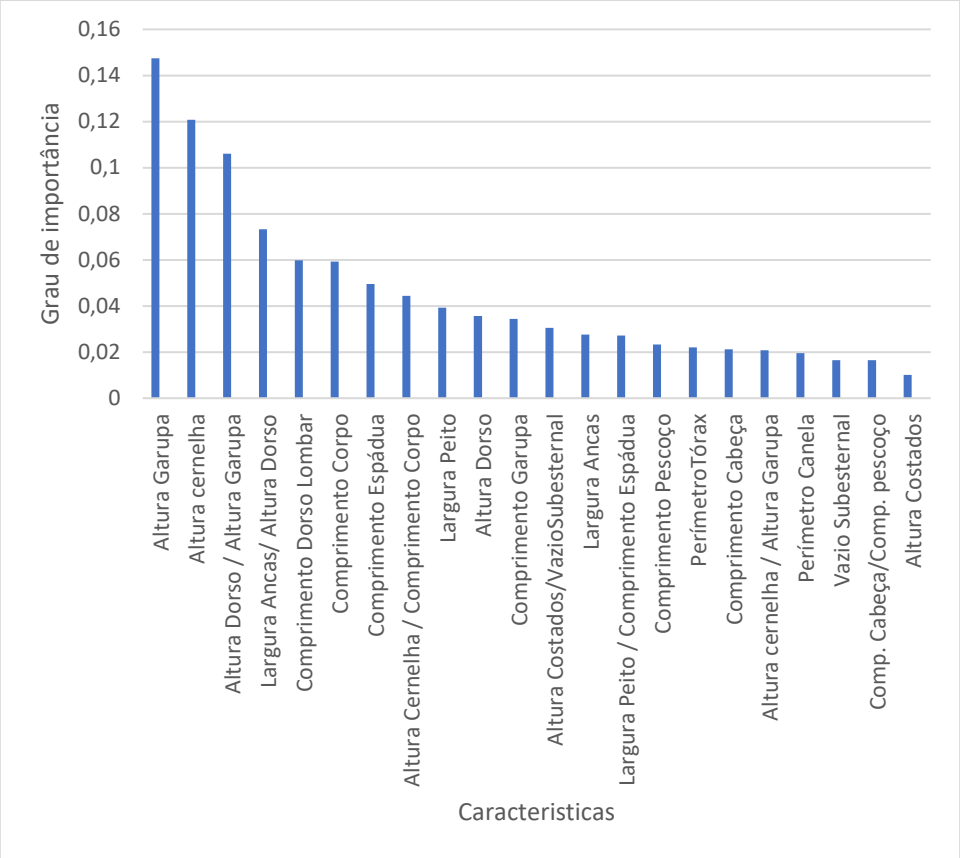
Este capítulo aborda os experimentos realizados bem como a apresentação e a análise dos resultados obtidos.

A Tabela 5 representa o grau de importância ou de pureza obtido de cada característica e a Figura 20 mostra os mesmos dados em um gráfico em ordem, do mais informativo para o menos. Sendo que os valores variam entre 0 e 1.

Tabela 5- Grau de importância de cada característica

Caraterísticas	Grau de importância
Comprimento Pescoço	0,02333932
Comprimento Cabeça	0,02120999
Comprimento Corpo	0,05924912
Altura Dorso	0,03568574
Largura Peito	0,03928991
Altura Costados	0,01011602
Altura Garupa	0,14739693
Comprimento Espádua	0,04959079
Perímetro Canela	0,01948966
Perímetro Tórax	0,02207603
Comprimento Garupa	0,03433967
Comprimento Dorso Lombar	0,05987937
Altura cernelha	0,12075817
Largura Ancas	0,02753056
Vazio Subesternal	0,01651528
Peito/ Comprimento espádua	0,02713029
Altura Dorso / Altura Garupa	0,10601755
Altura cernelha/ Altura Garupa	0,0207898
Altura Costados/Vazio Subesternal	0,03045792
Altura Cernelha/ Comp. Corpo	0,04445781
Comp. Cabeça / Comp. pescoço	0,01649602
Largura Ancas/Altura Dorso	0,07324375

Figura 20 - Gráfico do grau de importância de cada característica em ordem



Por meio do gráfico, é possível visualizar que a relação de duas medidas é altamente relevante e muitas delas possuem grau de importância até maior que as próprias medidas lineares. Portanto, usar apenas as quinze medidas lineares não se torna suficiente para avaliação dos cavalos pantaneiros.

Uma vez obtido o grau de importância de cada atributo, realizou-se testes usando, inicialmente, todas as medidas lineares e depois foram subtraídas uma a uma, a menos importante, resultando em conjuntos de atributos diferentes. É importante destacar que, quando o atributo é uma relação de duas medidas, considera-se utilizar as duas medidas no conjunto formado.

Por exemplo, segundo a Figura 20, as três características mais relevantes são altura da garupa, altura da cernelha e a relação altura dorso/altura garupa. Desta maneira, o conjunto de medidas lineares necessárias para formar este grupo é: altura da garupa, altura da cernelha e a altura do dorso. Ou seja, é preciso incluir o atributo altura do dorso para que a relação altura dorso/altura garupa exista.

Vale lembrar também, que embora o conjunto seja formado por, por exemplo, sete medidas das quinze existentes, pode ser que haja um número de características a mais devido às relações que podem estar presentes neste conjunto. Por exemplo, o conjunto apresentado no parágrafo anterior (altura da garupa, altura da cernelha, altura do dorso e altura dorso/altura garupa) é formado por três medidas lineares, porém apresenta quatro atributos ou características necessárias para o aprendizado do algoritmo.

Além disso, é preciso destacar que todos os conjuntos, sem exceção, possuem o atributo sexo que é indispensável, visto que as éguas possuem um padrão de medidas diferente podendo ser menores que os machos, e ainda assim estar dentro do padrão racial desejado.

Os conjuntos de características finais formados, respeitando o grau de importância de cada uma, para a realização dos testes com os algoritmos são representados abaixo. Sendo que o menor grupo formado é composto por quatro medidas lineares. Visto que um número ainda menor que este se torna excessivamente baixo.

Grupo de 4 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso e largura das ancas;

Grupo de 5 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas e comprimento dorso lombar.

Grupo de 6 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar e comprimento do corpo.

Grupo de 7 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua e altura da cernelha/comprimento do corpo.

Grupo de 8 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo e largura do peito.

Grupo de 9 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo; largura do peito e comprimento da garupa.

Grupo de 10 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo; largura do peito; comprimento da garupa; altura costados/vazio subesternal; vazio subesternal e altura costados;.

Grupo de 11 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo; largura do peito; comprimento da garupa; altura costados/vazio subesternal; vazio subesternal; altura costados; largura peito/comprimento espádua e comprimento do pescoço.

Grupo de 12 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo; largura do peito; comprimento da garupa; altura costados/vazio subesternal; vazio subesternal; altura costados; largura peito/comprimento espádua; comprimento do pescoço e perímetro do tórax.

Grupo de 13 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo; largura do peito; comprimento da garupa; altura costados/vazio subesternal; vazio subesternal; altura costados; largura peito/comprimento espádua; comprimento do pescoço e perímetro do tórax.

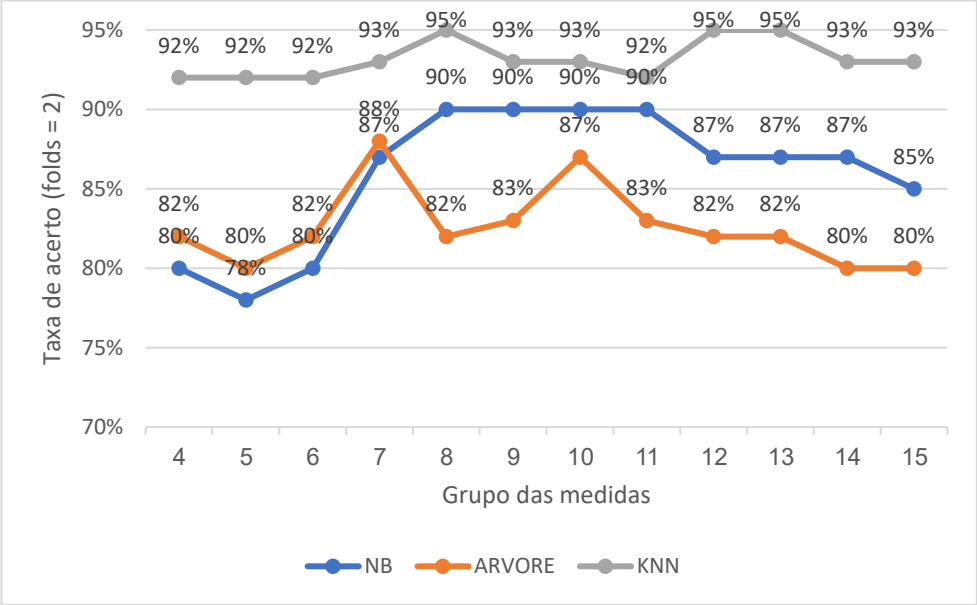
Grupo de 14 medidas: altura da garupa; altura da cernelha; altura dorso/altura garupa; largura ancas/altura dorso; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua; altura da cernelha/comprimento do corpo; largura do peito; comprimento da garupa; altura costados/vazio subesternal; vazio

subesternal; altura costados; largura peito/comprimento espádua; comprimento do pescoço; perímetro do tórax e comprimento da cabeça.

Por último, o grupo de 15 é formado por todas as medidas lineares e todas as relações existentes.

Com os conjuntos de características prontos, inicia-se a fase de treinamento e testes usando os algoritmos de classificação escolhidos. Percebe-se que a taxa de acerto da classificação variou entre 73% e no máximo 95% de forma geral. A Figura 21 mostra o desempenho do KNN, Naive Bayes e Árvore de decisão, conforme o aumento do número de características, usando-se apenas 2 folds.

Figura 21 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 2 folds



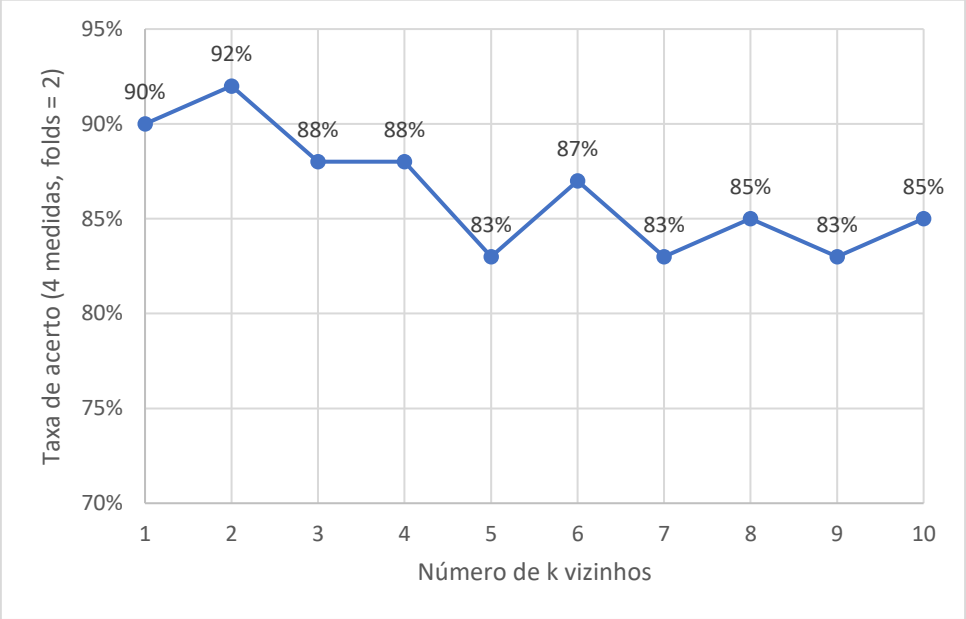
É, claramente, visível que o KNN é o algoritmo com taxa de acerto maior em todos os grupos de medidas diferentes, possuindo pouca variação, de 92% a 95% de acurácia. Sendo que o KNN alcançou a taxa de 95% nos grupos de medidas 8, 12 e 13. Outro ponto importante é que a taxa de 95% não se repetiu em nenhum outro caso além destas três ocorrências. Ou seja, k-fold =2 foi a melhor divisão para os dados. Isto significa que toda a base foi dividia em duas partes. Cada uma das partes utiliza 90% para treino e 10% para teste, ou seja, 20% para teste do total da base de dados, totalizando 60 dados para teste.

Ao analisar os gráficos, percebe-se que não há uma grande variação entre os

valores das taxas de acerto ao usar 2 folds, havendo uma diferença de apenas 5% entre a taxa mais baixa e a mais alta para este caso (2 folds), sendo a mais alta de 95%. Porém vale lembrar que tem-se disponível apenas um total de 300 dados para treino e teste.

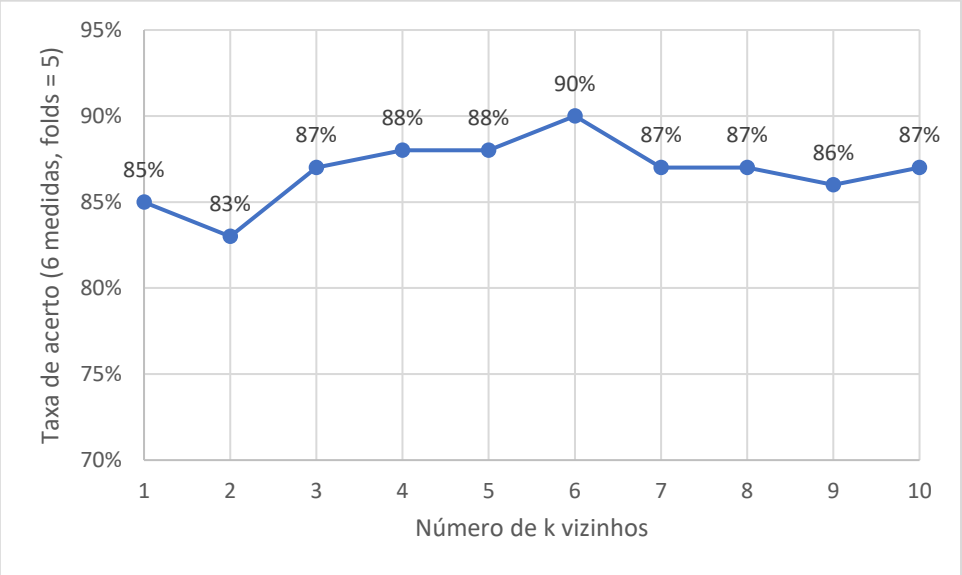
É importante destacar que todas as taxas de acerto mostradas neste gráfico da Figura 21 usam o melhor k de vizinhos para o KNN, já que foram testados valores de 1 a 10, separadamente, para depois comparar com os outros dois algoritmos. Abaixo, há um dos testes realizados para encontrar o melhor k de vizinhos para KNN aplicado no conjunto de 4 medidas e usando 2 folds, conforme a Figura 22.

Figura 22 - Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 2 folds



No gráfico da Figura 22, a maior taxa de acerto foi de 92%, usando k = 2. Por meio deste gráfico, é possível observar também que, de maneira geral, ao aumentar o número de k, a taxa de acerto tende a cair, para a maioria dos casos, salvo alguns casos como o da Figura 23, em que a taxa de acerto aumenta na metade e depois decai, porém a variação não é muito alta. Além disso, percebe-se que a maior taxa de acerto, neste caso (grupo de 6 medidas e 5 folds) é de 90% e a mais baixa de 83%.

Figura 23 - Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 5 folds

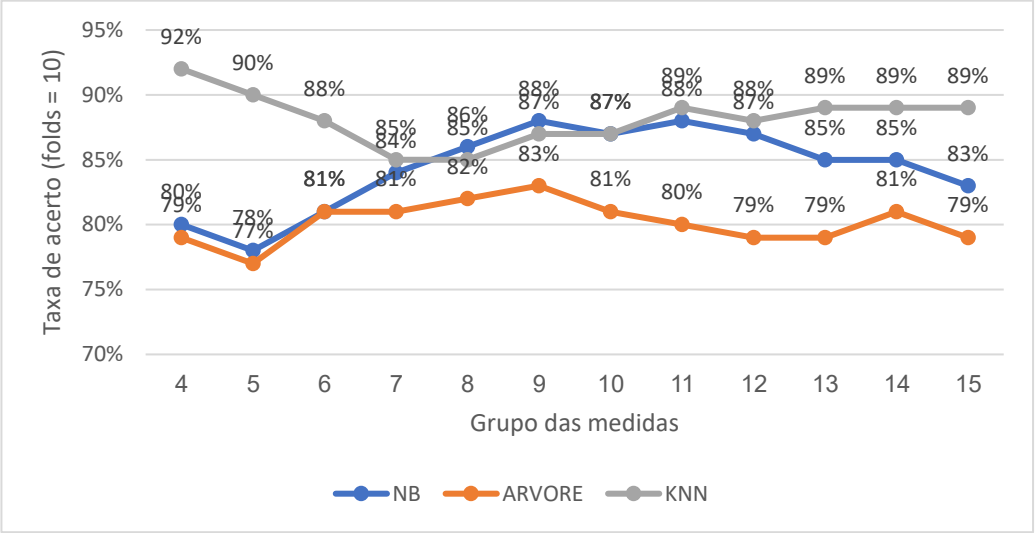


Todos os gráficos gerados para os diferentes folds, grupos de medidas e número k de vizinhos para o KNN são representados no apêndice.

Ainda, segundo o gráfico da Figura 21, é possível observar que o Naive Bayes (NB) teve como maior taxa de acerto de 90% e a Árvore de decisão alcançou 88% de acerto para o mesmo caso de 2 folds. Praticamente, em todos os testes, o NB ficou na segunda colocação e a Árvore de decisão em último, considerando a acurácia geral.

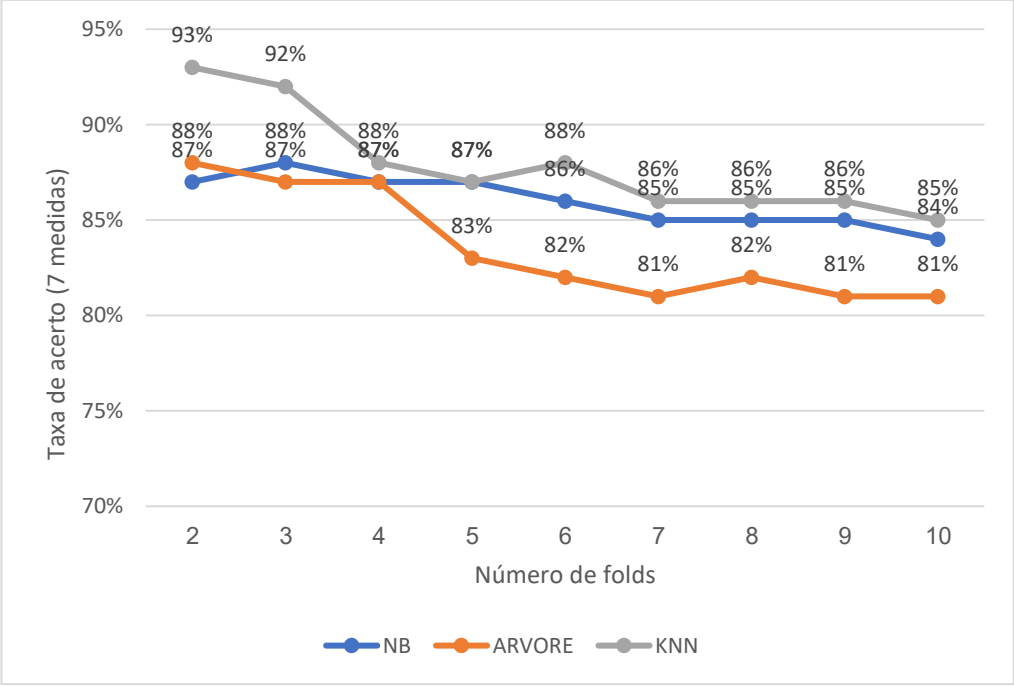
Em alguns raros casos, o Naive Bayes conseguiu competir com o KNN e ultrapassar a sua taxa de acerto, embora a diferença seja mínima, como é mostrado no gráfico da Figura 24. A acurácia do NB é de 86% no grupo de 8 medidas, enquanto o KNN atingiu 85%. Também é possível ver que o comportamento do Naive Bayes é aumentar a taxa de acerto, a medida que o número de características aumenta, e cair no final, formando quase uma curva. Este comportamento é mantido em todos os casos dos folds de 2 a 10. Todos estes casos são apresentados no apêndice para verificação.

Figura 24 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 10 folds



O seguinte gráfico da figura 25 mostra o comportamento da taxa de acerto dos três algoritmos conforme o aumento do número de folds. Para o grupo de medidas igual a 4, 5, 6, 11, 13 e 15 a taxa de acerto, de maneira geral, se manteve estável com o aumento do número de folds. Já para todos os outros grupos, o comportamento era diminuir a taxa de acerto ao aumentar o número de folds.

Figura 25 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 7 medidas com o aumento de folds de 2 a 10



Conforme foi explanado no capítulo 2.2.4, sobre a maldição da dimensionalidade, à medida que que o número de características diminui, menor é a complexidade e maior a taxa de acerto. Nos casos obtidos, nesta pesquisa, apenas Naive Bayes e Árvore de Decisão se comportaram desta maneira. Em contrapartida, para KNN, a taxa de acerto cai inicialmente, ao aumentar o número de características, mas depois começa a subir novamente. Este foi o comportamento geral ao empregar diferentes folds, conforme pode ser visto nos gráficos das figuras apresentados anteriormente, as figuras 26 e 27 mostradas abaixo. Além disso, todos os outros gráficos se encontram no apêndice.

Figura 26 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 8 folds

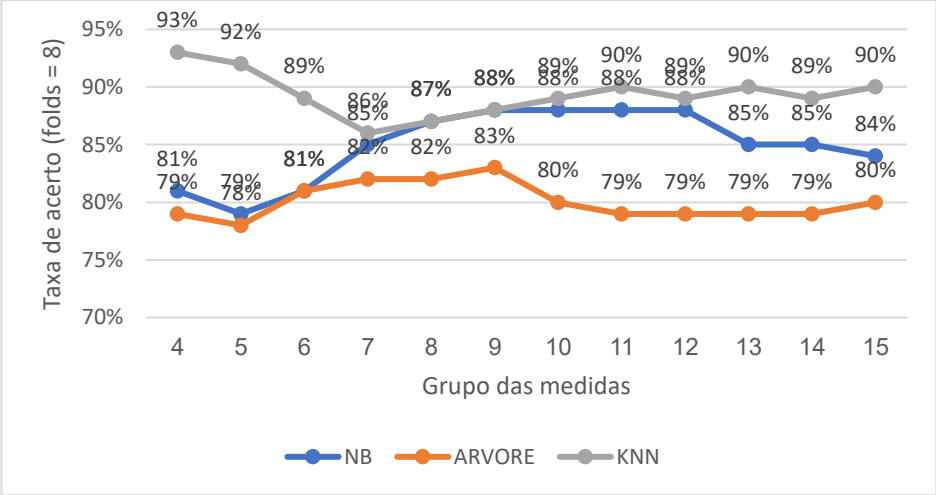
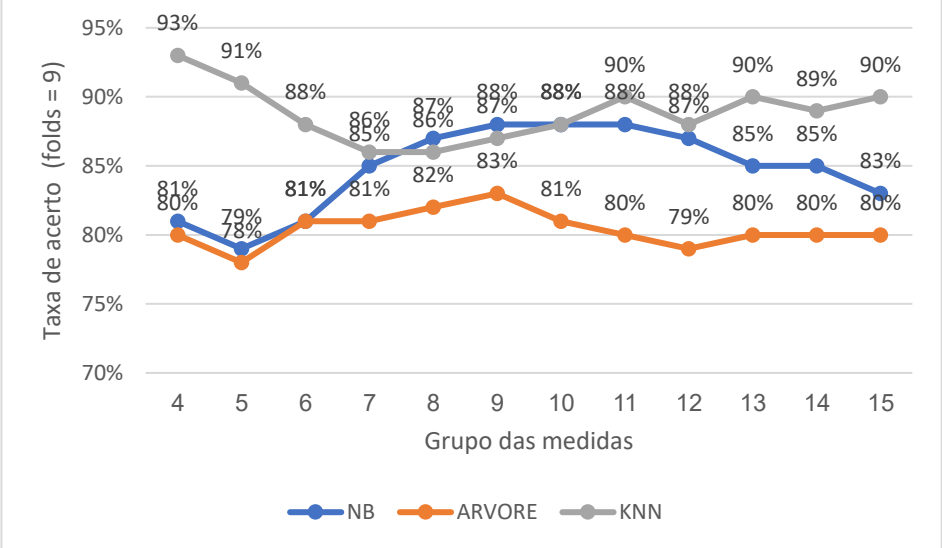


Figura 27 - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 9 folds



Ao analisar todos os gráficos que comparam a acurácia dos três algoritmos de classificação, é nítido que o classificador K-Nearest Neighbor (KNN) foi o melhor entre os três, obtendo as taxas de acerto mais altas em quase todos os casos testados. Por isso, KNN foi escolhido para realizar a classificação dos cavalos no aplicativo.

A fim de verificar quais classes o algoritmo erra mais durante a classificação, foi extraída a matriz de confusão, conforme mostra a Tabela 6. Por meio desta Tabela, é possível verificar que os 20% dos dados destinados ao teste representam um total de 60 cavalos, sendo que o algoritmo selecionou, aleatoriamente, 13 cavalos de baixa qualidade, 26 de média e 21 de alta qualidade. Ao analisar os dados da Tabela, percebe-se que o algoritmo KNN classificou dois cavalos de alta qualidade como sendo de média qualidade e classificou um cavalo de baixa qualidade na classe de alta qualidade. Já o restante dos dados de teste foi classificado corretamente.

Tabela 6 - Matriz de confusão da classificação do algoritmo KNN

		Classificação do algoritmo		
	Qualidade	baixa	Média	alta
Classe real	baixa	12	0	1
	média	0	26	0
	alta	0	2	19

Como mencionado, anteriormente, a taxa de acerto mais alta foi de 95%, usando o algoritmo KNN e k-folds = 2. No entanto, obtiveram-se três ocorrências desta taxa. A primeira aparece no grupo de 8 medidas, a segunda no de 12 e a última no grupo de 13 medidas. Visto que a taxa de acerto para estes três grupos foi a mesma, escolheu-se o grupo de 8 medidas, já que este apresenta menos medidas lineares para a classificação, o que torna a classificação menos custosa computacionalmente e menos trabalhoso para quem realiza a medição do cavalo. Além disso, o grupo de 8 medidas utiliza k-vizinhos igual a 2 para atingir esta taxa de acerto. A Tabela 7 apresenta a configuração final para a melhor classificação.

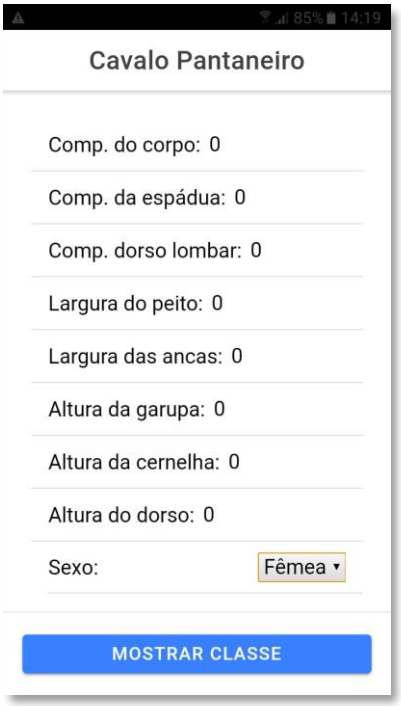
Tabela 7 - Resultado da configuração obtida para classificar os cavalos pantaneiros

Algoritmo escolhido	**k-vizinhos**	**k-folds**	**Grupo de medidas**	**Taxa de acerto**
KNN	2	2	8	95%

Portanto, a base de dados final é constituída pelas seguintes características: As 8 medidas lineares altura da garupa; altura da cernelha; altura do dorso; largura das ancas; comprimento dorso lombar; comprimento do corpo; comprimento da espádua e largura do peito. Assim como as três relações contidas no grupo de 8 medidas: altura dorso/altura garupa; largura ancas/altura dorso e altura da cernelha/comprimento do corpo. E finalmente, o sexo de cada cavalo da base e a sua respectiva classe.

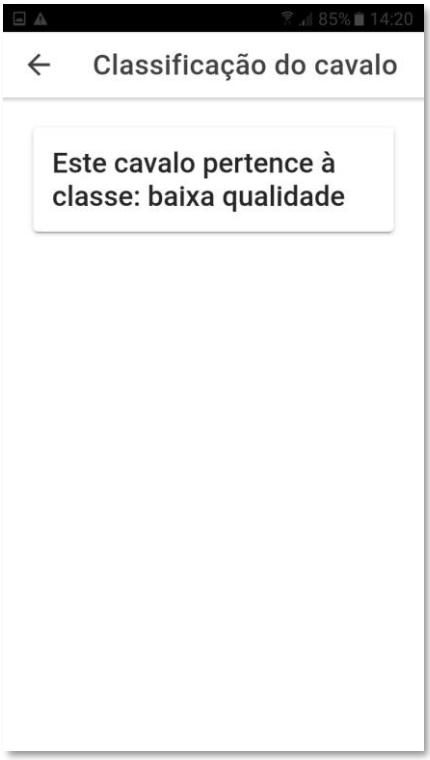
Após escolhidos o algoritmo de classificação e as características, desenvolveu-se o aplicativo, o qual é implementado usando a configuração mostrada acima, na tabela 7. A interface principal da aplicação pode ser vista na figura 28. Nesta tela inicial, o usuário informa as 8 medidas solicitadas e o sexo do cavalo. Vale lembrar que o algoritmo utiliza também as três relações que são simples divisões de algumas das 8 medidas informadas no formulário.

Figura 28 - Interface principal do aplicativo para inserir as medidas do cavalo



Com os dados inseridos e após apertar o botão “Classificar Cavalo” o KNN calcula a distância euclidiana em relação a todos os dados de cavalos disponíveis na base armazenada no dispositivo e utiliza os dois cavalos vizinhos mais próximos para realizar a classificação. Assim, o aplicativo mostra o resultado da classificação que pode ser de alta, média ou baixa qualidade, conforme ilustra a figura 29.

Figura 29 - Interface do aplicativo mostrando a classificação do cavalo



Para validação da aplicação, este foi levado ao campo com uma especialista, a qual realizou a medição de quatro cavalos no total que estavam disponíveis e todos foram classificados como de média qualidade por ela. Ao inserir as 8 medidas e o sexo, o aplicativo foi capaz de classificar, com sucesso, os animais na classe predita pela especialista.

O próximo e último capítulo é sobre a conclusão da presente pesquisa e trabalhos futuros.

5. Considerações Finais

A inclusão de novas ferramentas tecnológicas para auxiliar os usuários no campo de agronegócios, torna-se cada vez mais importante, proporcionando muito mais facilidade, comodidade, produtividade e agilidade no dia a dia. Esse documento apresentou uma aplicação de técnicas de aprendizado de máquina visando classificar os cavalos pantaneiros em alta, média e baixa qualidade por meio de uma aplicação móvel.

Constatou-se que o método *feature_importances*, que utiliza a teoria da entropia para calcular o grau de importância de cada característica como critério para seleção das melhores características, teve resultados bastante úteis. Obteve-se taxas de acerto acima de 80% dos modelos de classificação no conjunto de dados. Os testes e resultados confirmam que a etapa de seleção de características é bastante importante para a construção dos modelos.

Além disso, esta seleção é de suma importância durante o processo de medição dos cavalos em campo. Visto que estes animais nem sempre são mansos. Portanto quanto menos tempo o usuário passar medindo o cavalo, menos trabalhoso e menos perigoso se torna este processo.

Quanto aos resultados obtidos pelos classificadores KNN, Árvore de Decisão (C4.5) e o Naive Bayes observam-se diferentes resultados em testes anteriores e pós-seleção, embora não tão discrepantes. Em especial, nota-se que o algoritmo KNN foi o melhor entre os três comparados, apresentando como maior taxa de acerto igual a 95%. Já a Árvore de Decisão (C4.5) teve as taxas de acerto mais baixas entre eles, sendo que a taxa máxima obtida foi de 87%. Esse fato pode estar diretamente relacionado à natureza das características, onde cada algoritmo utilizado, no presente trabalho, lida com elas de forma diferente.

Ao longo desta pesquisa, enfrentou-se algumas dificuldades e, sem dúvida, a maior delas é a quantidade de dados disponibilizados. Portanto, conclui-se que o treinamento foi possível e com a adição futura de mais dados das diversas classes será possível uma melhora dos resultados. E a fim de evitar o *Overfitting*, que pode ser causado por conjuntos insuficientes de dados e alta dimensionalidade, utilizou-se três técnicas que são a validação cruzada, balanceamento dos dados e a seleção de características.

É impossível dizer qual a quantidade exata necessária para a classificação nesta pesquisa. A quantidade de dados necessária para cada caso pode variar dependendo da tarefa a ser executada. Porém, segundo Mitchell (1997) quanto mais dados, melhor se torna o aprendizado dos modelos.

Portanto, há a necessidade de conseguir o máximo possível de dados para repetir os testes. Especificamente, são necessários mais exemplares de cavalos fora do padrão, visto que se tem disponível apenas 100 amostras desta classe. Já os de alta e média qualidade somam, juntos, mais de 3500 dados. Isto se deve ao fato de os dados de cavalos fora do padrão serem descartados, visto que eles não podem ser registrados na ABCCP. Muitas vezes, os cavalos fora do padrão nem chegam a ser medidos, uma vez que os especialistas conseguem identificar alguns casos fora do padrão racial, somente ao olhar para o animal.

Apesar da taxa de acerto ser, consideravelmente, alta, é necessário descobrir a aceitação desta taxa pelo público alvo (criadores de cavalos pantaneiros, apaixonados e leigos em geral). Por isso, a validação com estes usuários deve ser realizada.

Portanto, espera-se em um primeiro momento tornar o aplicativo um instrumento a ser utilizado por especialistas para realizar a medição e cadastro de cavalos novos. Desta forma, o especialista poderá confirmar a resposta retornada pelo aplicativo para cada cavalo e com isso, é possível verificar a taxa de erro da aplicação. Para tanto, como trabalho futuro, o aplicativo deve ser aprimorado para possibilitar o cadastro de animais e seu armazenamento em um banco de dados online, de maneira a possibilitar o armazenamento temporário no dispositivo móvel do usuário em caso de falta de rede de internet no local.

Além disso, é possível incluir um tutorial, no próprio aplicativo, para os leigos no assunto, em forma de imagens contendo textos explicativos sobre o passo a passo para aferir cada medida, antes de inserir a mesma.

Como possibilidade para trabalho futuro também, é interessante desenvolver um método de medição do cavalo usando a câmera de um dispositivo móvel. Desta maneira, o usuário não precisa entrar em contato com a animal, podendo capturar as medidas a uma distância segura do cavalo. Isto pode ser realizado por meio da visão computacional que é uma área capaz de lidar com informações de imagens e vídeos digitais.

Referências

Anaconda. **Anaconda Navigator.** Disponível em <https://docs.anaconda.com/anaconda/navigator/> Acesso em: 20 nov. 2019.

BALIEIRO, E. de S. **Subsídios ao estudo do cavalo pantaneiro.** Anuário, CCCN, São Paulo, 1971.

BATISTA, Gustavo Enrique de Almeida Prado et al. **Pré-processamento de dados em aprendizado de máquina supervisionado**. 2003. Tese de Doutorado. Universidade de São Paulo.

BECK. S.L. Eqüinos - ração, manejo e equitação. São Paulo: Criadores,1985.

Breiman, L. Random forests. Machine learning, v. 45, n. 1, p. 5-32, 2001.

CHAPELLE, Olivier; SCHOLKOPF, Bernhard; ZIEN, Alexander. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. **IEEE Transactions on Neural Networks**, v. 20, n. 3, p. 542-542, 2009.

CORRÊA FILHO, E.A. O cavalo pantaneiro. Revista Medicina Veterinária,São Paulo, v.8, n.3, p.395-412, 1973.

CREPALDI, Paola Guarisso et al. Um estudo sobre a árvore de decisão e sua importância na habilidade de aprendizado. 2011.

da Silva, L. A. C., Santos, S. A., Silva, R. A. S., Mc Manus, C., Petzold, H. **Adaptação do cavalo pantaneiro ao estresse da lida diária de gado no pantanal, Brasil**. Archivos de Zootecnia [en linea]. 2005, 54(206-207), 509-513[fecha de Consulta 12 de Febrero de 2020]. ISSN: 0004-0592. Disponible en: https://www.redalyc.org/articulo.oa?id=49520761

DASH, Manoranjan; LIU, Huan. Feature selection for classification. **Intelligent data analysis**, v. 1, n. 1-4, p. 131-156, 1997.

Decker, K. M. & S. Focardi (1995). Technology overview: A report on data mining. Technical Report CSCS TR-95-02, CSCS-ETH, Swiss Scientific Computing Center.

DOMINGUES, O. Contribuição ao estudo do cavalo pantaneiro. Rio de Janeiro: MNLZ, 1957.

DUDA, R. O.; HART, P. E.; STORK, D. G. Pattern classification. 2001.

Everitt, B. S. (1993). Cluster Analysis. Edward Arnold, 3 edition.

FERRERO, Carlos Andres. **Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia**. 2009. Tese de Doutorado. Universidade de São Paulo.

Friedman, J; Hastie, T; Tibshirani R. **The elements of statistical learning**, v. 1. Springer series in statistics Springer, Berlin, 2001.

Grupo Arariboia. (1989) "Inteligência Artificial: um Curso Prático", Editora LTC, Rio de Janeiro.

GIANNONI, M.A. **métodos de melhoramento genético e sistemas de acasalamentos aplicados aos eqüinos**. Jabotical, FUNEP, 1988. 63p.

GUYON, Isabelle; ELISSEEFF, André. An introduction to variable and feature selection. **Journal of machine learning research**, v. 3, n. Mar, p. 1157-1182, 2003.

IBGE. Produção da Pecuária Municipal–2015. 2015.

Ingalls, D., Palacz, K., Uhler, S., Taivalsaari, A., & Mikkonen, T. (2008, May). **The lively kernel a self-supporting system on a web page. In *Workshop on Self-sustaining Systems*** (pp. 31-50). Springer, Berlin, Heidelberg.

Ionic Framework. **What is Ionic Framework?** Disponível em <https://ionicframework.com/docs/intro> Acesso em: 20 nov. 2019.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor. &Robert Tibshirani,"An Introduction to Statistical Learning". 2013.

JOHN, G. H.; LANGLEY, P.: **Estimating continuous distributions in bayesian classifiers.** In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pages 338-345. Morgan Kaufmann Publishers Inc, 1995.

Jupyter. **The Jupyter Notebook.** Disponível em < https://jupyter.org/> Acesso em: 20 nov. 2019.

KÜMMEL, Bruno Couto. Método baseado em aprendizado de máquina para seleção de características para distinção entre RNAs não-codificadores longos e RNAs codificadores

de proteínas. 2017.

LANGLOIS, B.; MINKEMA. D.; BRUNS, E. Genetic problems in horse. Livestock Production Science, v.10, p.69-81. 1983.

LIMA, RA de S.; CINTRA, A. G. Revisão do estudo do complexo do agronegócio do cavalo. **Ministério da Agricultura, Brasília, DF**, 2015.

McCallum, A., K. Nigam, & L. H. Ungar (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In R. Ramakrishnan, S. Stolfo, R. Bayardo, & I. Parsa (Eds.), Proceedinmgs of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining — KDD-00, New York,USA, pp. 169–178. ACM Press.

MCMANUS, C.; MISERANI, M. G. G.; SANTOS, S. A.; MARIANTE, A. S.;SILVA, J. S.; ABREU, U. G. P.; MAZZA, M. C. M.; SERENO, J. R. B. **Índices corporais do cavalo pantaneiro**. In: REUNIÃO ANUAL DA SOCIEDADEBRASILEIRA DE ZOOTECNIA, 38., 2001, Piracicaba, SP. Anais... Piracicaba:FEALQ, 2001. p.559-560.

MITCHELL, T. M.: **Machine Learning**, 1997.

Monard, M. C., Batista, G. E., Kawamoto, S. and Pugliesi, J.B. (1997) “Uma Introdução ao Aprendizado Simbólico de Máquina por Exemplos”, Nota Didática do ICMC-USP, n.° 20

PANDAS. **Python Data Analysis Library.** Disponível em <https://pandas.pydata.org/>. Acesso em: 20 nov. 2019.

W3C. **About W3C**. Disponível em: < https://www.w3.org/Consortium/> Acesso em: 20 nov. 2019.

PYTHON. About. Disponível em: <https://www.python.org/about/> Acesso em: 20 nov. 2019.

RIBEIRO, Lucas Mikilita et al. **Desenvolvimento de biblioteca paralela em python para aplicações em engenharia biomédica**. 2011. Tese de Doutorado. UNIVERSIDADE DE SÃO PAULO.

SANTOS, S. A. et al. **Descrição do manejo geral de cavalos pantaneiros na região do**

Pantanal. Embrapa Pantanal-Boletim de Pesquisa e Desenvolvimento (INFOTECA-E), 2005.

SANTOS, S. A.; SALIS, S. M.; COMASTRI FILHO, J. A. Cavalo Pantaneiro: rústico por natureza. **Embrapa Informação Tecnológica: Brasilia, Brasil**, 2016.

SANTOS, S. A.; SILVA, R. A. M. S.; AZEVEDO, J. R. M.; MELLO, M. A. R.;SOARES, A. C.; SIBUYA, C. A.; ANARUMA, C. A. **Serum electrolyte and totalprotein alterations in Pantaneiro horse during long distance exercise**. Arquivo Brasileiro de Medicina Veterinária e Zootecnia, Belo Horizonte, v.53, n.3,p.351-357, 2001.

SANTOS, Sandra Aparecida, et al. Avaliacao e conservacao do cavalo pantaneiro. *Embrapa Pantanal-Circular Técnica (INFOTECA-E)*, 1995.

SCIKIT LEARN. **Machine Learning in Python.** Disponível em < https://scikit-learn.org/stable/> Acesso em: 20 nov. 2019.

SILVA, Maurício Samy. ***Criando sites com HTML: sites de alta qualidade com HTML e CSS*.** Novatec Editora, 2008.

SOUSA, R. T. **Avaliação de classificadores na classificação de radiografias de tórax para diagnóstico de pneumonia infantil**. Dissertação (Mestrado) — Programa de Pós-Graduação do Instituto de Informática da Universidade Federal de Goiás, 2013.

ZHANG, Shichao et al. Learning k for knn classification. **ACM Transactions on Intelligent Systems and Technology (TIST)**, v. 8, n. 3, p. 1-19, 2017.

ANEXOS

ANEXO A - Número de indivíduos, média, desvio padrão e coeficiente de variação das características de conformação, índices zootécnicos e relações corporais de equinos machos e fêmeas, durante os períodos de 26/07/72 a 29/09/91.

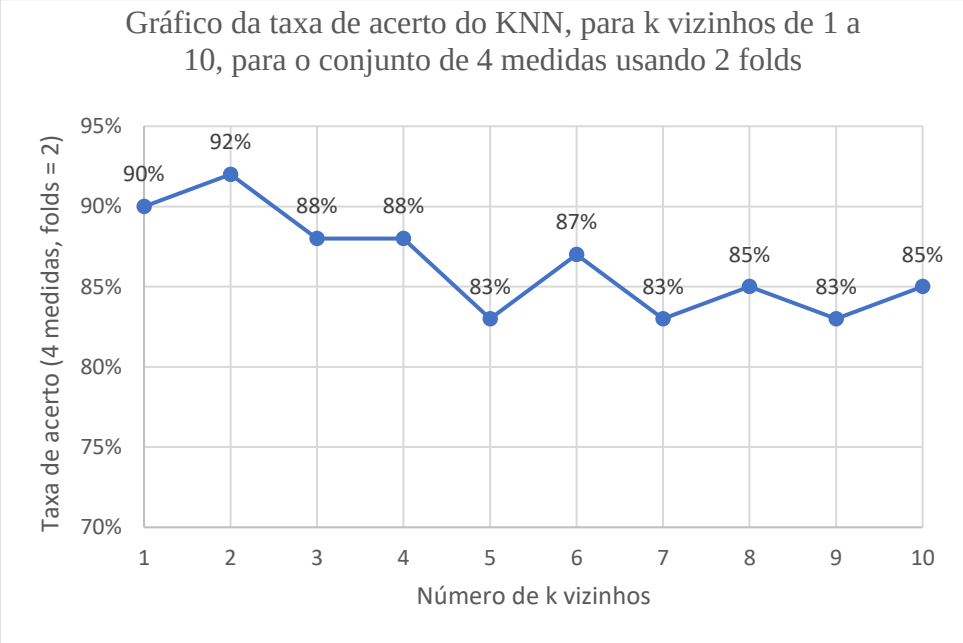
VARIÁVEIS OBSERVADAS	MACHOS			FÊMEAS		
	N	Média ± s	CV(%)	N	Média ± s	CV
Alturas (cm):						
Cernelha (AC)	180	141,7 ± 1,7	0,01	1035	136,9 ± 2,2	0,02
Dorso (AD)	180	138,2 ± 2,6	0,02	1032	134,5 ± 2,7	0,02
Garupa (AG)	180	141,8 ± 2,2	0,01	1034	137,9 ± 2,5	0,02
Costado (AT)	180	54,4 ± 2,5	0,05	1031	58,5 ± 3,5	0,06
Comprimentos (cm):						
Cabeça (CC)	180	54,4 ± 2,5	0,05	1034	53,6 ± 2,5	0,05
Pescoço (CP)	180	57,7 ± 4,9	0,08	1034	57,5 ± 6,0	0,10
Dorso (CD)	147	50,4 ± 8,1	0,16	914	42,6 ± 9,6	0,23
Gárupa (CG)	180	45,2 ± 2,0	0,06	1034	44,9 ± 2,8	0,06
Espádua (CE)	180	47,6 ± 3,6	0,08	1034	46,2 ± 3,4	0,07
Corpo (CR)	180	146,2 ± 22,4	0,15	1033	141,3 ± 5,1	0,04
Larguras (cm):						
Cabeça (LC)	180	20,9 ± 2,0	0,10	1033	20,5 ± 2,5	0,12
Peito (LP)	180	34,1 ± 3,1	0,09	1034	31,6 ± 3,0	0,09
Anca (LA)	180	43,5 ± 2,7	0,06	1034	43,5 ± 3,0	0,07

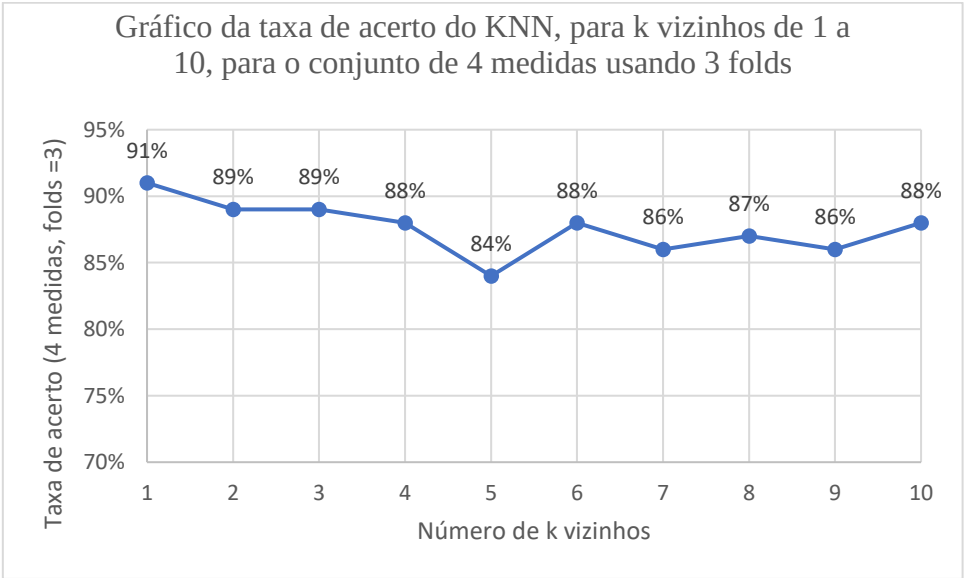
ANEXO B – Número de indivíduos, média, desvio padrão e coeficiente de variação das características de conformação, índices zootécnicos e relações corporais de equinos machos e fêmeas, durante os períodos de 26/07/72 a 29/09/91.
(Continuação)

VARIÁVEIS OBSERVADAS	MACHOS			FÊMEAS		
	N	Média ± s	CV(%)	N	Média ± s	CV(%)
Perímetros (cm):						
Tórax (Pt)	180	162,9 ± 4,6	0,03	1033	160,1 ± 6,1	0,04
Canela (Pc)	179	18,4 ± 1,1	0,06	1034	17,5 ± 0,9	0,05
Índices:						
Corporal (IC)	180	0,90 ± 1,4	0,16	1031	0,88 ± 0,1	0,03
Corporal relativo (ICR)	180	103,2 ± 16,0	0,15	1033	103,3 ± 3,5	0,03
Dáctilo-torácico (IDT)	179	0,113 ± 0,01	0,06	1033	0,109 ± 0,01	0,06
Peitoral (IP)	180	0,71 ± 0,06	0,08	1031	0,75 ± 0,07	0,09
Conformação (ICC)	180	187,1 ± 10,15	0,05	1033	187,4 ± 13,7	0,05
Relações:						
AC:CC	180	0,98 ± 0,06	0,06	1033	0,97 ± 0,03	0,03
Pt:AC	180	1,15 ± 0,03	0,03	1033	1,17 ± 0,04	0,04

APÊNDICES

APÊNDICE A - Gráficos da taxa de acerto do KNN, para k vizinhos de 1 a 10, para os conjunto de 4 a 15 medidas, usando de 2 a 10 folds





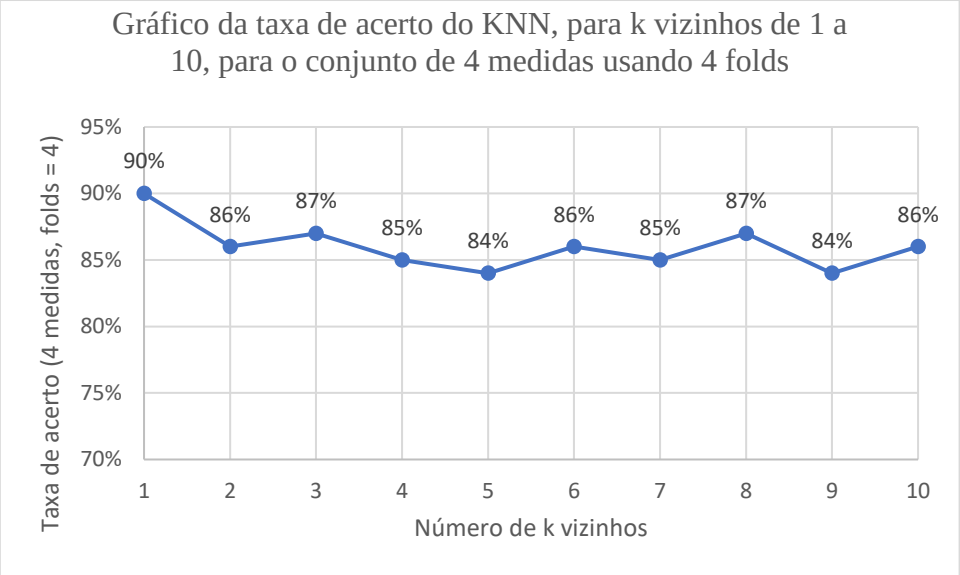


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 4 folds
Taxa de acerto (4 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
90%
86%
87%
85%
84%
86%
85%
87%
84%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

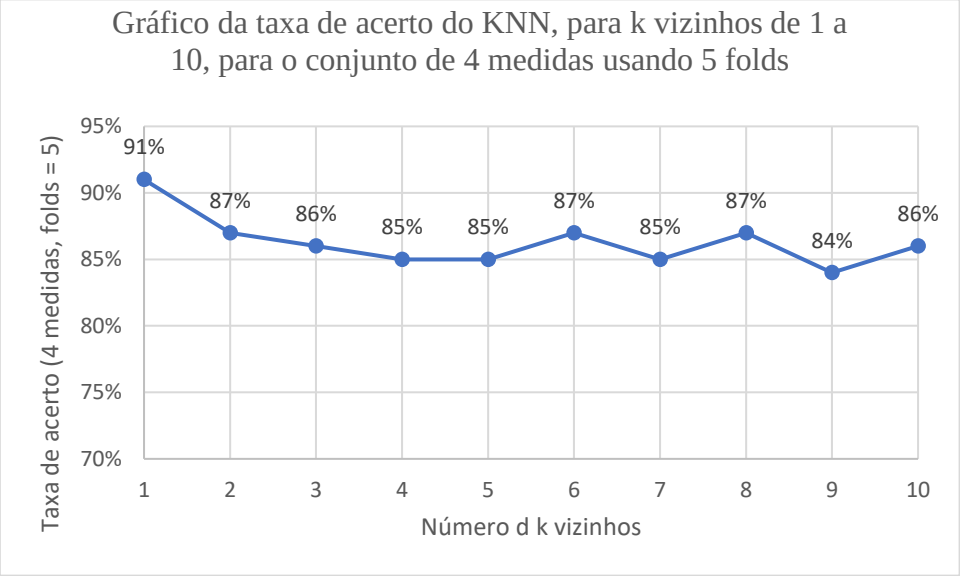


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 5 folds
Taxa de acerto (4 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
91%
87%
86%
85%
85%
87%
85%
87%
84%
86%
1
2
3
4
5
6
7
8
9
10
Número d k vizinhos

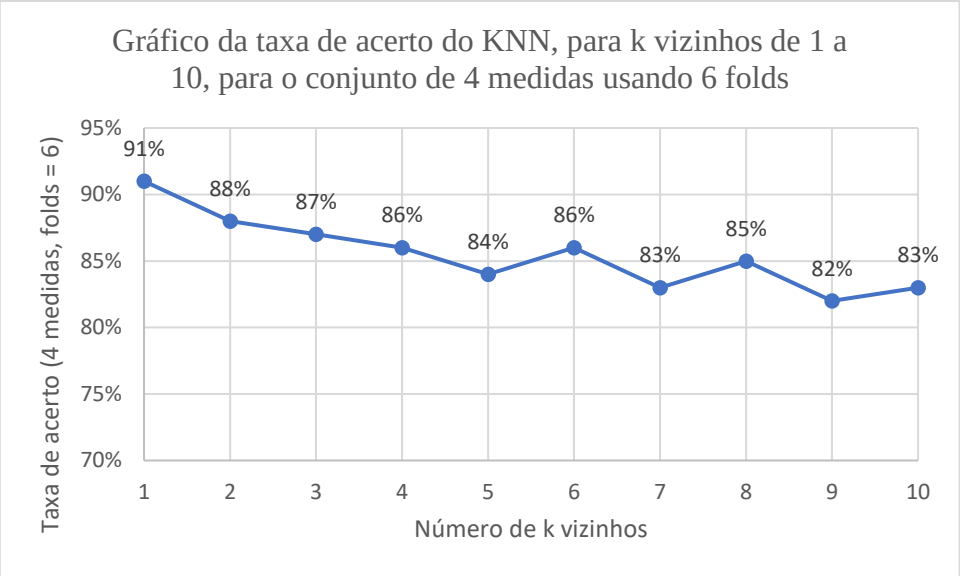


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 6 folds
Taxa de acerto (4 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
91%
88%
87%
86%
84%
86%
83%
85%
82%
83%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

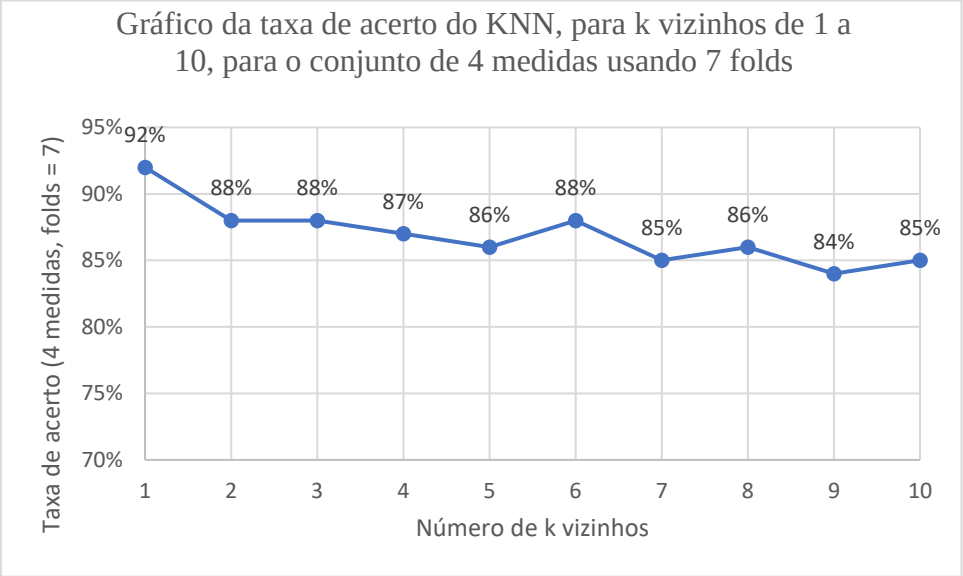


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 7 folds
Taxa de acerto (4 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
92%
88%
88%
87%
86%
88%
85%
86%
84%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

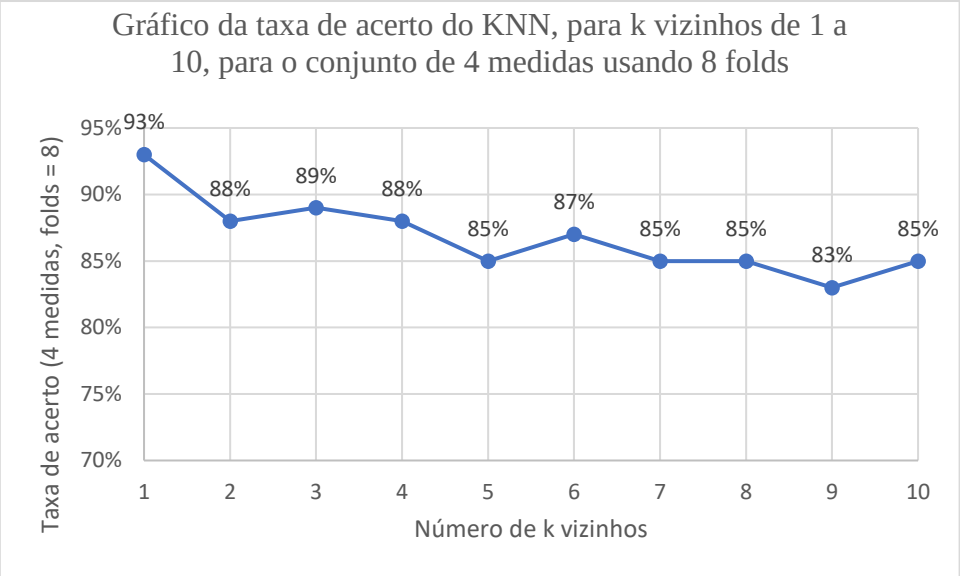


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 8 folds
Taxa de acerto (4 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
93%
88%
89%
88%
85%
87%
85%
85%
83%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

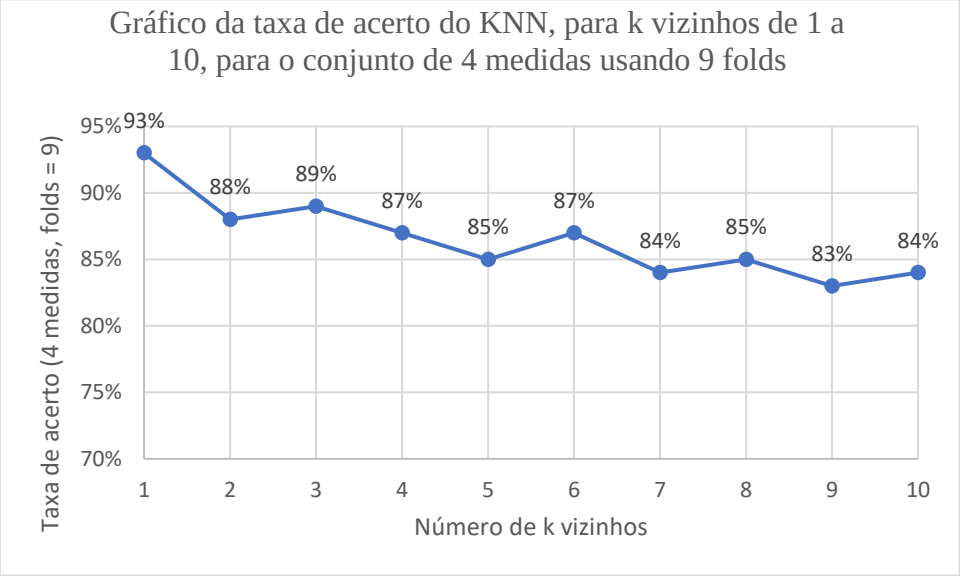


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 9 folds
Taxa de acerto (4 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
93%
88%
89%
87%
85%
87%
84%
85%
83%
84%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

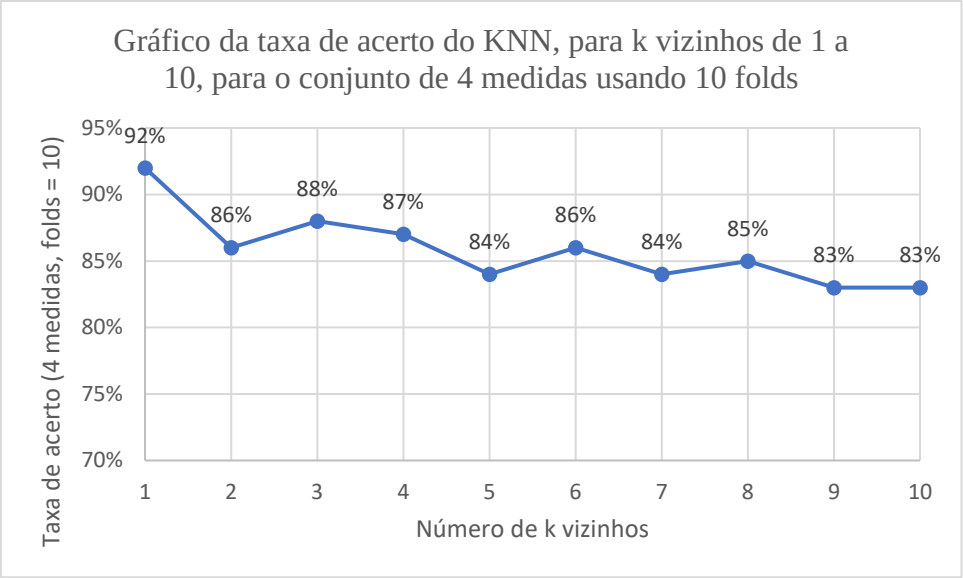


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 4 medidas usando 10 folds
Taxa de acerto (4 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
92%
86%
88%
87%
84%
86%
84%
85%
83%
83%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

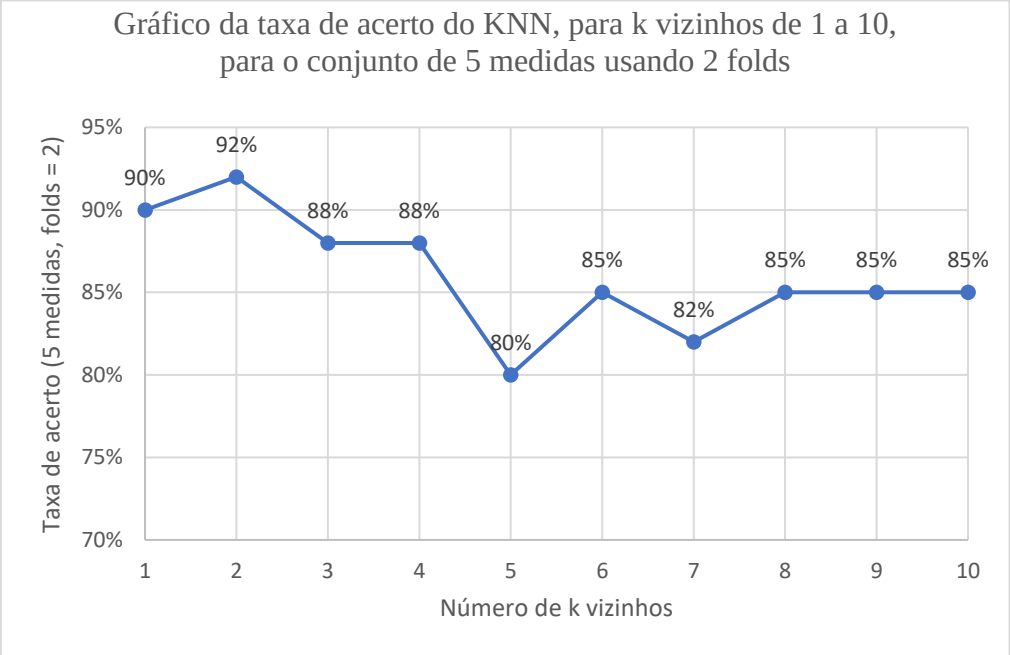


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 2 folds
Taxa de acerto (5 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
90%
92%
88%
88%
80%
85%
82%
85%
85%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

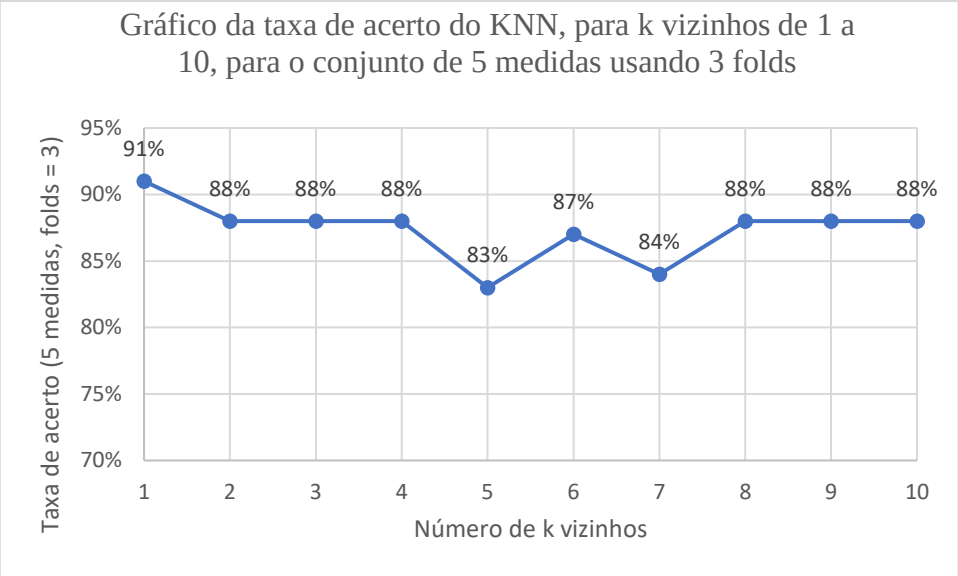


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 3 folds
Taxa de acerto (5 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
91%
88%
88%
88%
83%
87%
84%
88%
88%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

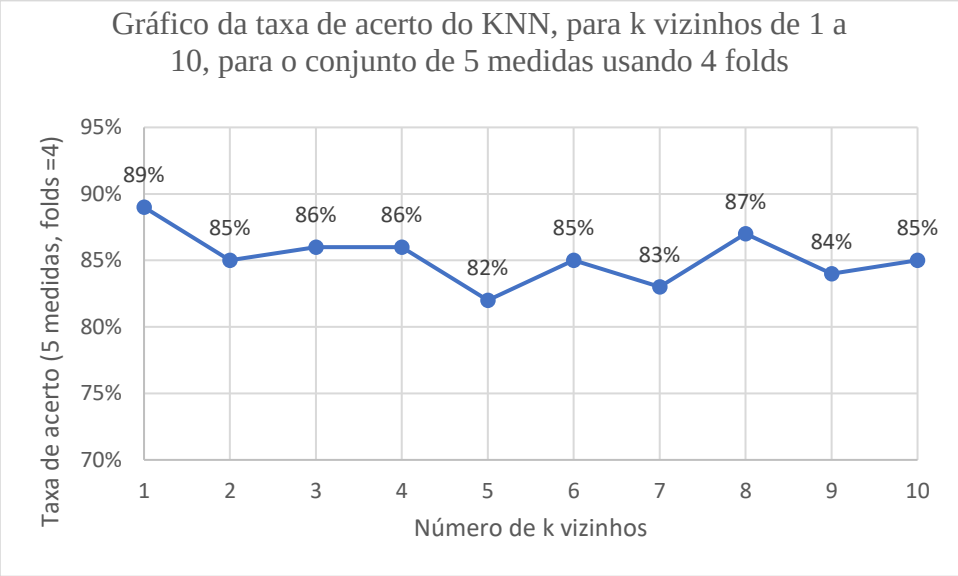


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 4 folds
Taxa de acerto (5 medidas, folds =4)
95%
90%
85%
80%
75%
70%
89%
85%
86%
86%
82%
85%
83%
87%
84%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

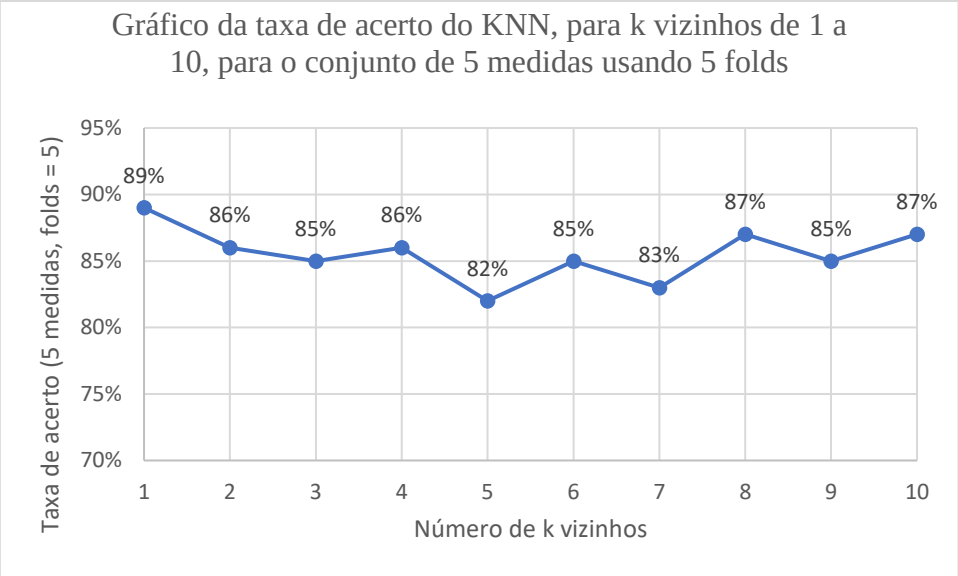


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 5 folds
Taxa de acerto (5 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
89%
86%
85%
86%
82%
85%
83%
87%
85%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

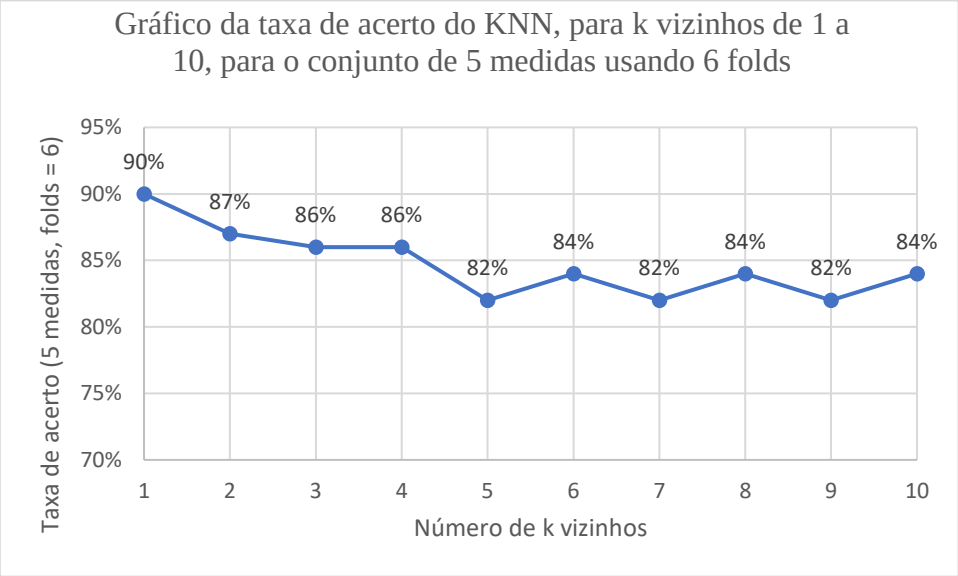


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 6 folds
Taxa de acerto (5 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
90%
87%
86%
86%
82%
84%
82%
84%
82%
84%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

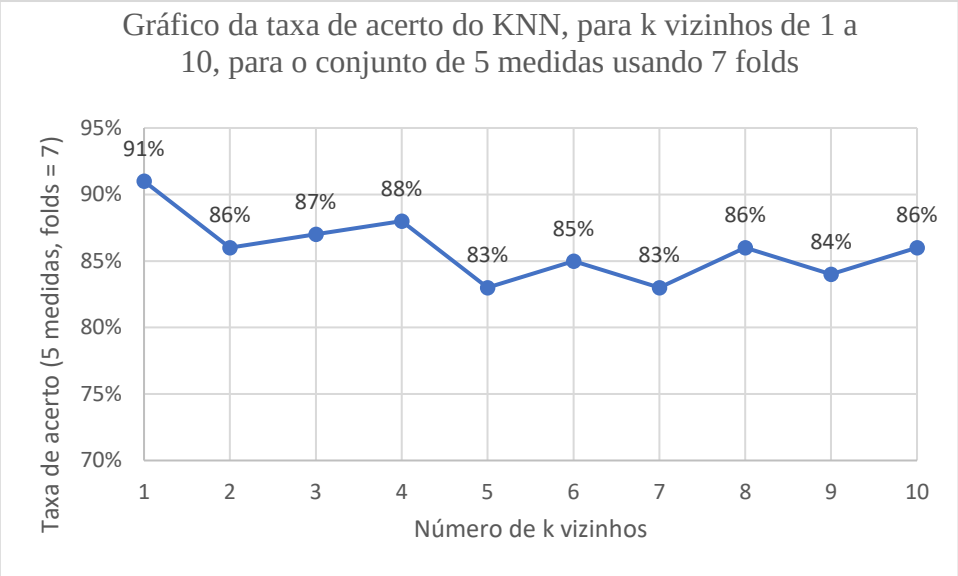


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 7 folds
Taxa de acerto (5 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
91%
86%
87%
88%
83%
85%
83%
86%
84%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

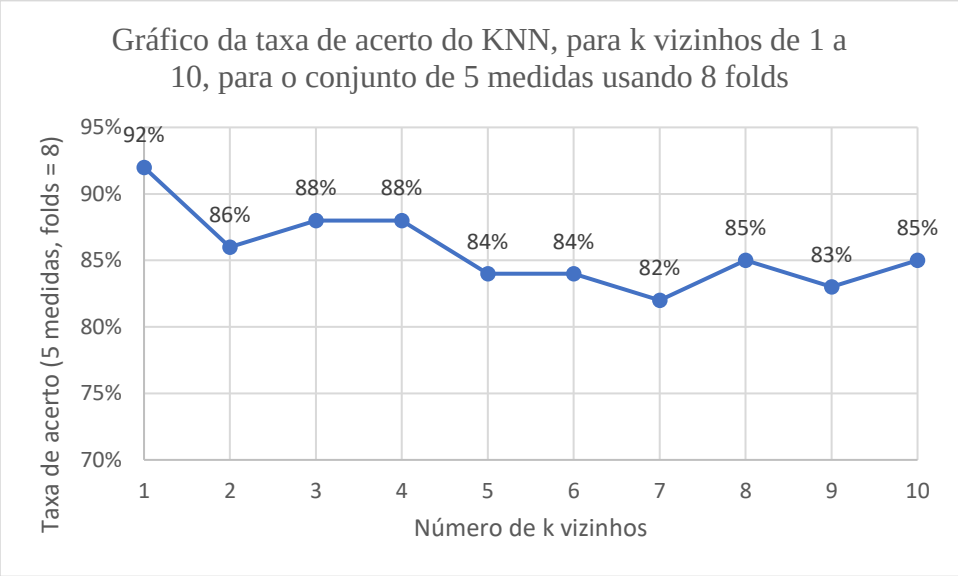


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 8 folds
Taxa de acerto (5 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
92%
86%
88%
88%
84%
84%
82%
85%
83%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

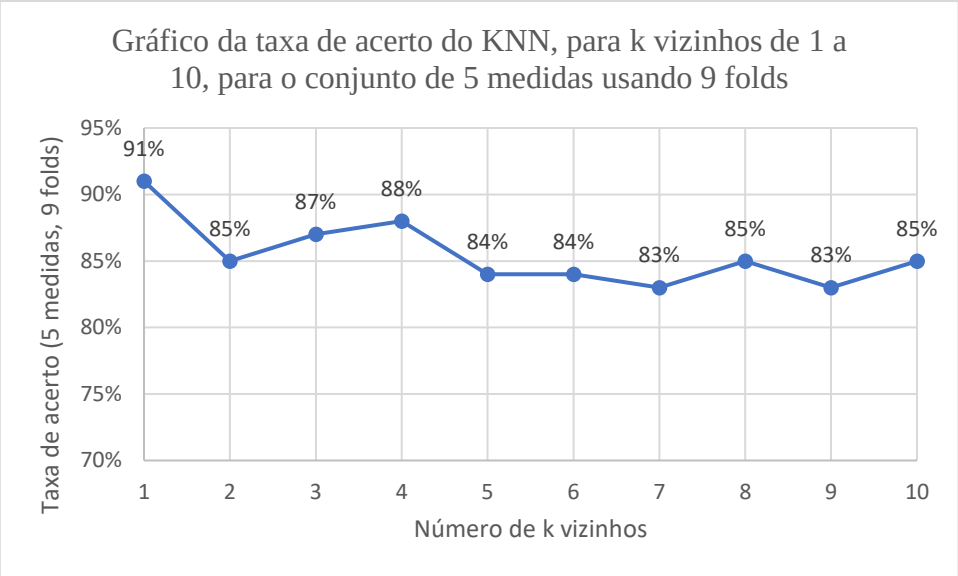


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 9 folds
Taxa de acerto (5 medidas, 9 folds)
95%
90%
85%
80%
75%
70%
91%
85%
87%
88%
84%
84%
83%
85%
83%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

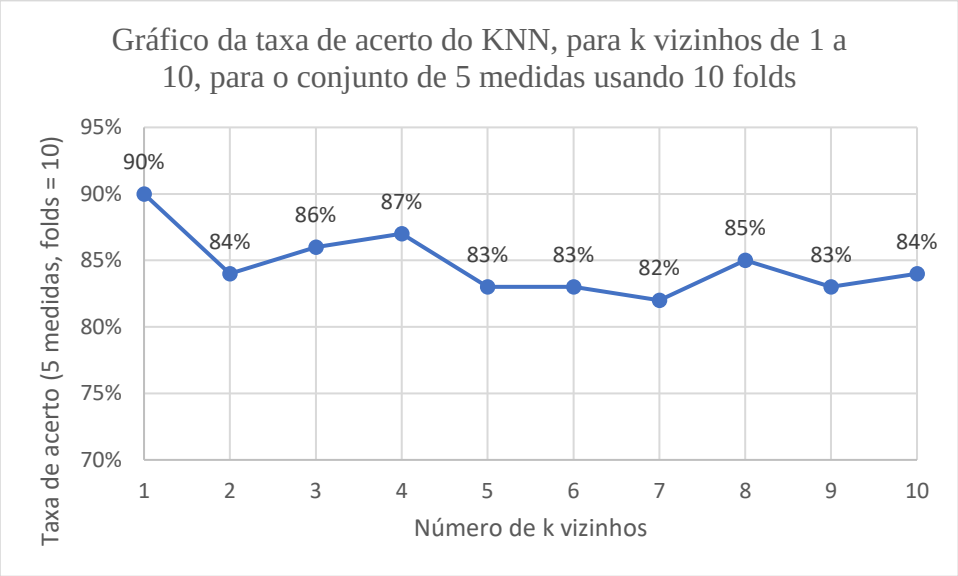


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 5 medidas usando 10 folds
Taxa de acerto (5 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
90%
84%
86%
87%
83%
83%
82%
85%
83%
84%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

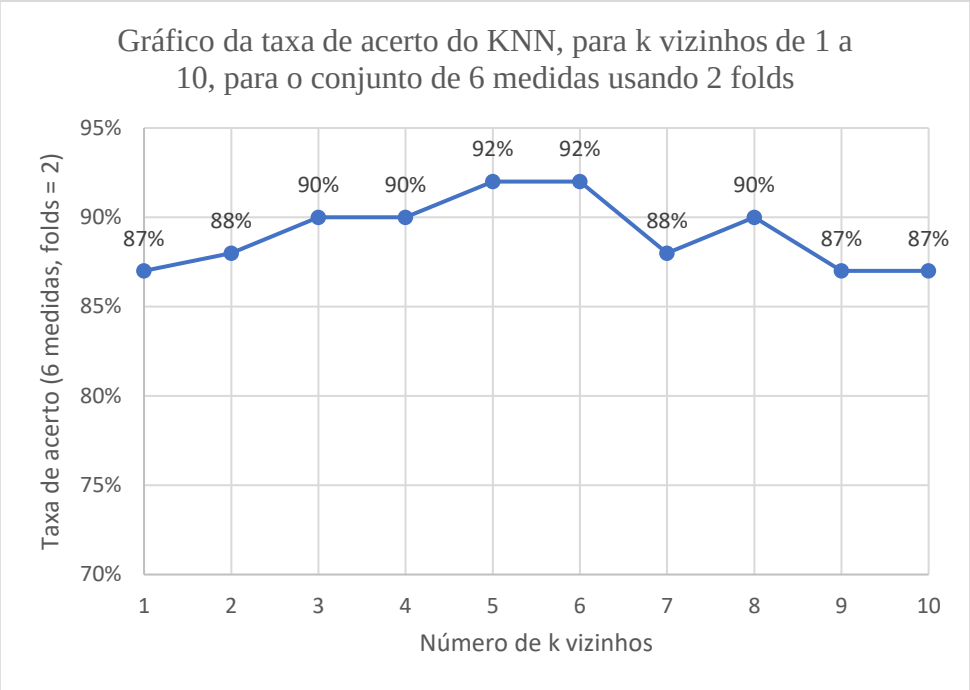


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 2 folds
Taxa de acerto (6 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
87%
88%
90%
90%
92%
92%
88%
90%
87%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

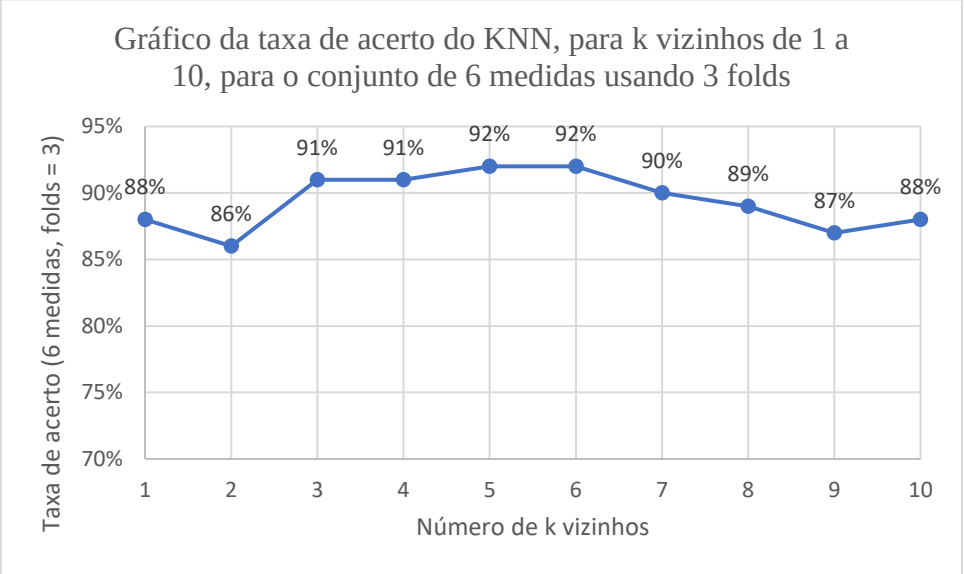


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 3 folds
Taxa de acerto (6 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
88%
86%
91%
91%
92%
92%
90%
89%
87%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

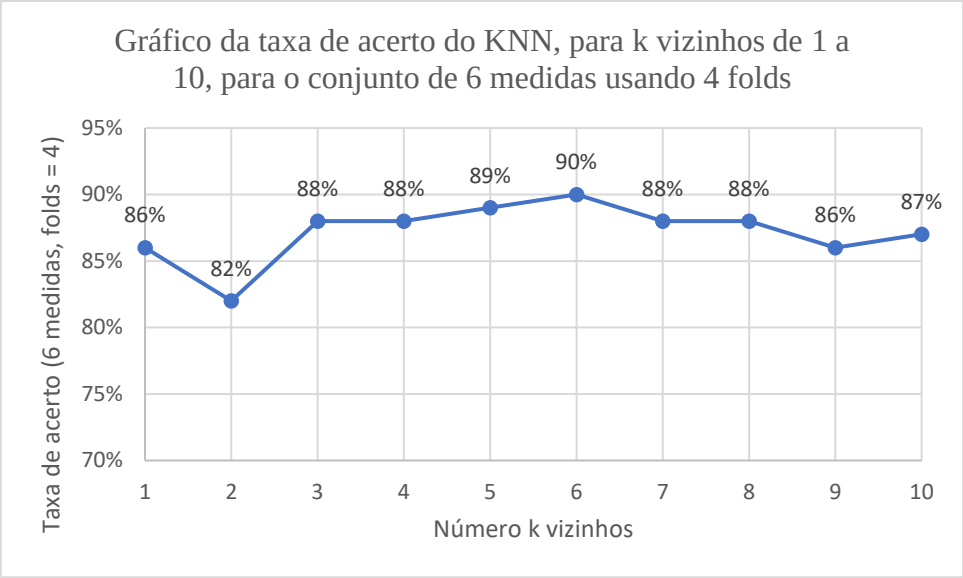


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 4 folds
Taxa de acerto (6 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
86%
82%
88%
88%
89%
90%
88%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número k vizinhos

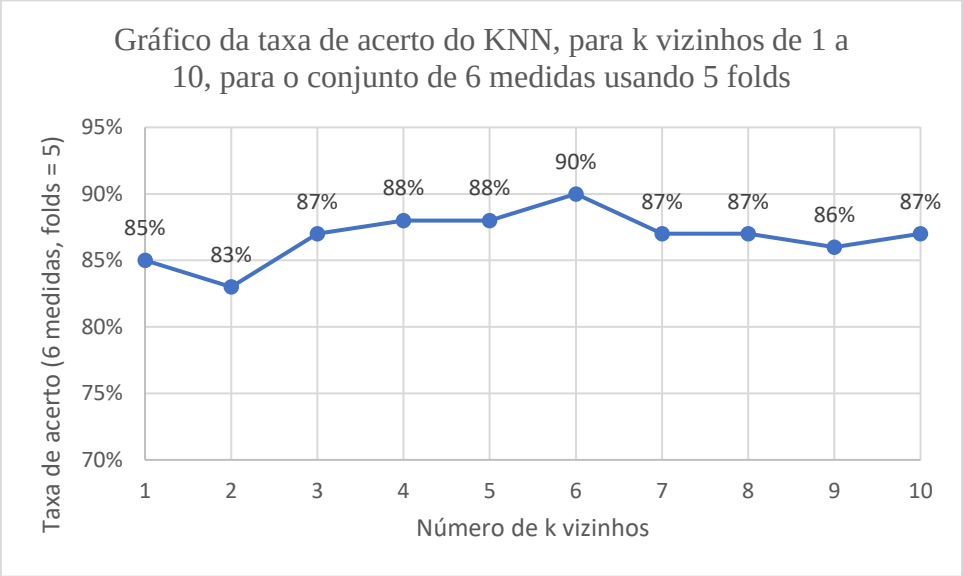


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 5 folds
Taxa de acerto (6 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
85%
83%
87%
88%
88%
90%
87%
87%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

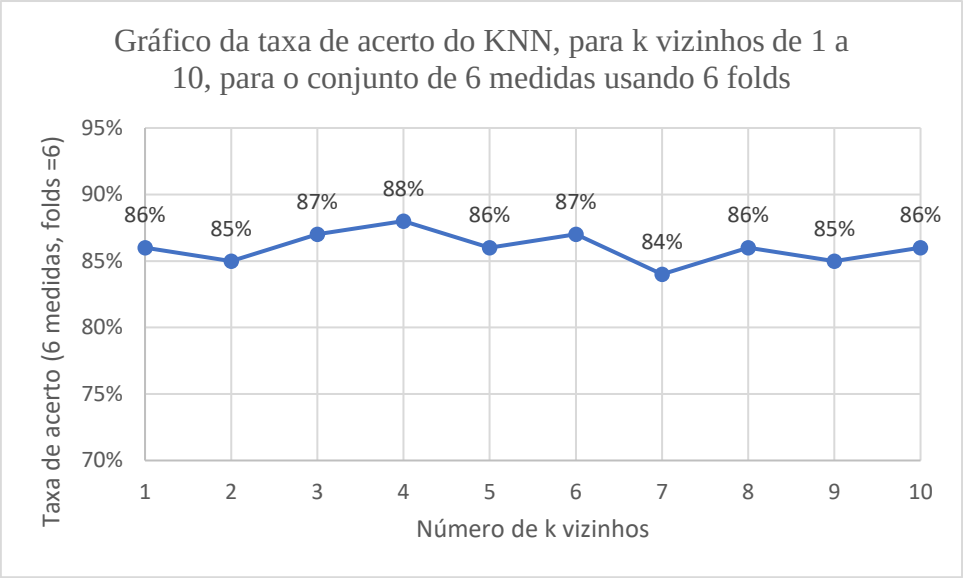


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 6 folds
Taxa de acerto (6 medidas, folds =6)
95%
90%
85%
80%
75%
70%
86%
85%
87%
88%
86%
87%
84%
86%
85%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

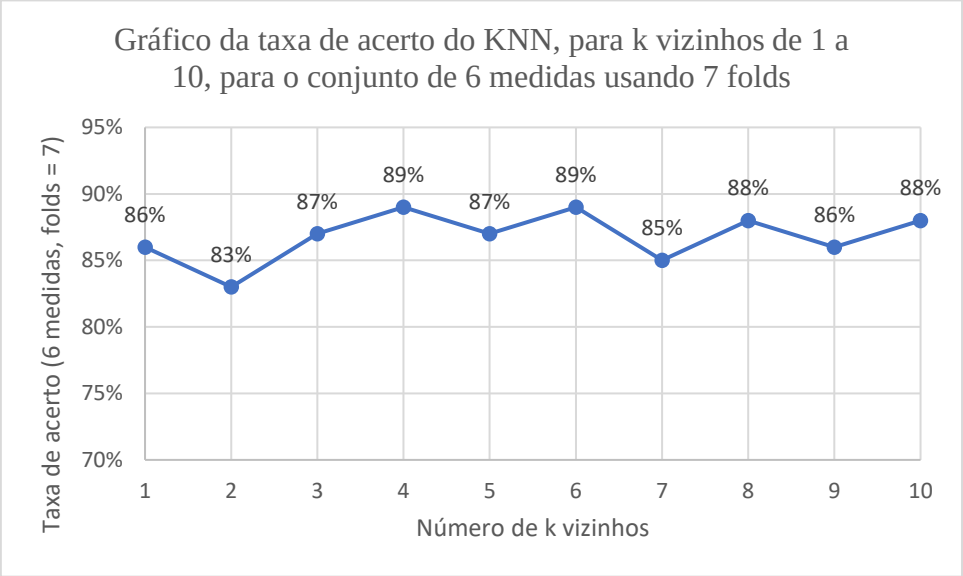


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 7 folds
Taxa de acerto (6 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
86%
83%
87%
89%
87%
89%
85%
88%
86%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

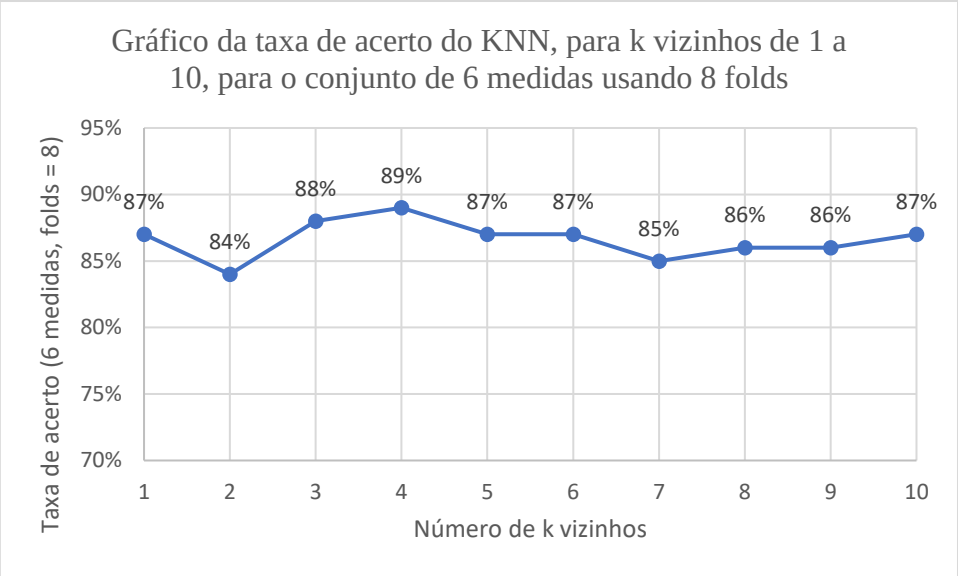


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 8 folds
Taxa de acerto (6 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
87%
84%
88%
89%
87%
87%
85%
86%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

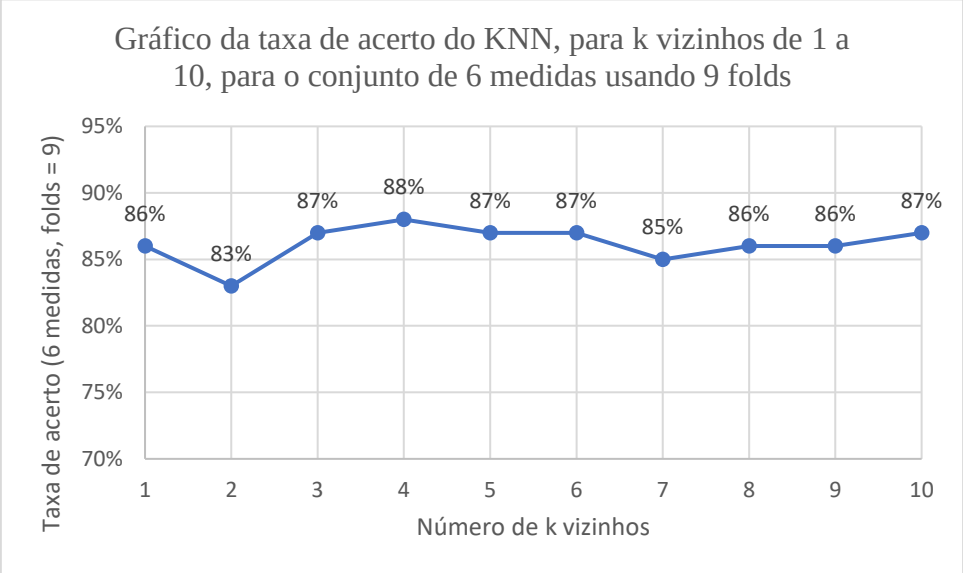


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 6 medidas usando 9 folds
Taxa de acerto (6 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
86%
83%
87%
88%
87%
87%
85%
86%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

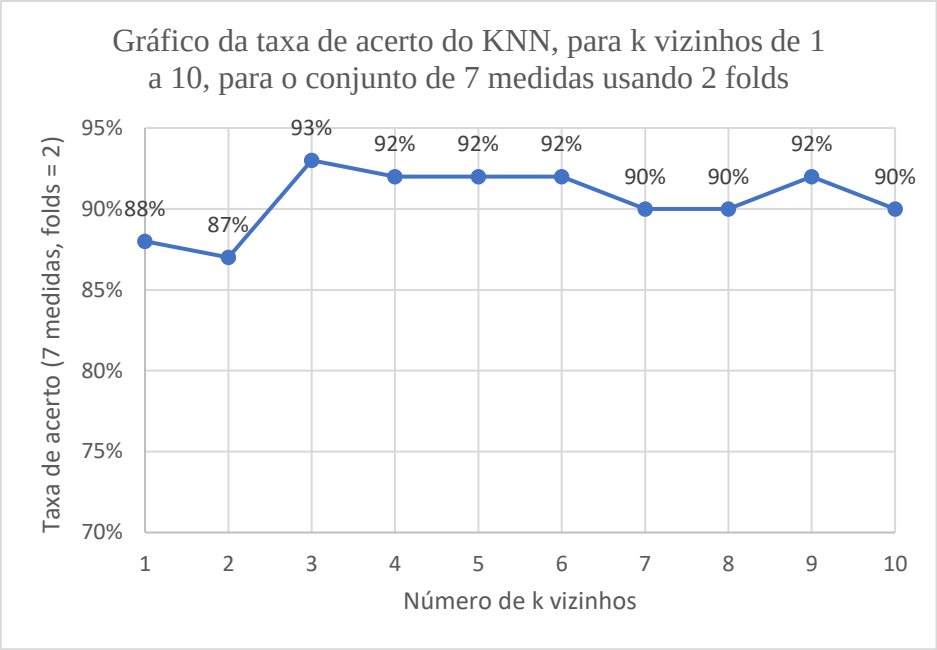


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 2 folds
Taxa de acerto (7 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
88%
87%
93%
92%
92%
92%
90%
90%
92%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

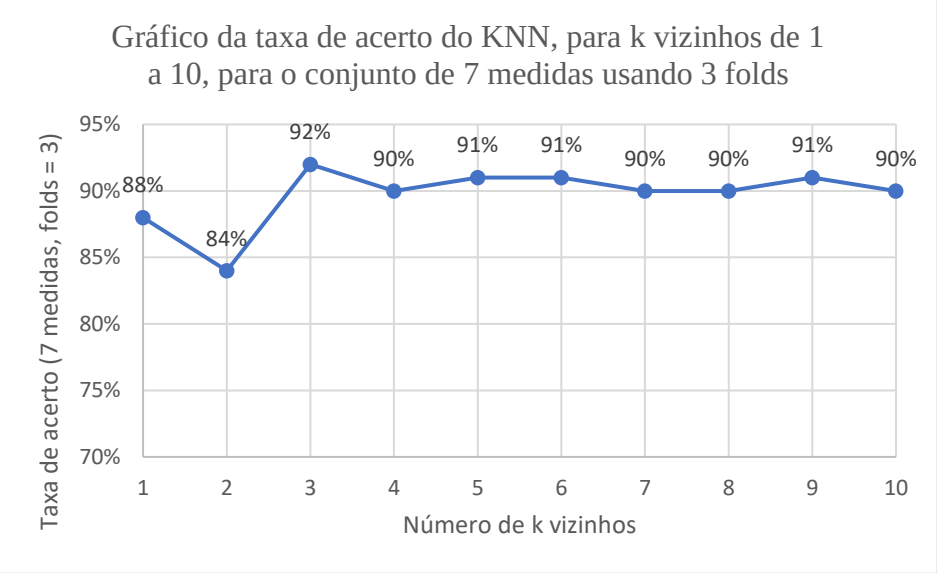


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 3 folds
Taxa de acerto (7 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
88%
84%
92%
90%
91%
91%
90%
90%
91%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

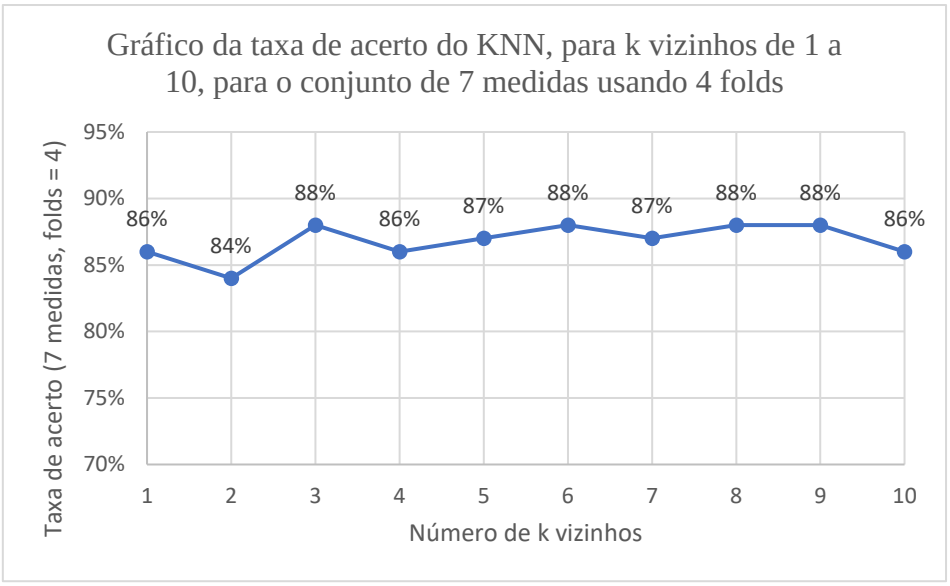


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 4 folds
Taxa de acerto (7 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
86%
84%
88%
86%
87%
88%
87%
88%
88%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

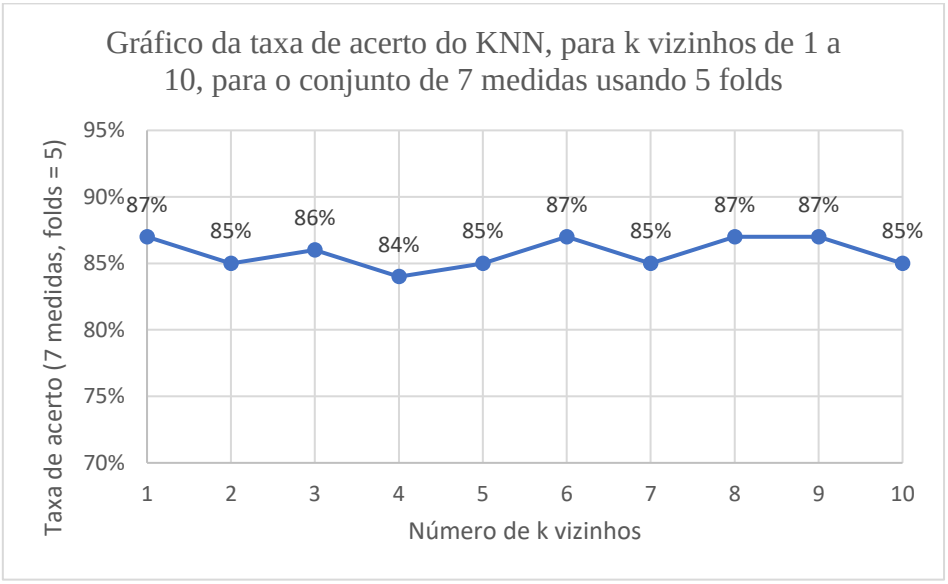


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 5 folds
Taxa de acerto (7 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
87%
85%
86%
84%
85%
87%
85%
87%
87%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

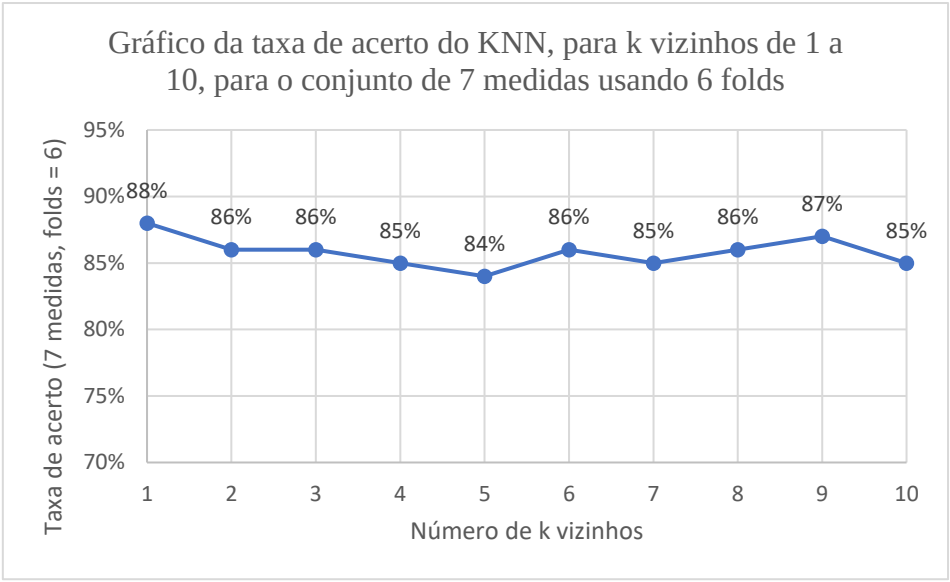


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 6 folds
Taxa de acerto (7 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
88%
86%
86%
85%
84%
86%
85%
86%
87%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

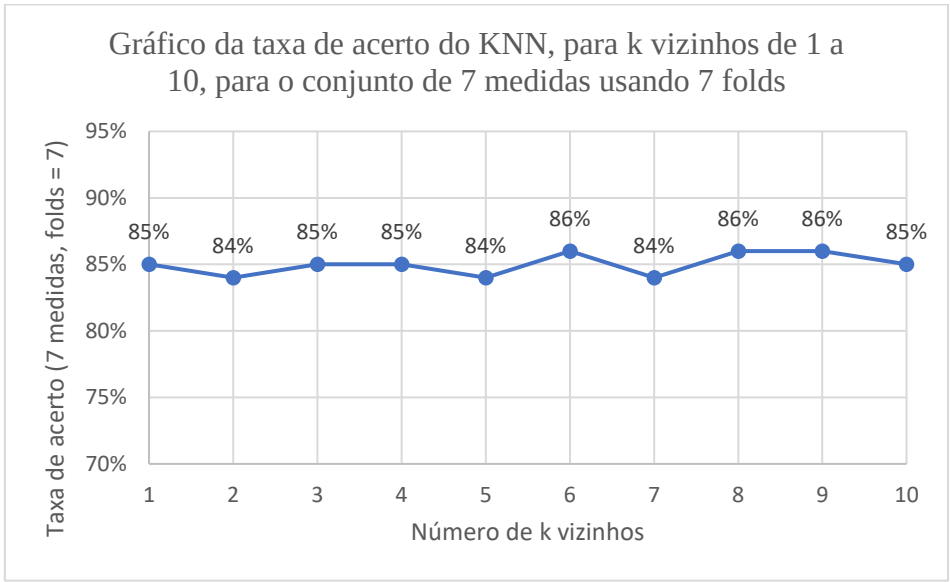


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 7 folds
Taxa de acerto (7 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
85%
84%
85%
85%
84%
86%
84%
86%
86%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

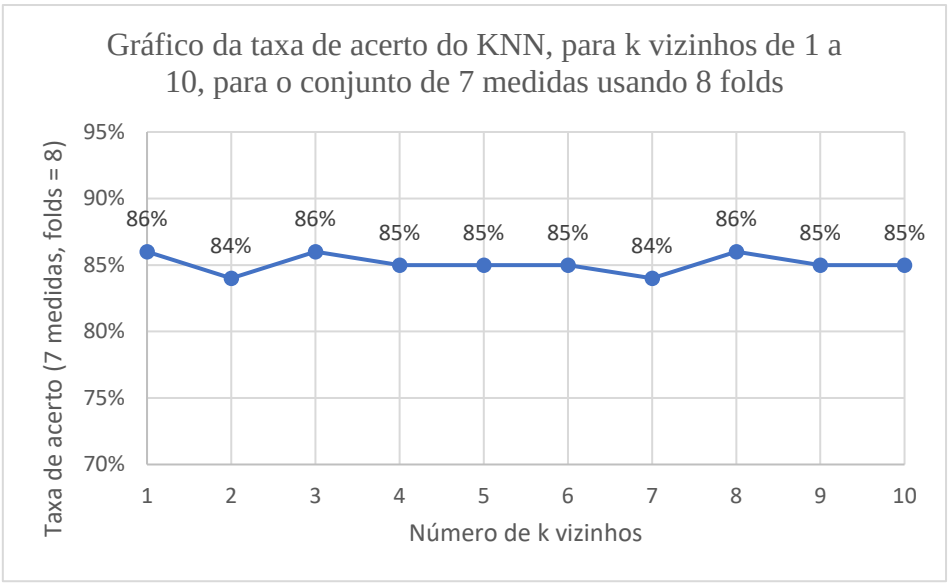


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 8 folds
Taxa de acerto (7 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
86%
84%
86%
85%
85%
85%
84%
86%
85%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

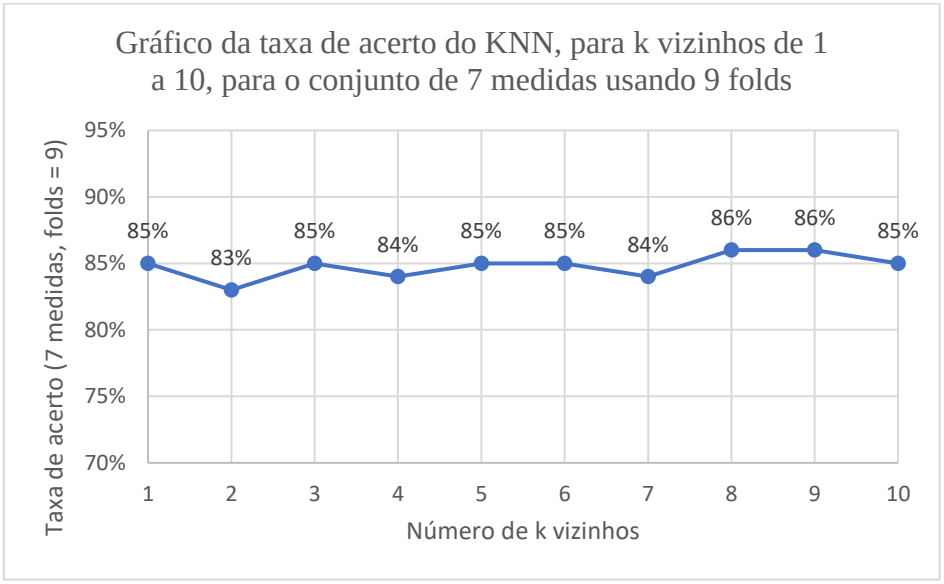


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 9 folds
Taxa de acerto (7 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
85%
83%
85%
84%
85%
85%
84%
86%
86%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

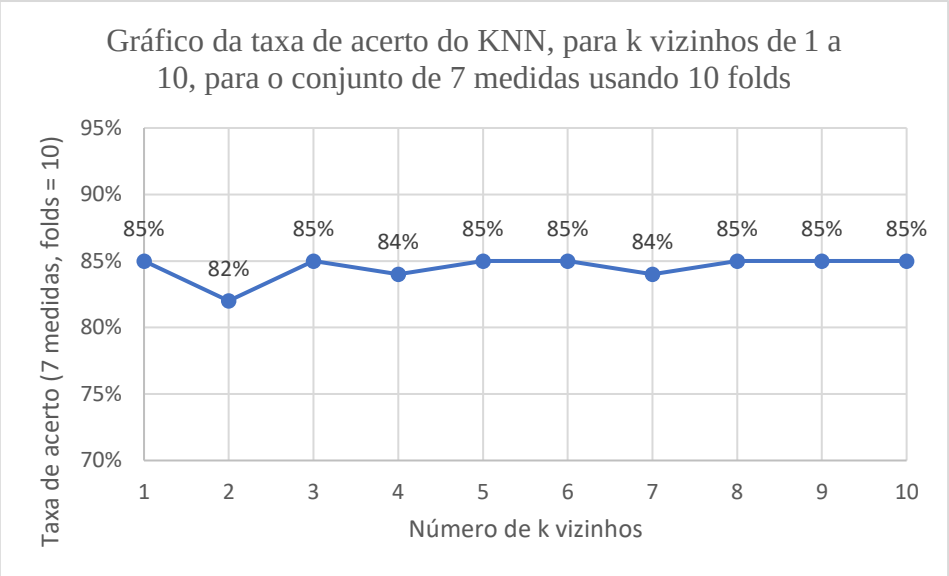


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 7 medidas usando 10 folds
Taxa de acerto (7 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
85%
82%
85%
84%
85%
85%
84%
85%
85%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

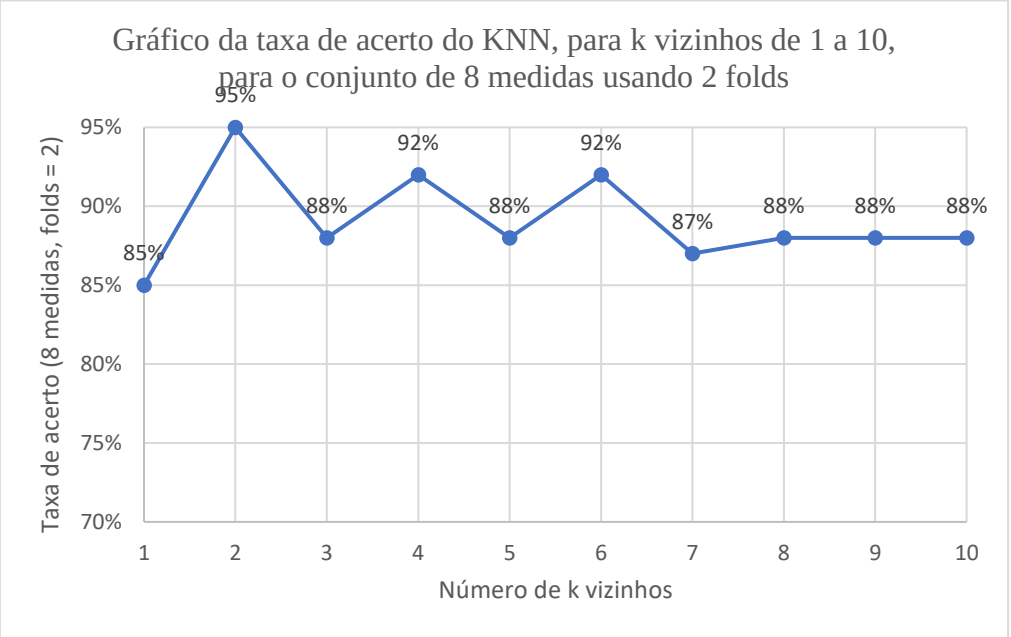


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 2 folds
Taxa de acerto (8 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
85%
95%
88%
92%
88%
92%
87%
88%
88%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

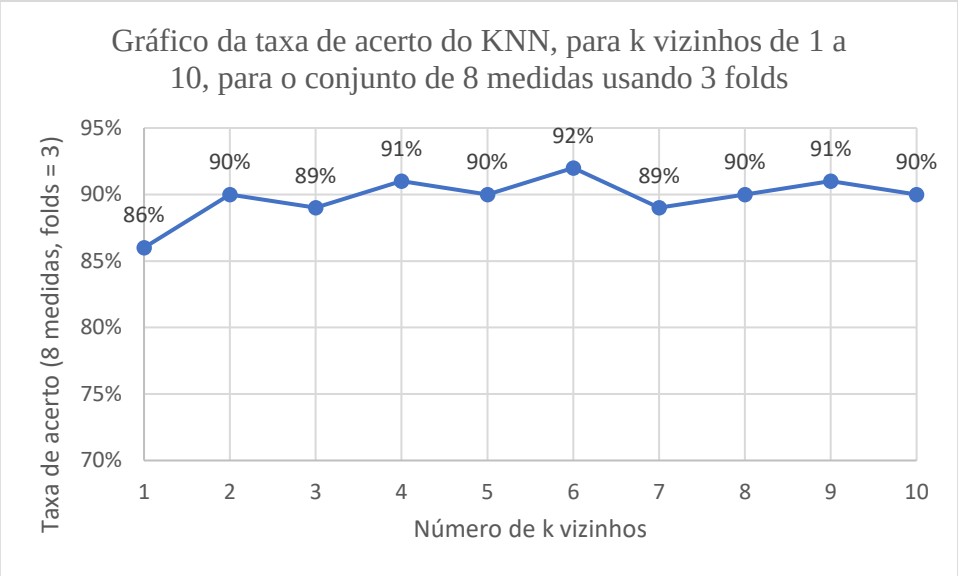


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 3 folds
Taxa de acerto (8 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
86%
90%
89%
91%
90%
92%
89%
90%
91%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

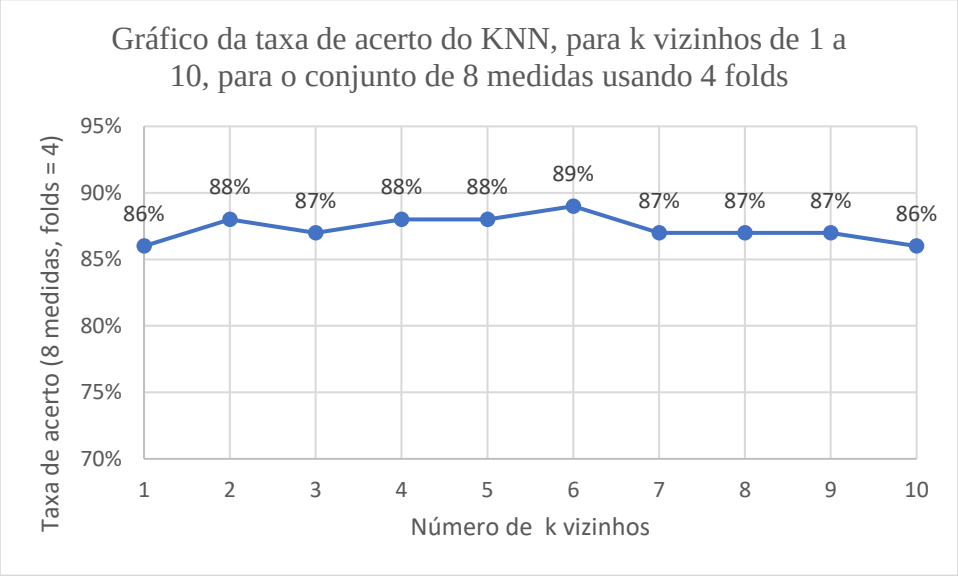


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 4 folds
Taxa de acerto (8 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
86%
88%
87%
88%
88%
89%
87%
87%
87%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

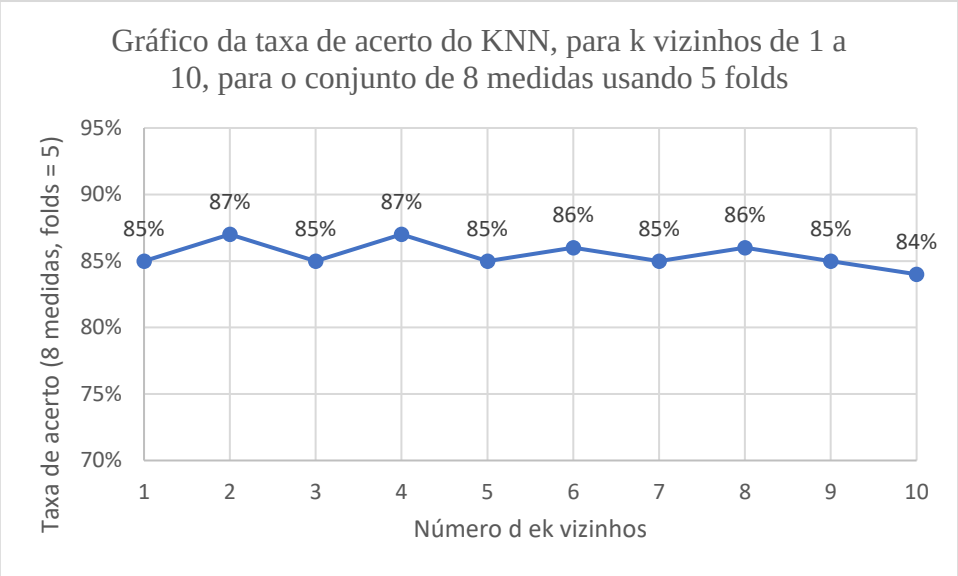


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 5 folds
Taxa de acerto (8 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
85%
87%
85%
87%
85%
86%
85%
86%
85%
84%
1
2
3
4
5
6
7
8
9
10
Número d ek vizinhos

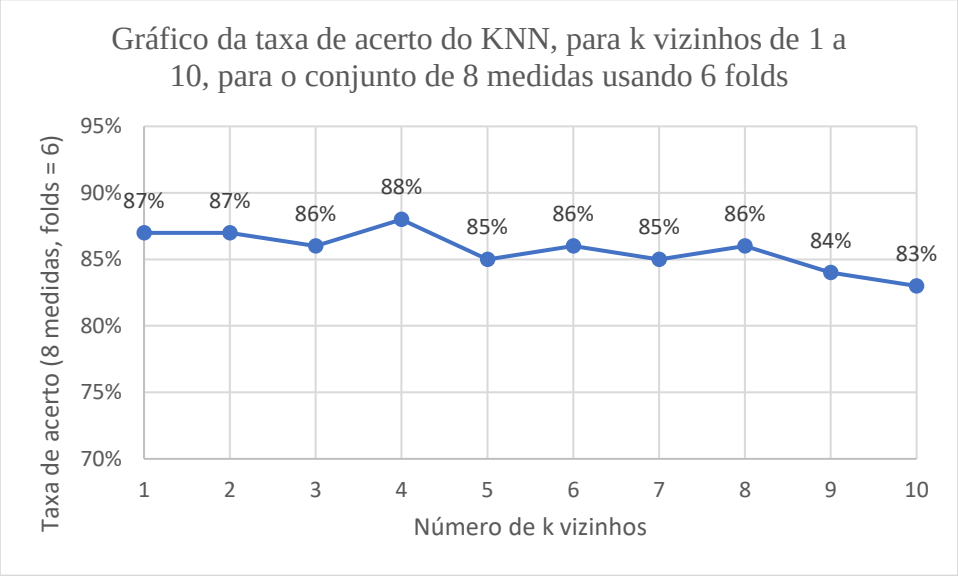


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 6 folds
Taxa de acerto (8 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
87%
87%
86%
88%
85%
86%
85%
86%
84%
83%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

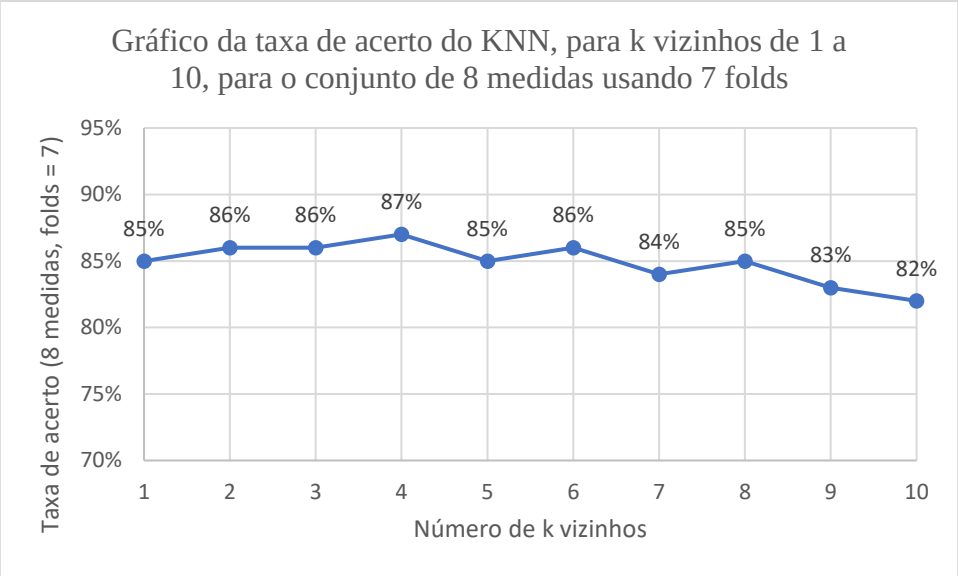


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 7 folds
Taxa de acerto (8 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
85%
86%
86%
87%
85%
86%
84%
85%
83%
82%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

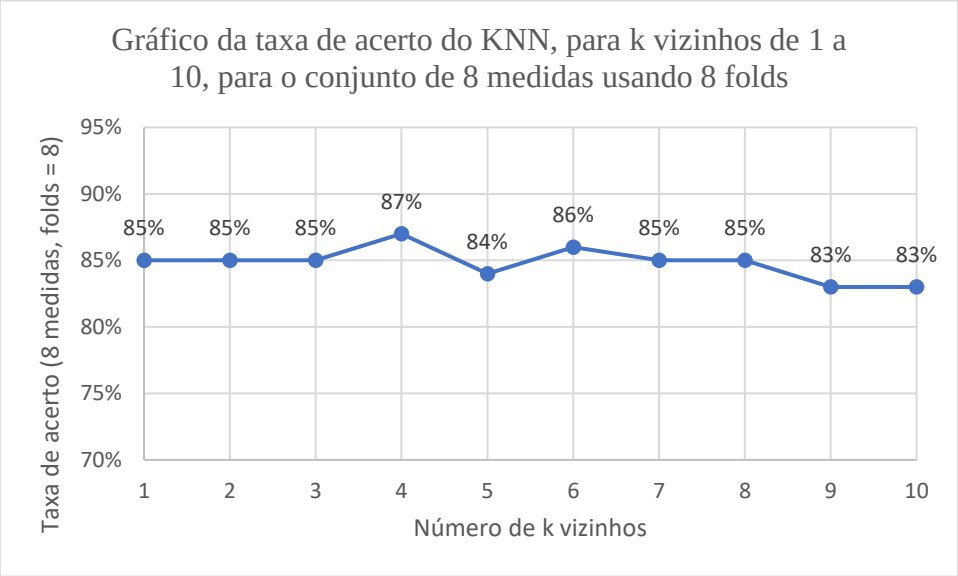


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 8 folds
Taxa de acerto (8 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
85%
85%
85%
87%
84%
86%
85%
85%
83%
83%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

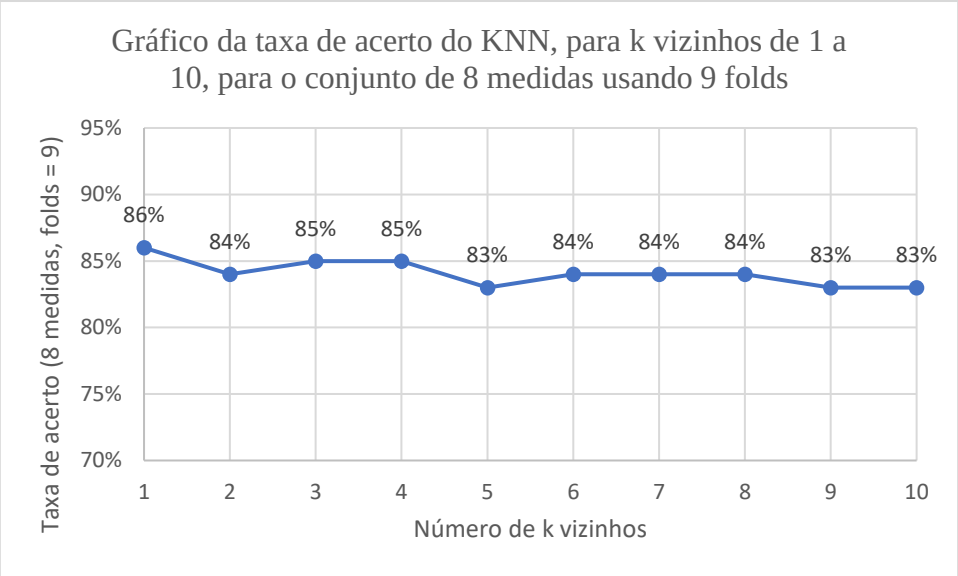


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 9 folds
Taxa de acerto (8 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
86%
84%
85%
85%
83%
84%
84%
84%
83%
83%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

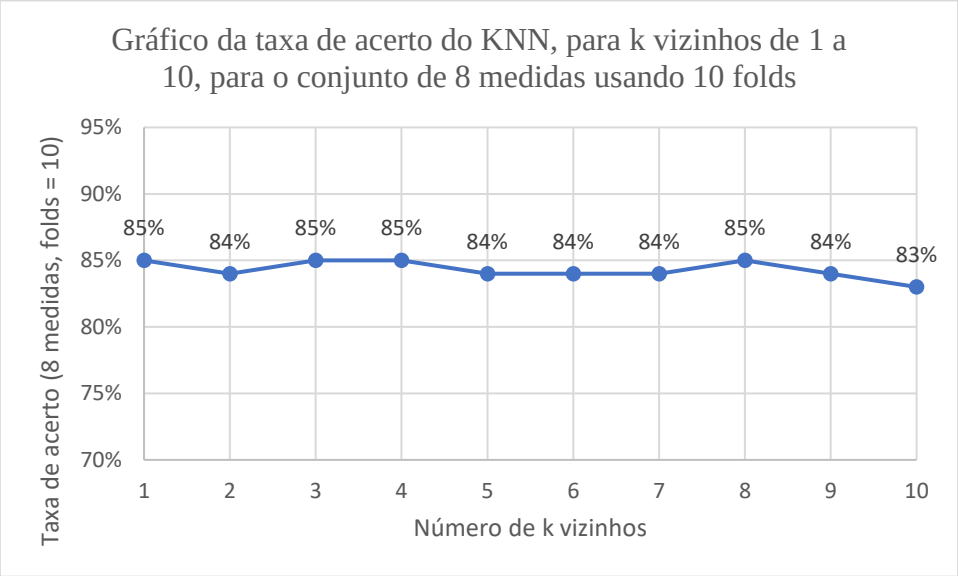


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 8 medidas usando 10 folds
Taxa de acerto (8 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
85%
84%
85%
85%
84%
84%
84%
85%
84%
83%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

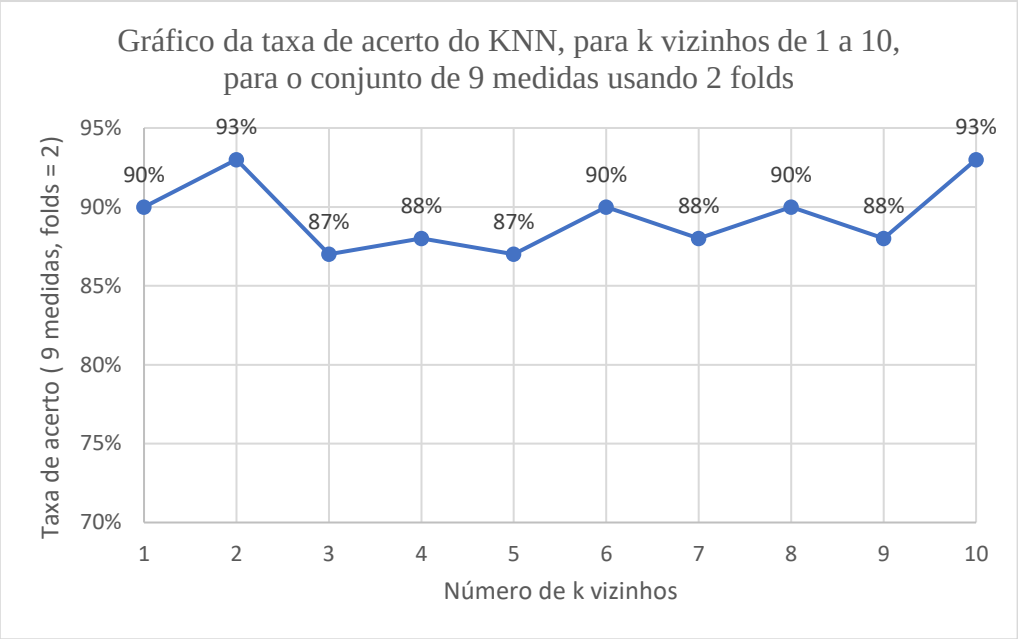


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 2 folds
Taxa de acerto (9 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
90%
93%
87%
88%
87%
90%
88%
90%
88%
93%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

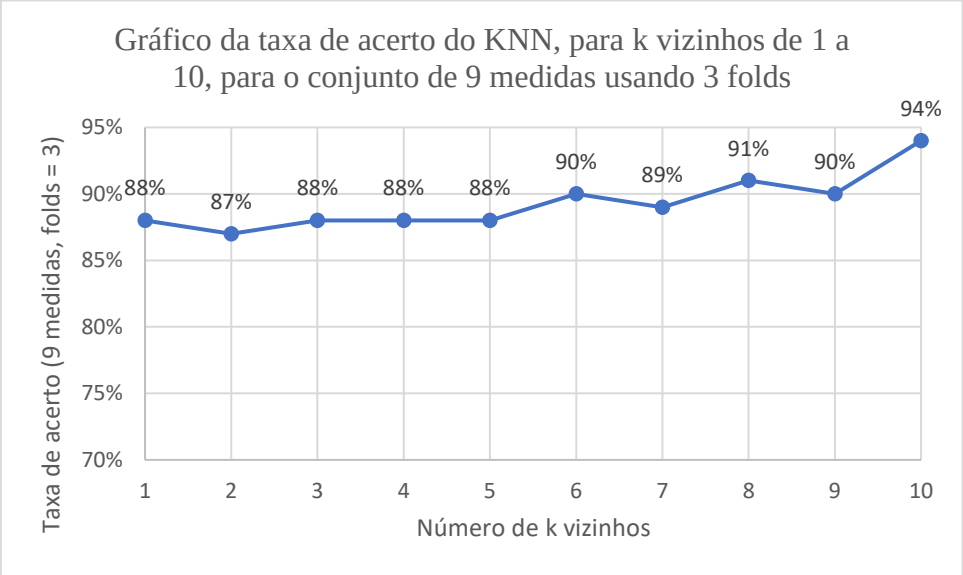


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 3 folds
Taxa de acerto (9 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
88%
87%
88%
88%
88%
90%
89%
91%
90%
94%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

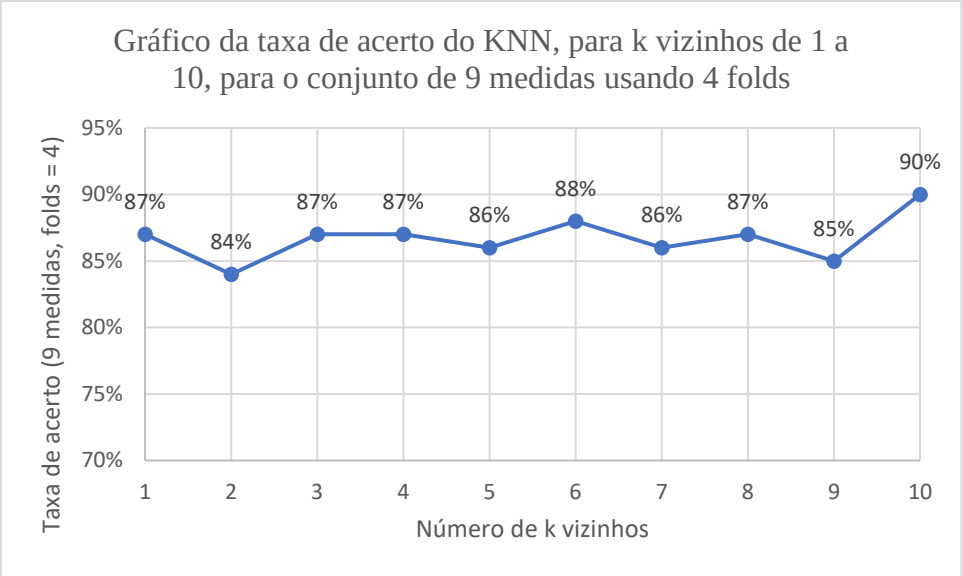


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 4 folds
Taxa de acerto (9 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
87%
84%
87%
87%
86%
88%
86%
87%
85%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

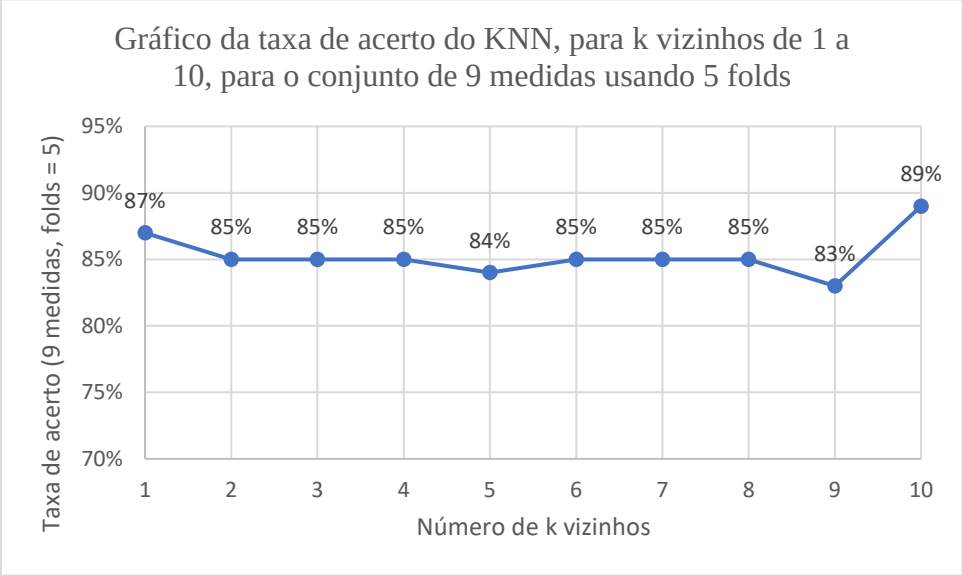


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 5 folds
Taxa de acerto (9 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
87%
85%
85%
85%
84%
85%
85%
85%
83%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

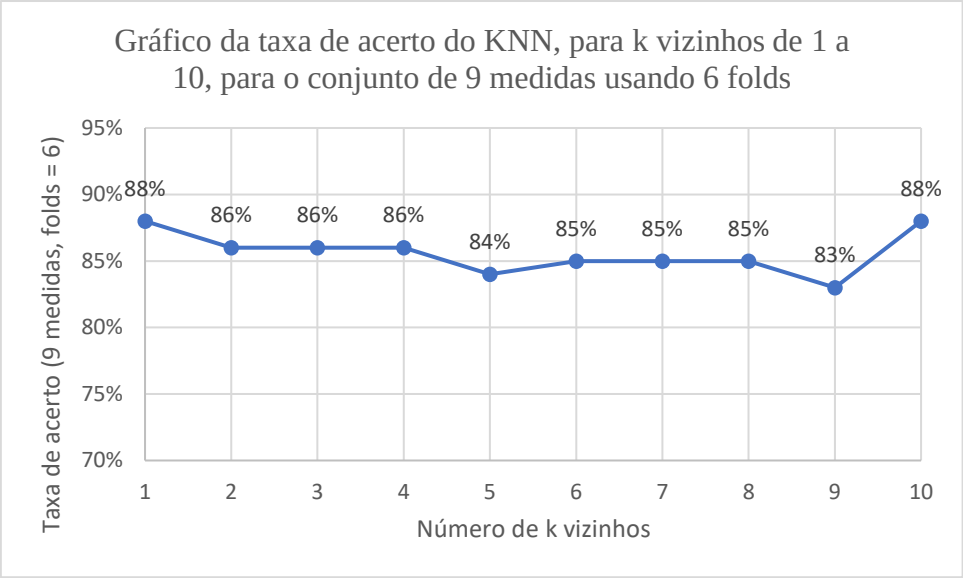


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 6 folds
Taxa de acerto (9 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
88%
86%
86%
86%
84%
85%
85%
85%
83%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

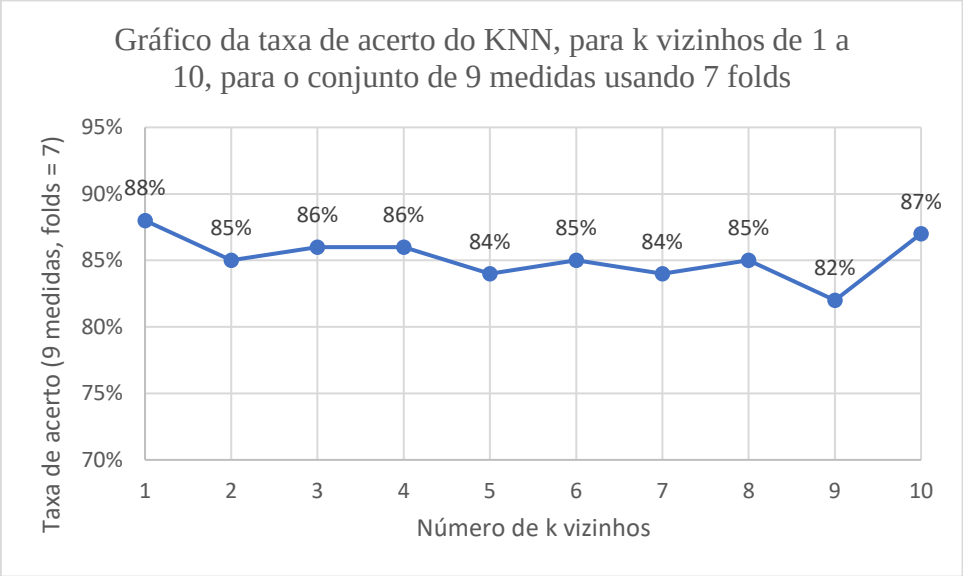


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 7 folds
Taxa de acerto (9 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
88%
85%
86%
86%
84%
85%
84%
85%
82%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

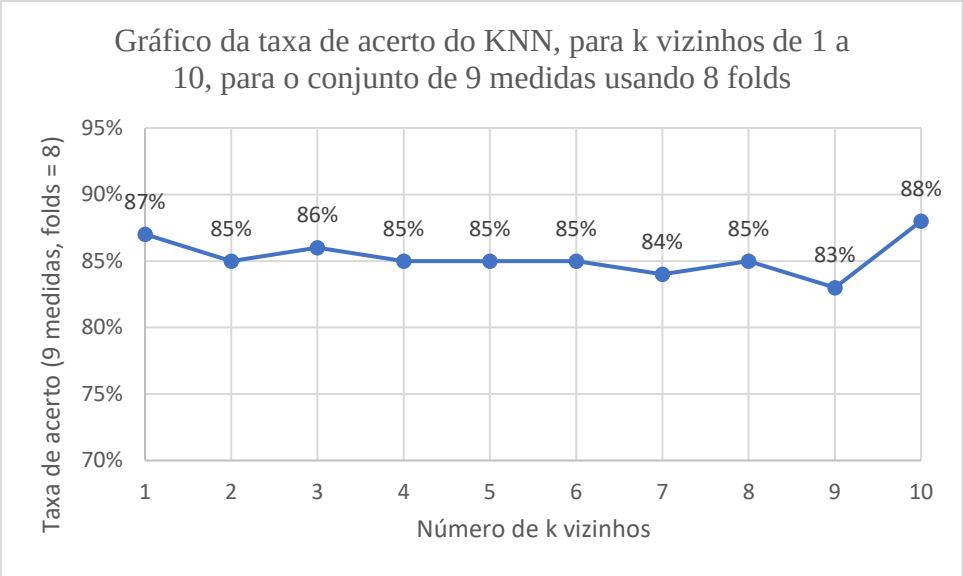


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 8 folds
Taxa de acerto (9 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
87%
85%
86%
85%
85%
85%
84%
85%
83%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

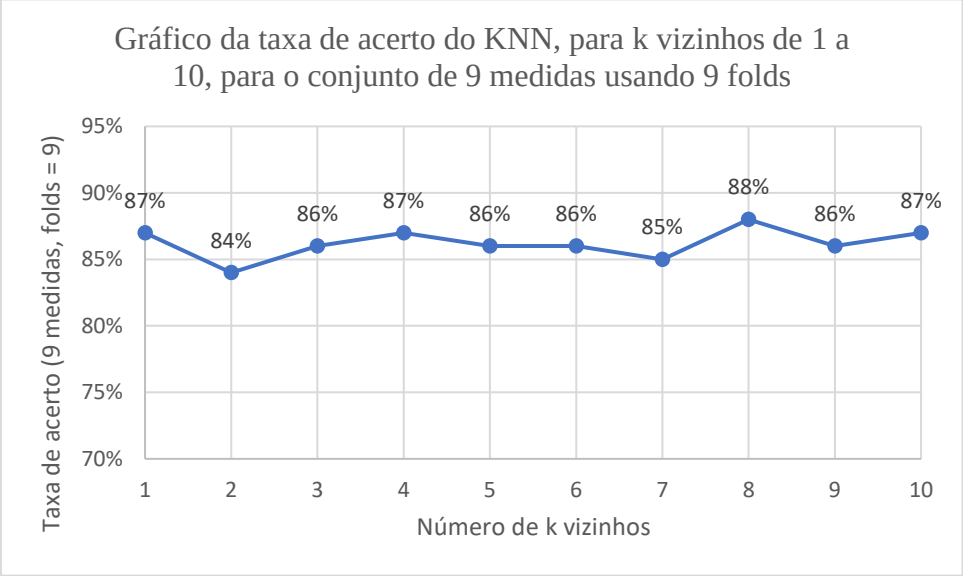


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 9 folds
Taxa de acerto (9 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
87%
84%
86%
87%
86%
86%
85%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

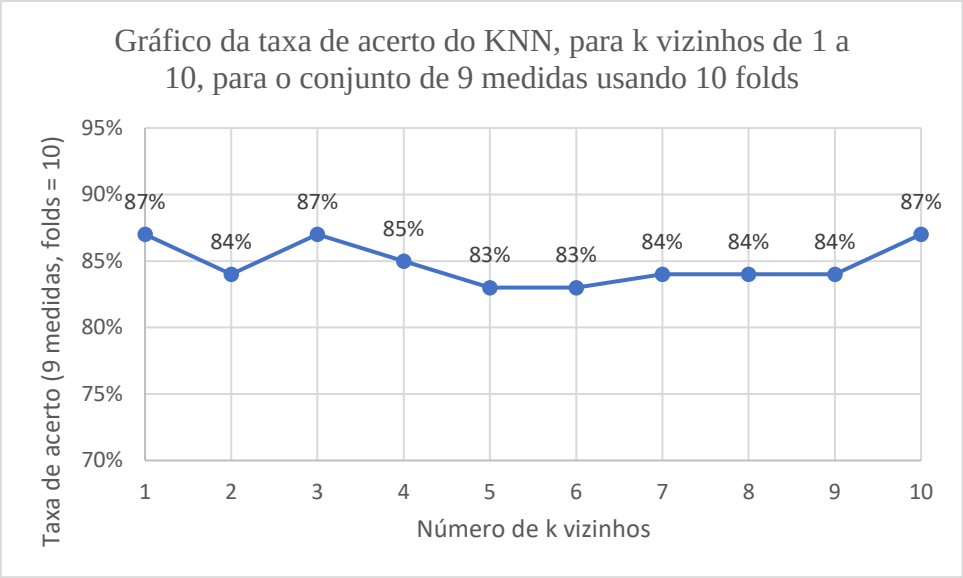


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 9 medidas usando 10 folds
Taxa de acerto (9 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
87%
84%
87%
85%
83%
83%
84%
84%
84%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

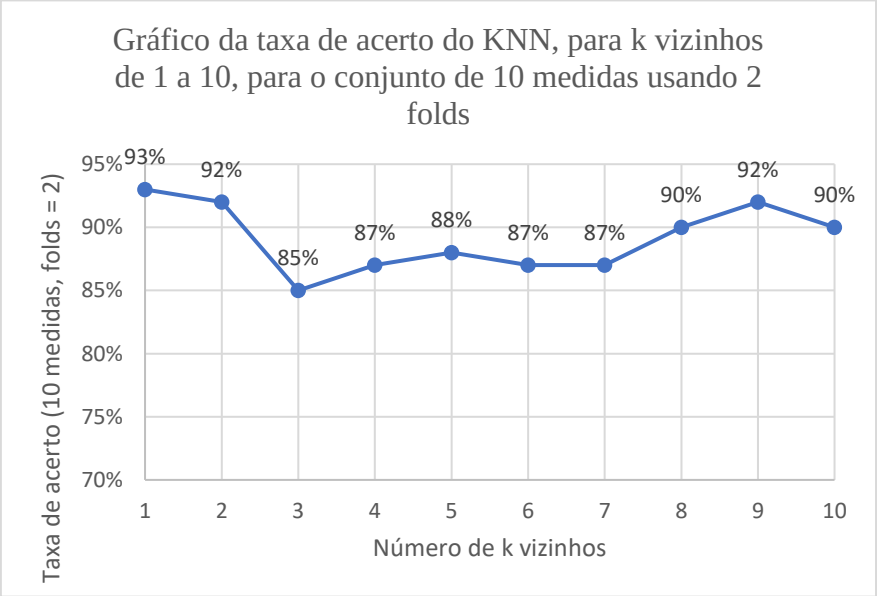


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 2 folds
Taxa de acerto (10 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
93%
92%
85%
87%
88%
87%
87%
90%
92%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

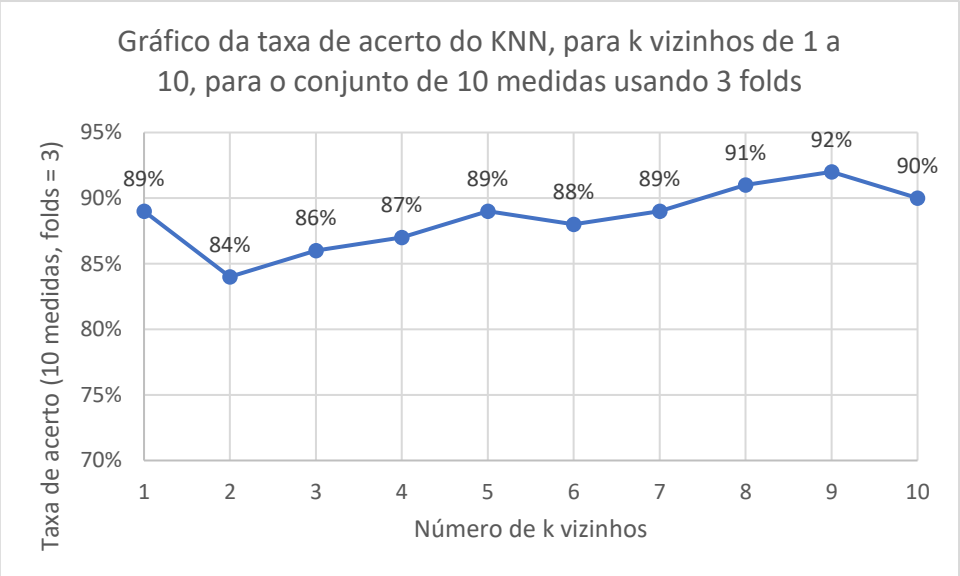


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 3 folds
Taxa de acerto (10 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
89%
84%
86%
87%
89%
88%
89%
91%
92%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

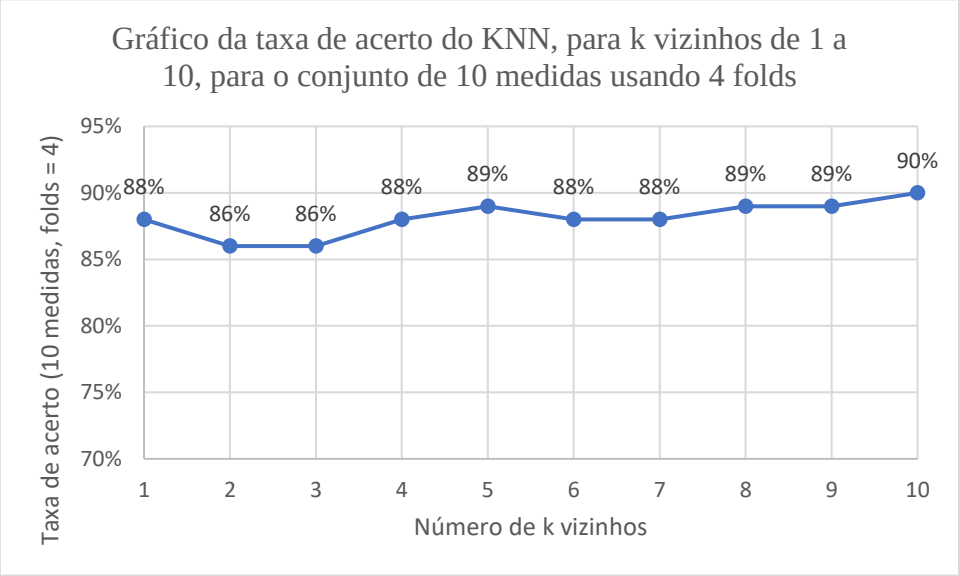


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 4 folds
Taxa de acerto (10 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
88%
86%
86%
88%
89%
88%
88%
89%
89%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

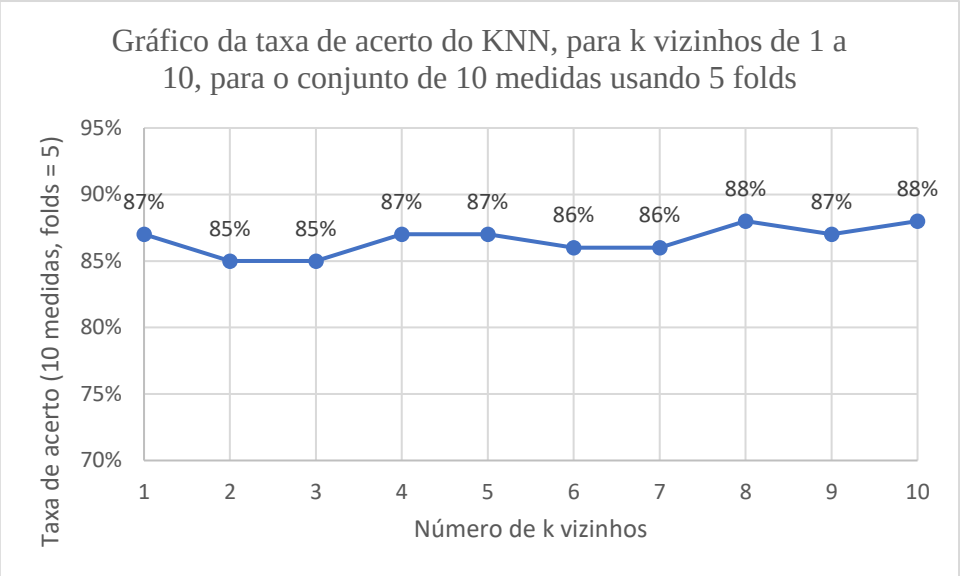


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 5 folds
Taxa de acerto (10 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
87%
85%
85%
87%
87%
86%
86%
88%
87%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

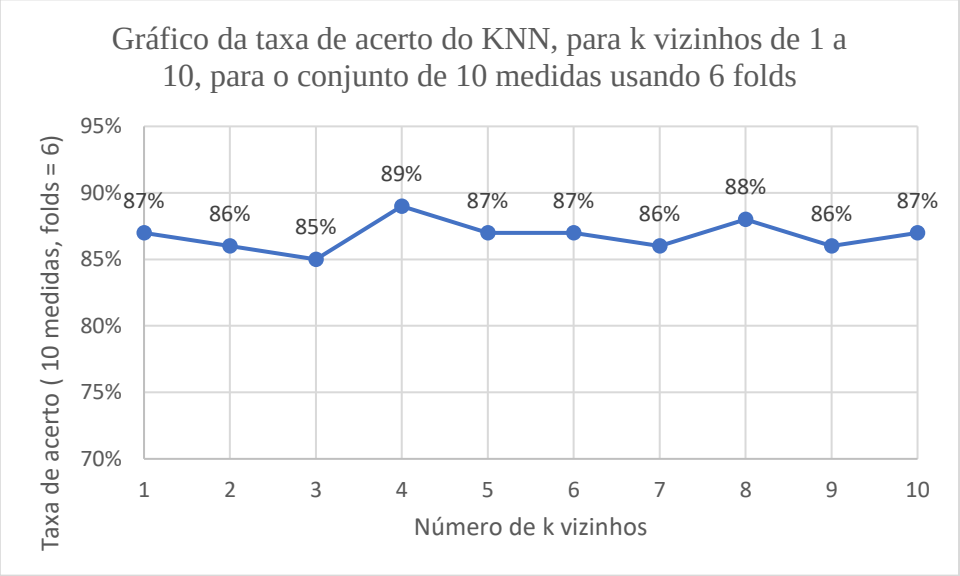


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 6 folds
Taxa de acerto (10 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
87%
86%
85%
89%
87%
87%
86%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

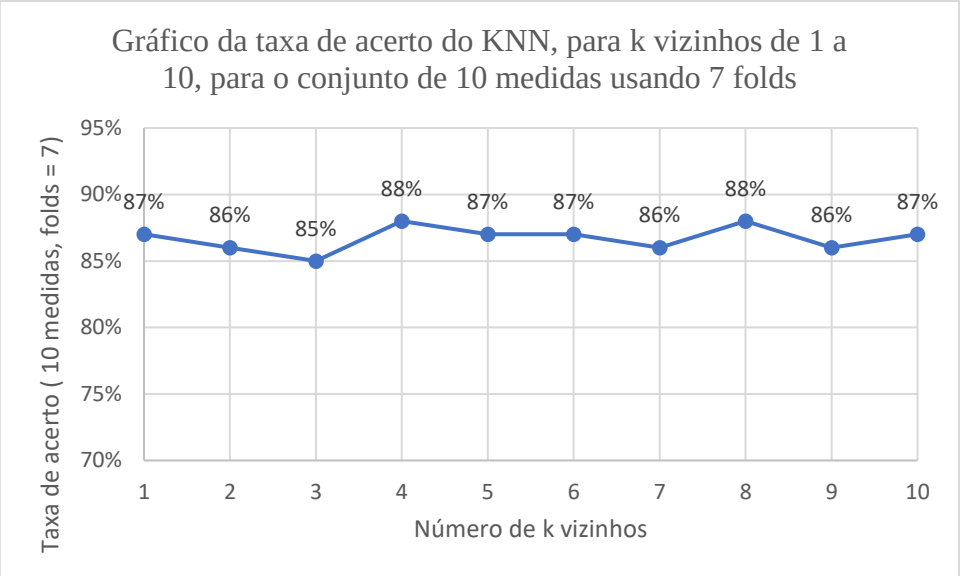


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 7 folds
Taxa de acerto (10 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
87%
86%
85%
88%
87%
87%
86%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

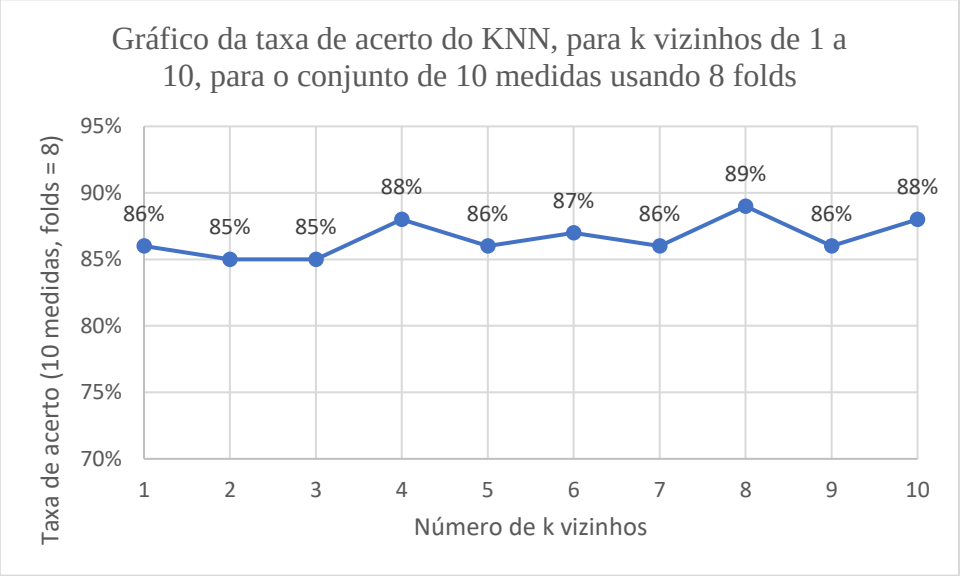


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 8 folds
Taxa de acerto (10 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
86%
85%
85%
88%
86%
87%
86%
89%
86%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

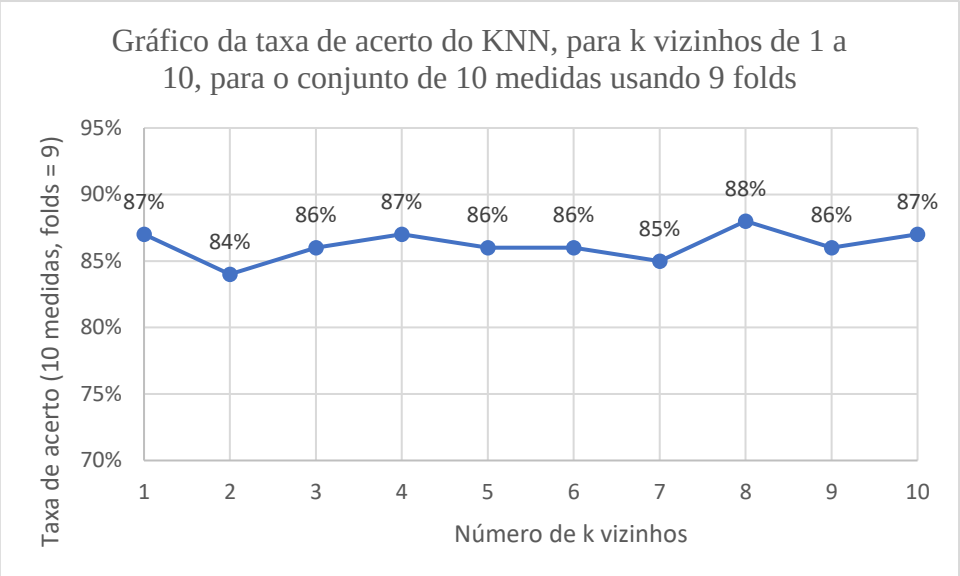


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 9 folds
Taxa de acerto (10 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
87%
84%
86%
87%
86%
86%
85%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

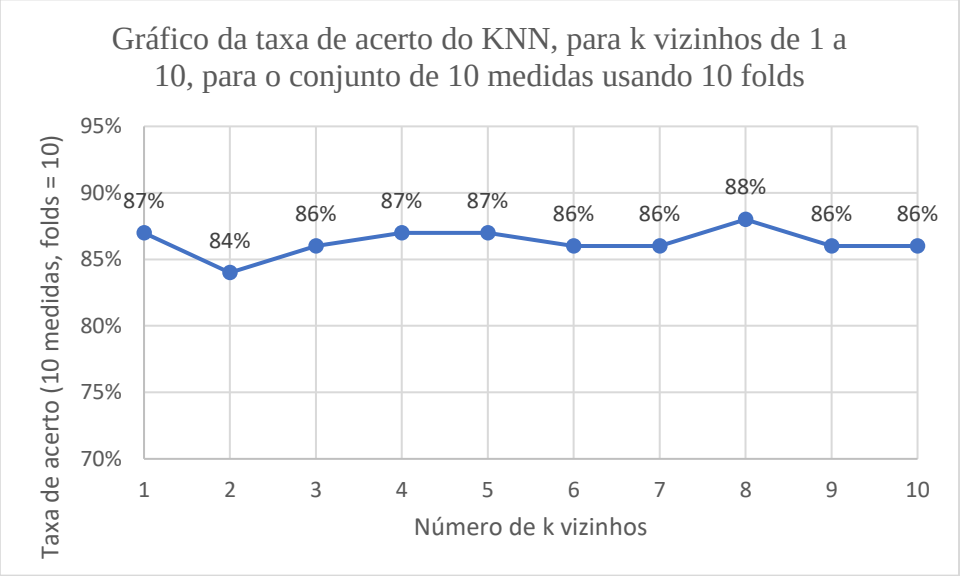


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 10 medidas usando 10 folds
Taxa de acerto (10 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
87%
84%
86%
87%
87%
86%
86%
88%
86%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

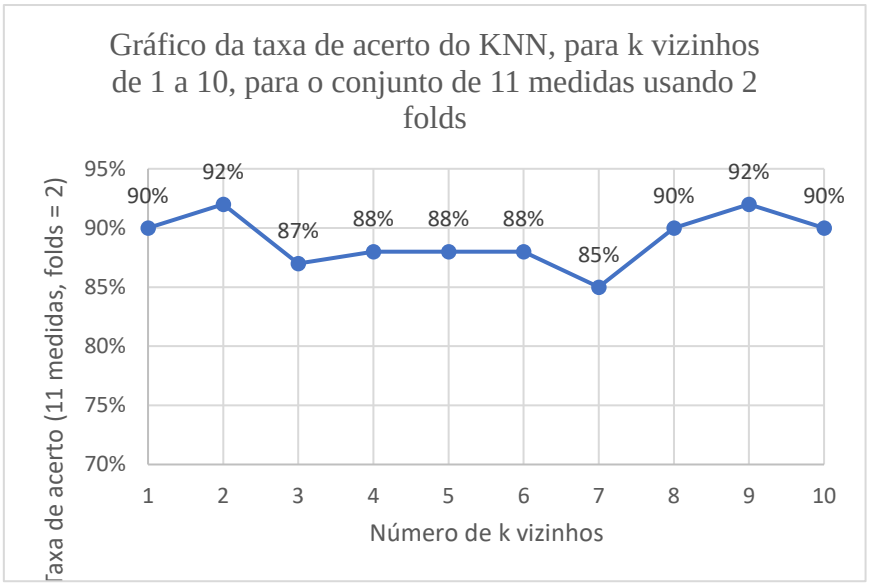


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 2 folds
Taxa de acerto (11 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
90%
92%
87%
88%
88%
88%
85%
90%
92%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

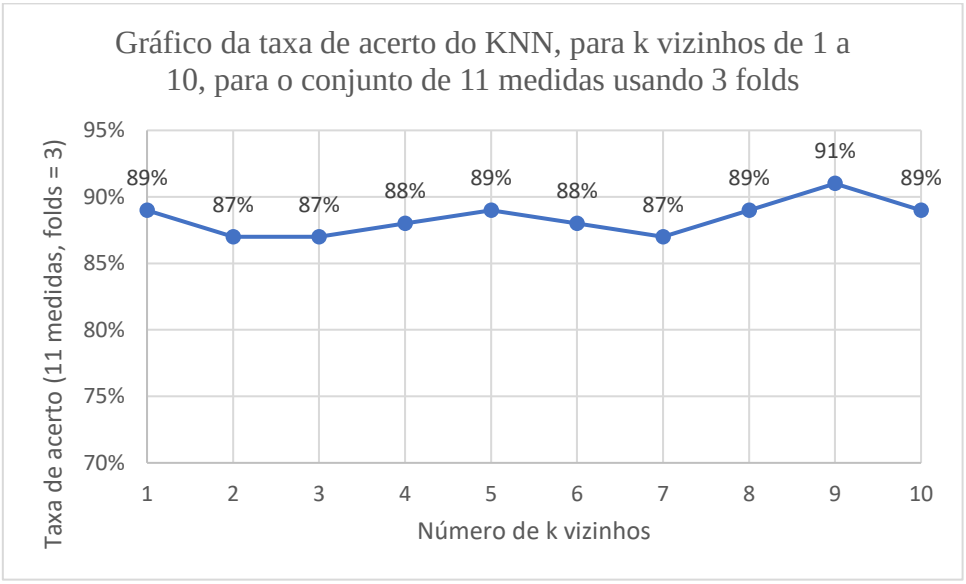


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 3 folds
Taxa de acerto (11 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
89%
87%
87%
88%
89%
88%
87%
89%
91%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

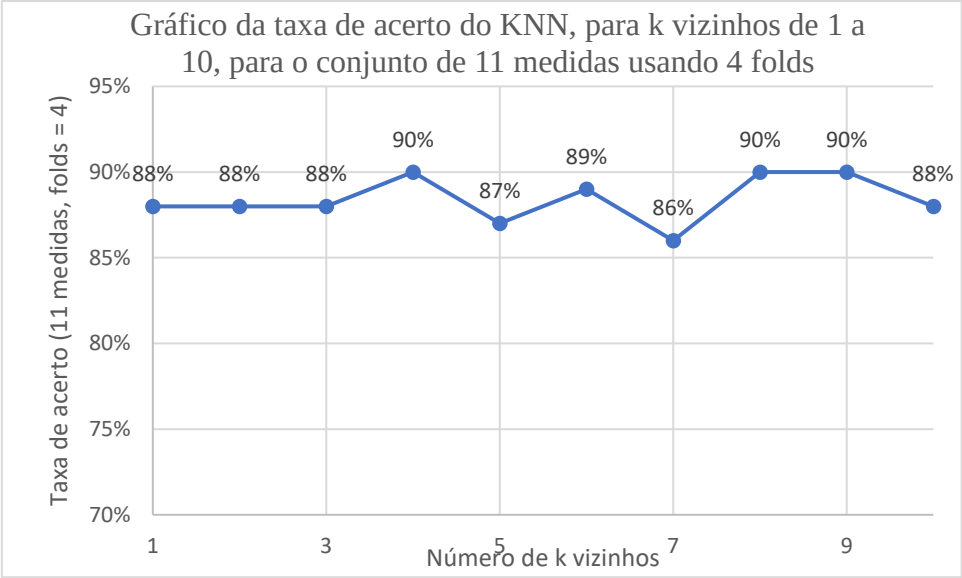


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 4 folds
Taxa de acerto (11 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
88%
88%
88%
90%
87%
89%
86%
90%
90%
88%
1
3
5
7
9
Número de k vizinhos

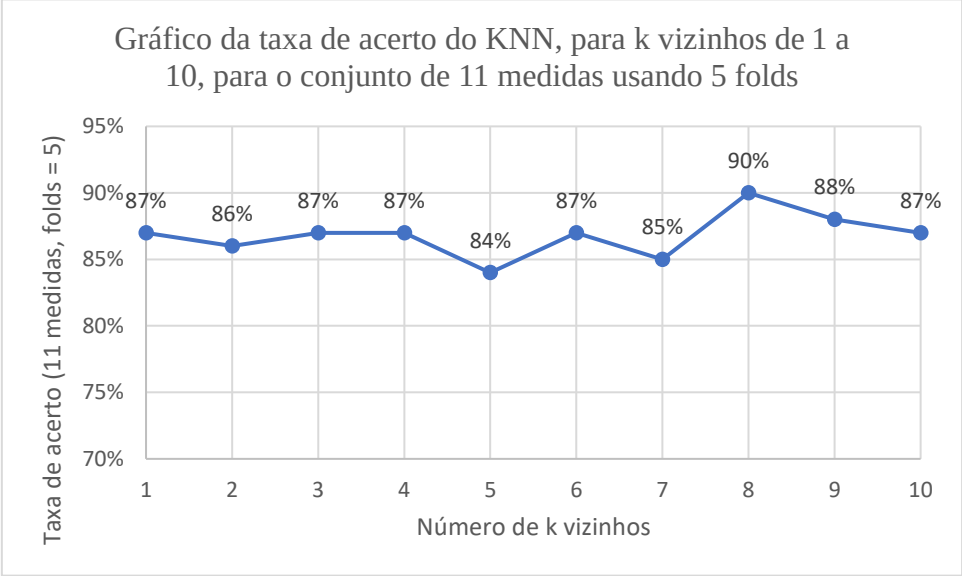


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 5 folds
Taxa de acerto (11 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
87%
86%
87%
87%
84%
87%
85%
90%
88%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

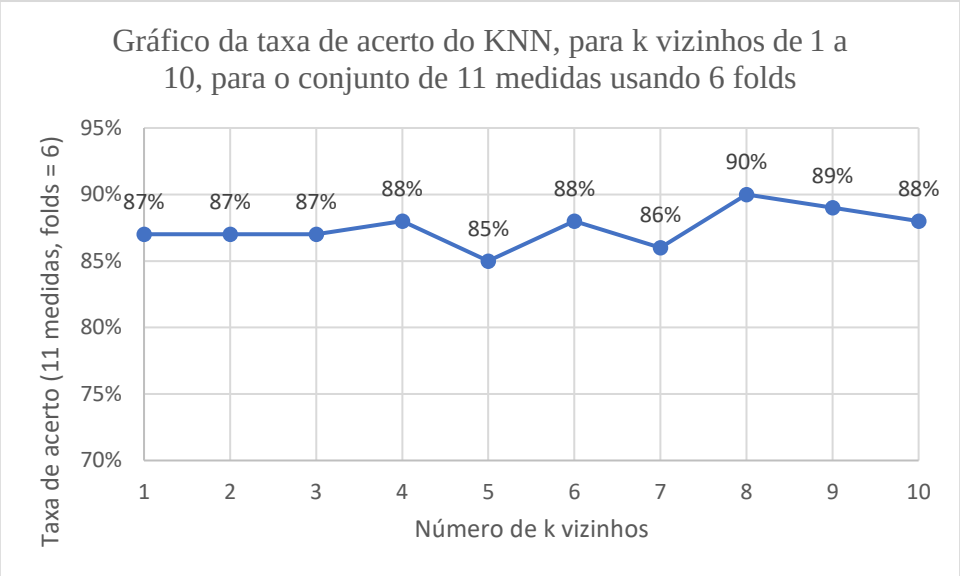


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 6 folds
Taxa de acerto (11 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
87%
87%
87%
88%
85%
88%
86%
90%
89%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

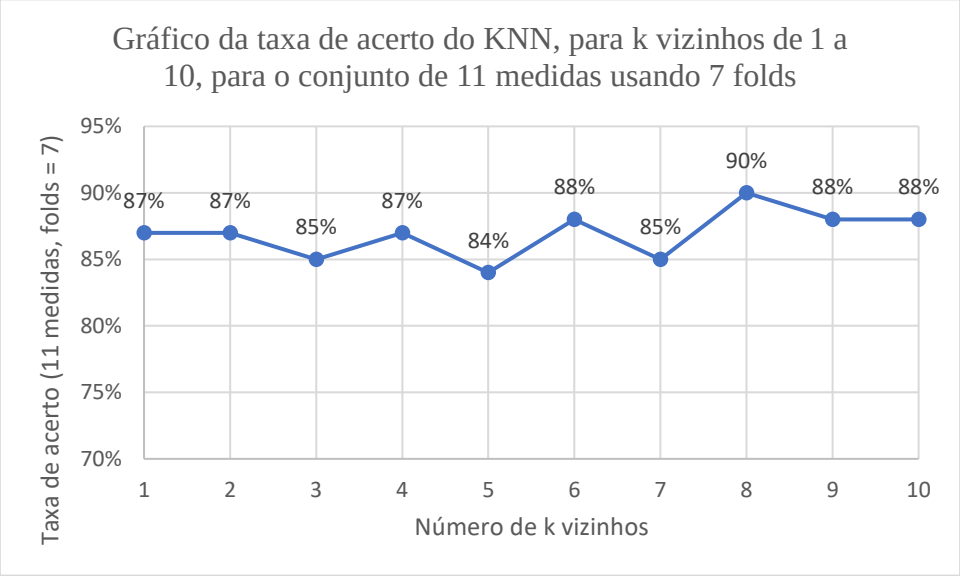


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 7 folds
Taxa de acerto (11 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
87%
87%
85%
87%
84%
88%
85%
90%
88%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

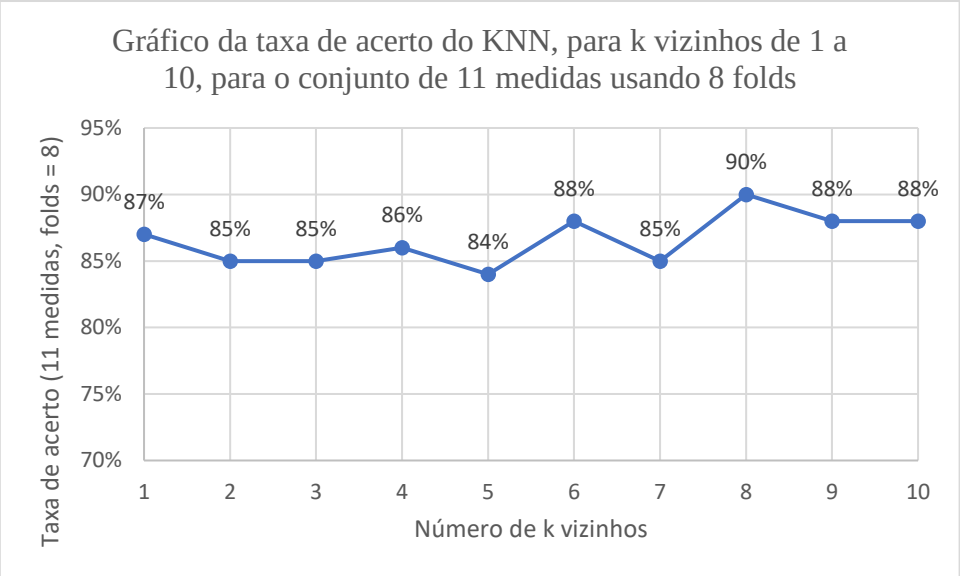


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 8 folds
Taxa de acerto (11 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
87%
85%
85%
86%
84%
88%
85%
90%
88%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

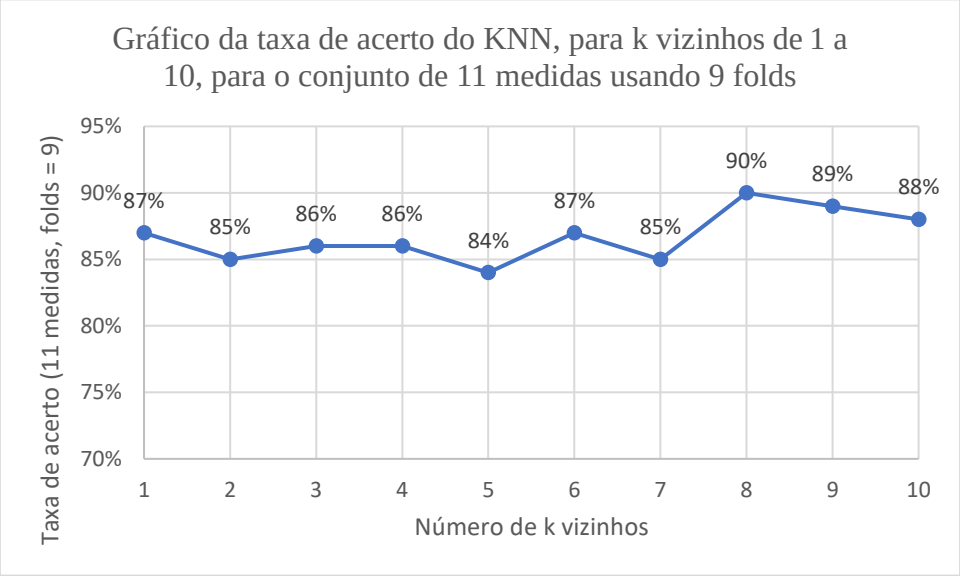


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 9 folds
Taxa de acerto (11 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
87%
85%
86%
86%
84%
87%
85%
90%
89%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

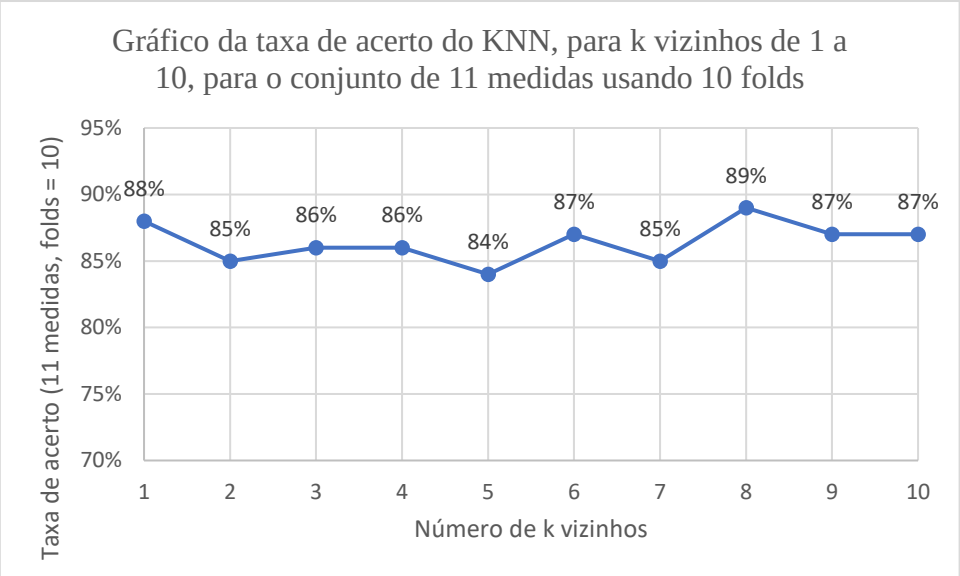


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 11 medidas usando 10 folds
Taxa de acerto (11 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
88%
85%
86%
86%
84%
87%
85%
89%
87%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

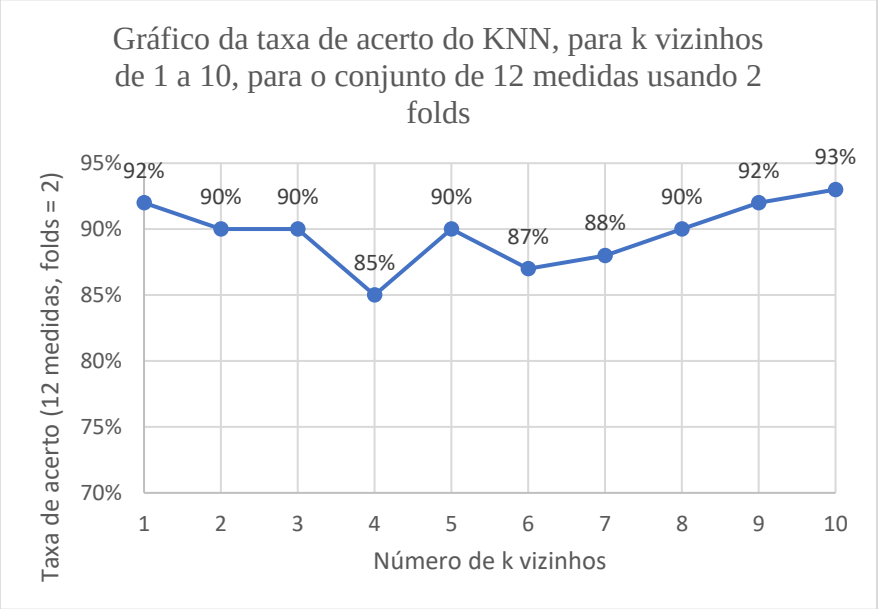


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 2 folds
Taxa de acerto (12 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
92%
90%
90%
85%
90%
87%
88%
90%
92%
93%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

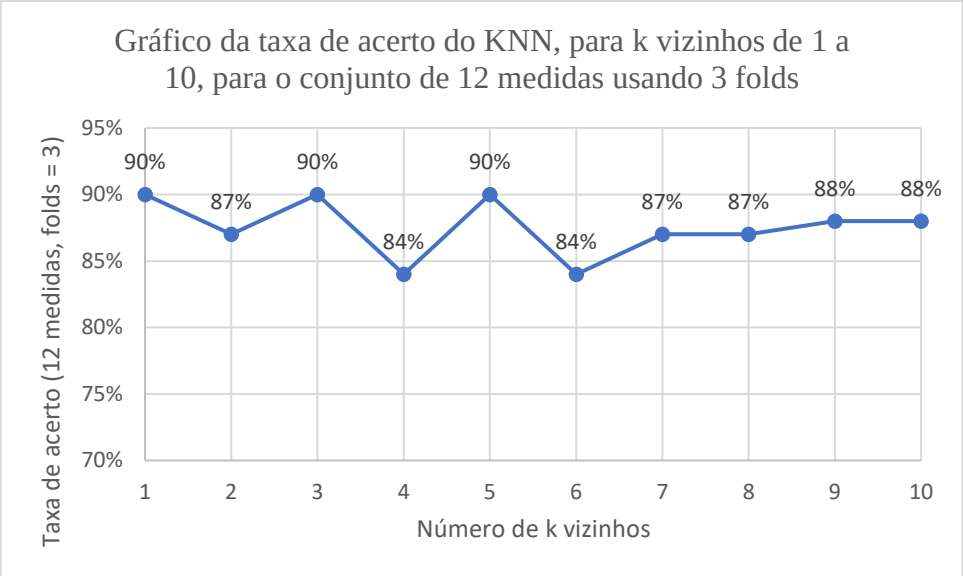


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 3 folds
Taxa de acerto (12 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
90%
87%
90%
84%
90%
84%
87%
87%
88%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

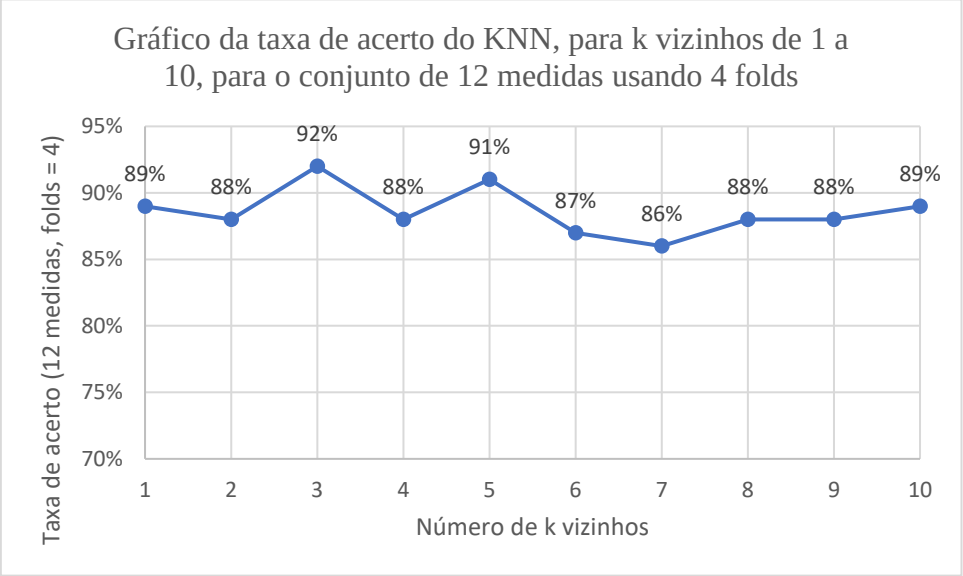


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 4 folds
Taxa de acerto (12 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
89%
88%
92%
88%
91%
87%
86%
88%
88%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

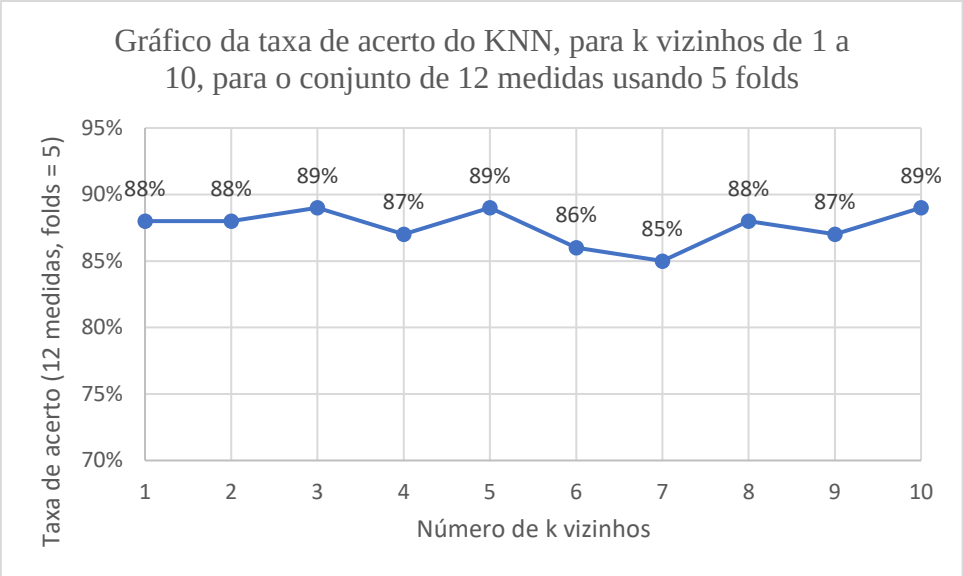


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 5 folds
Taxa de acerto (12 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
88%
88%
89%
87%
89%
86%
85%
88%
87%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

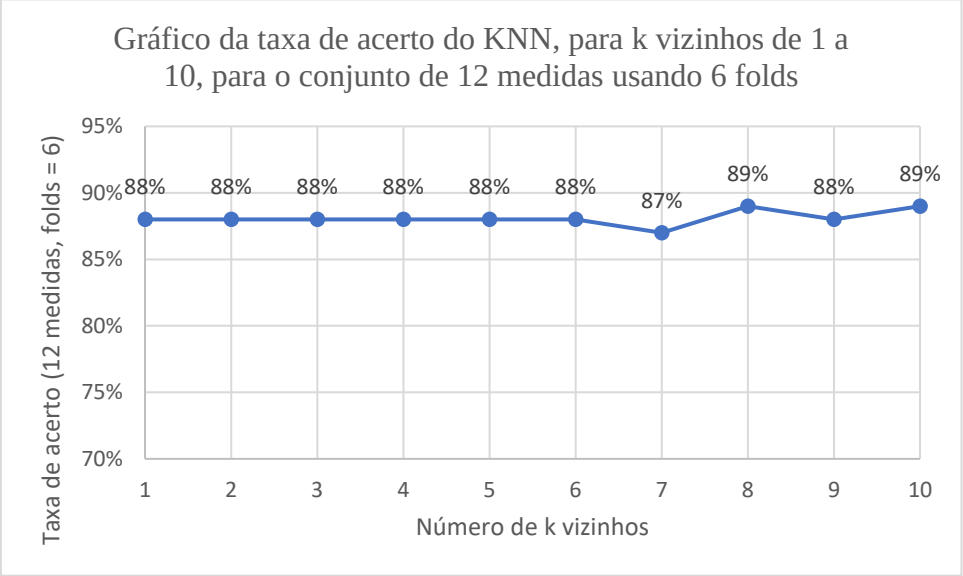


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 6 folds
Taxa de acerto (12 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
88%
88%
88%
88%
88%
88%
87%
89%
88%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

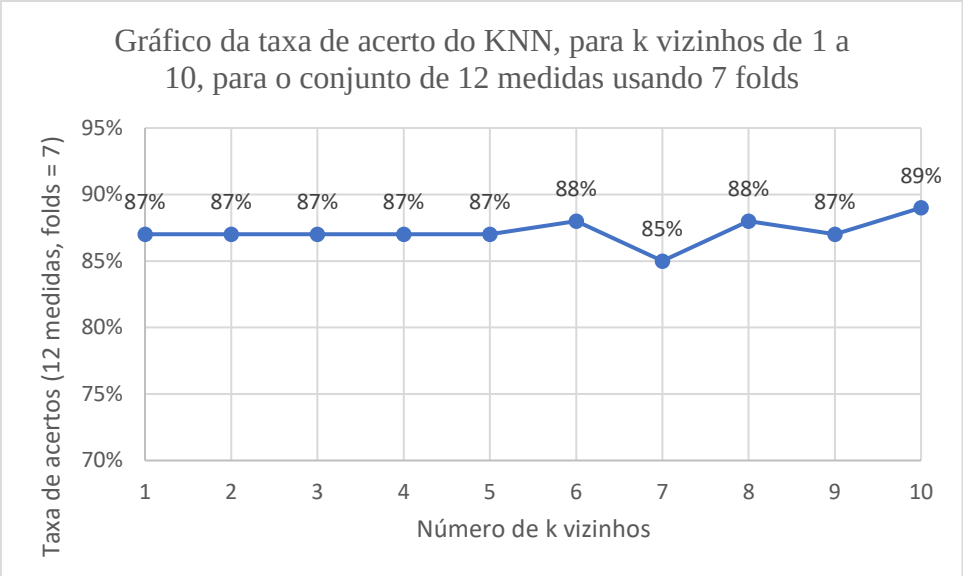


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 7 folds
Taxa de acertos (12 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
87%
87%
87%
87%
87%
88%
85%
88%
87%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

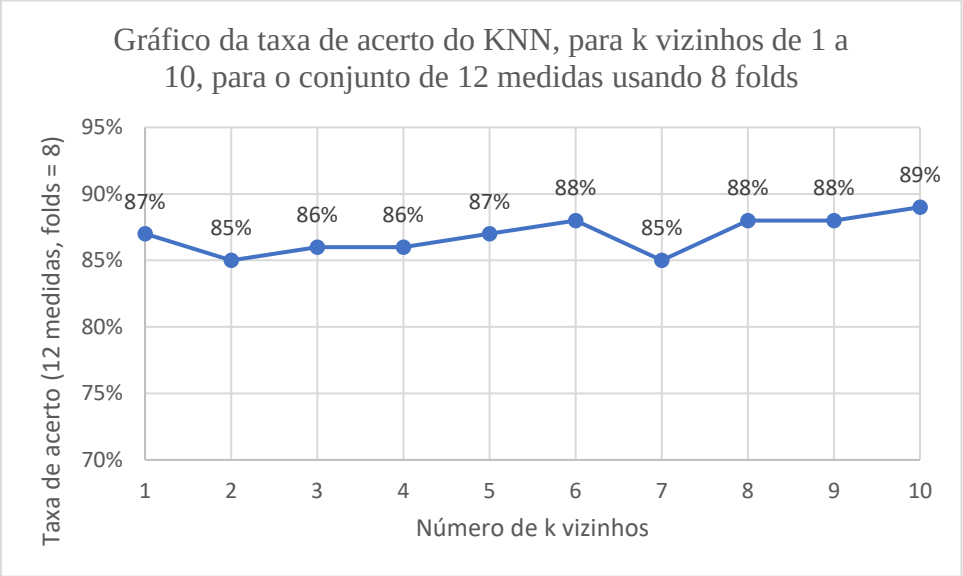


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 8 folds
Taxa de acerto (12 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
87%
85%
86%
86%
87%
88%
85%
88%
88%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

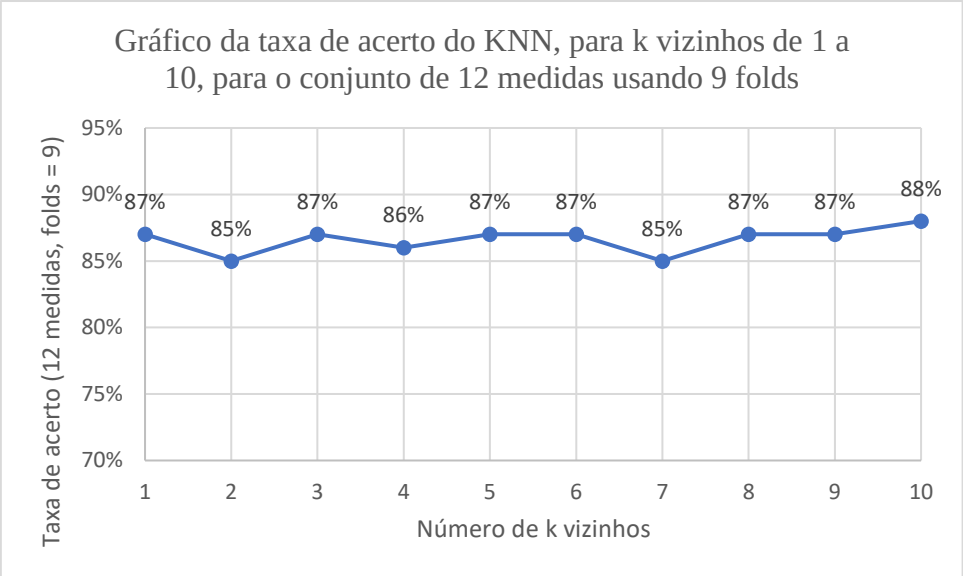


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 9 folds
Taxa de acerto (12 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
87%
85%
87%
86%
87%
87%
85%
87%
87%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

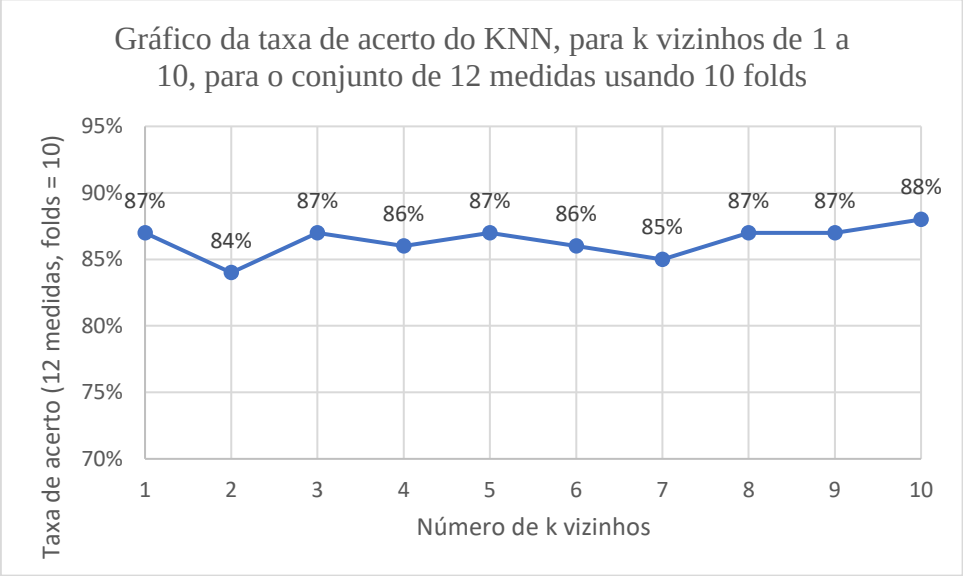


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 12 medidas usando 10 folds
Taxa de acerto (12 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
87%
84%
87%
86%
87%
86%
85%
87%
87%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

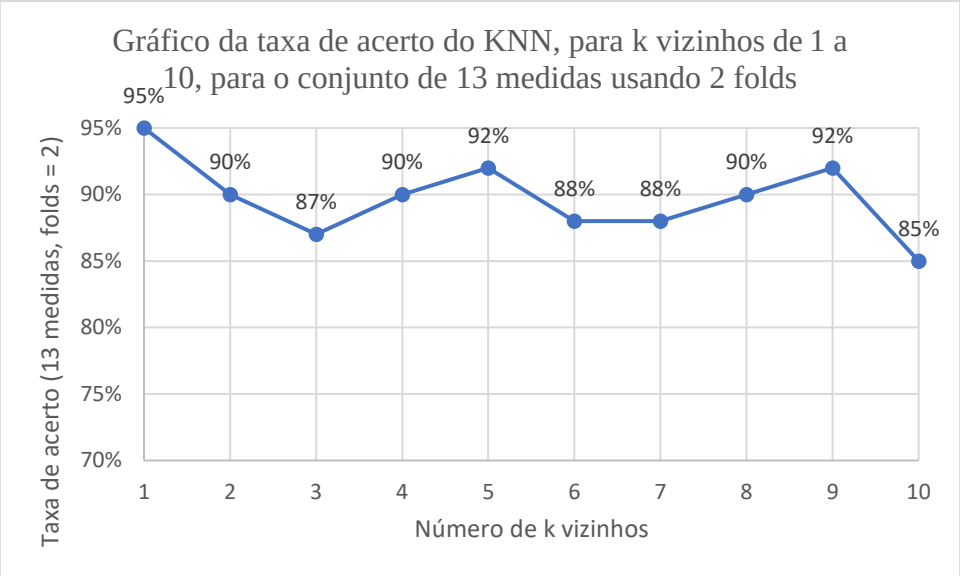


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 2 folds
Taxa de acerto (13 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos
95%
90%
87%
90%
92%
88%
88%
90%
92%
85%

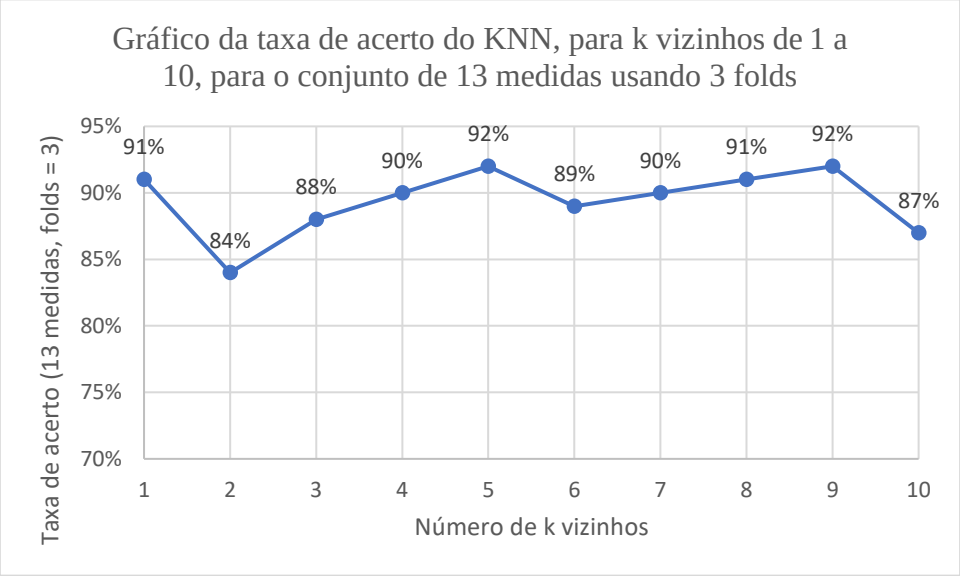


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 3 folds
Taxa de acerto (13 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos
91%
84%
88%
90%
92%
89%
90%
91%
92%
87%

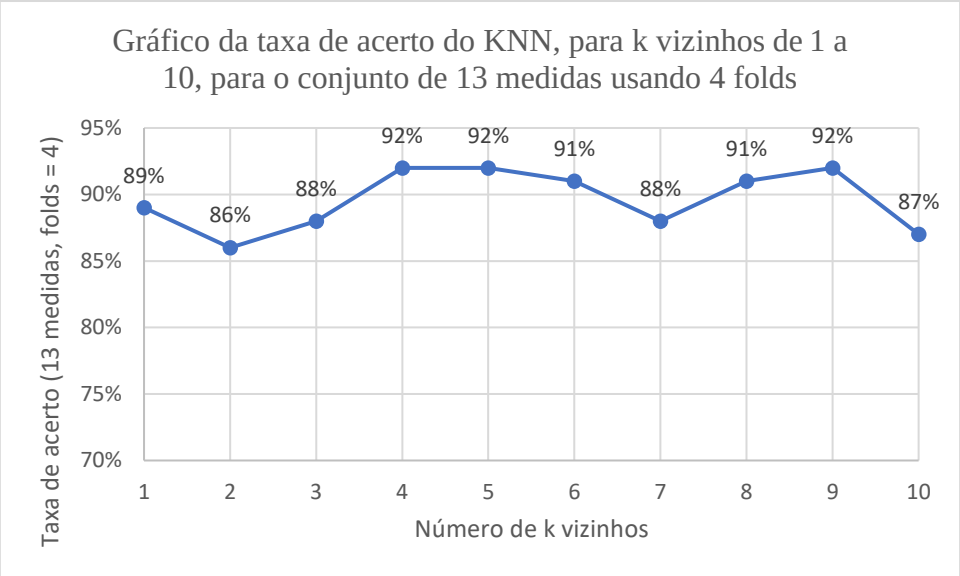


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 4 folds
Taxa de acerto (13 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
89%
86%
88%
92%
92%
91%
88%
91%
92%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

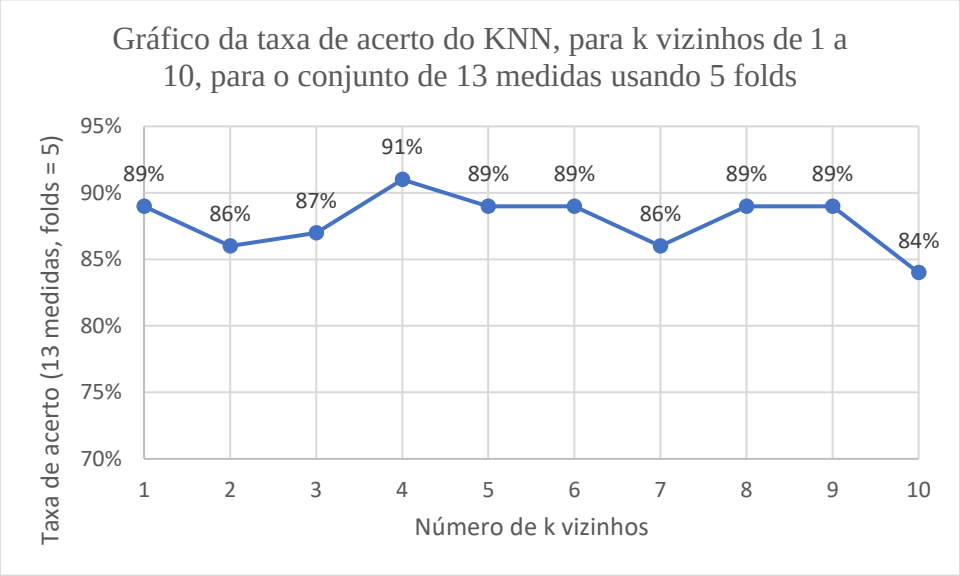


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 5 folds
Taxa de acerto (13 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
89%
86%
87%
91%
89%
89%
86%
89%
89%
84%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

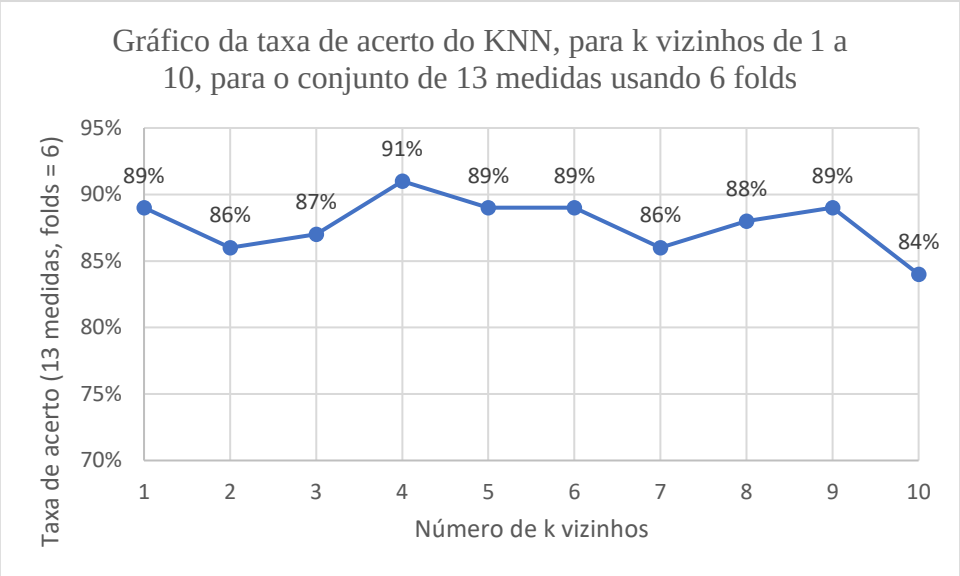


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 6 folds
Taxa de acerto (13 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
89%
86%
87%
91%
89%
89%
86%
88%
89%
84%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

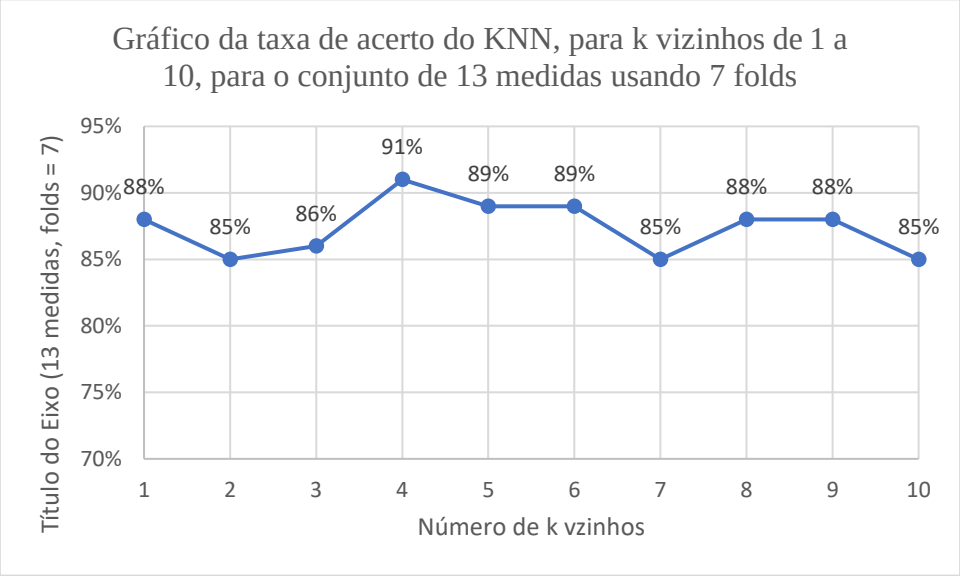


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 7 folds
Título do Eixo (13 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
88%
85%
86%
91%
89%
89%
85%
88%
88%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vzinhos

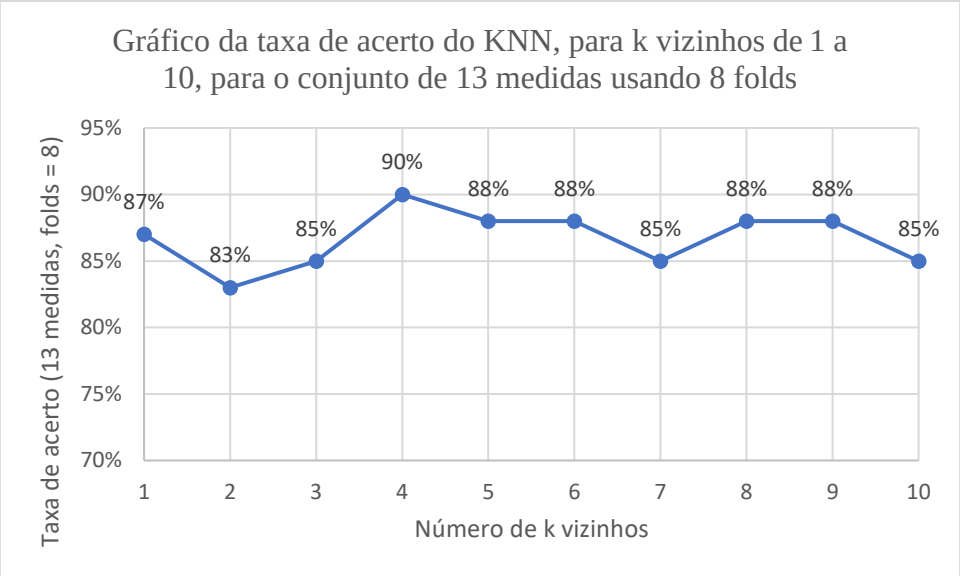


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 8 folds
Taxa de acerto (13 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
87%
83%
85%
90%
88%
88%
85%
88%
88%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

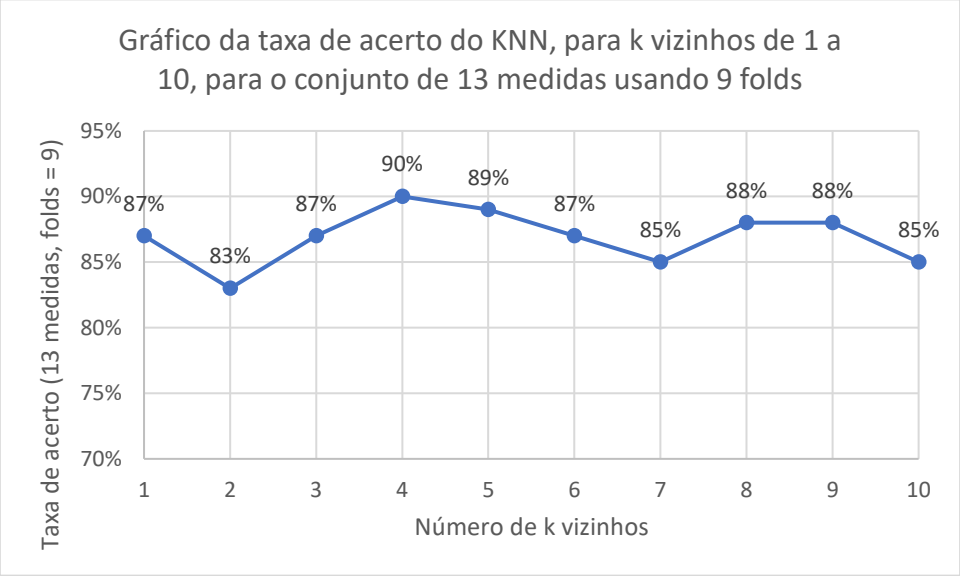


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 9 folds
Taxa de acerto (13 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
87%
83%
87%
90%
89%
87%
85%
88%
88%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

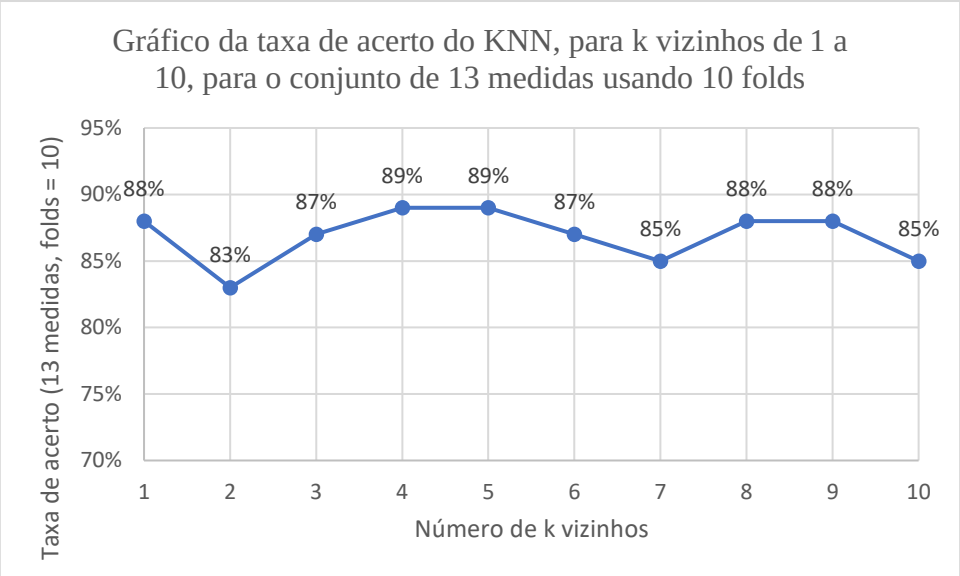


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 13 medidas usando 10 folds
Taxa de acerto (13 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
88%
83%
87%
89%
89%
87%
85%
88%
88%
85%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

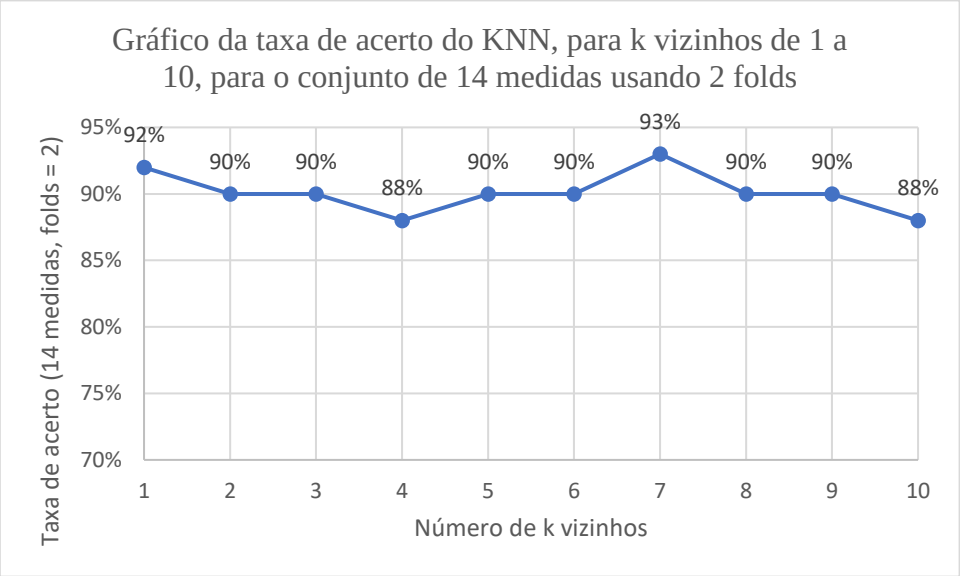


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 2 folds
Taxa de acerto (14 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
92%
90%
90%
88%
90%
90%
93%
90%
90%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

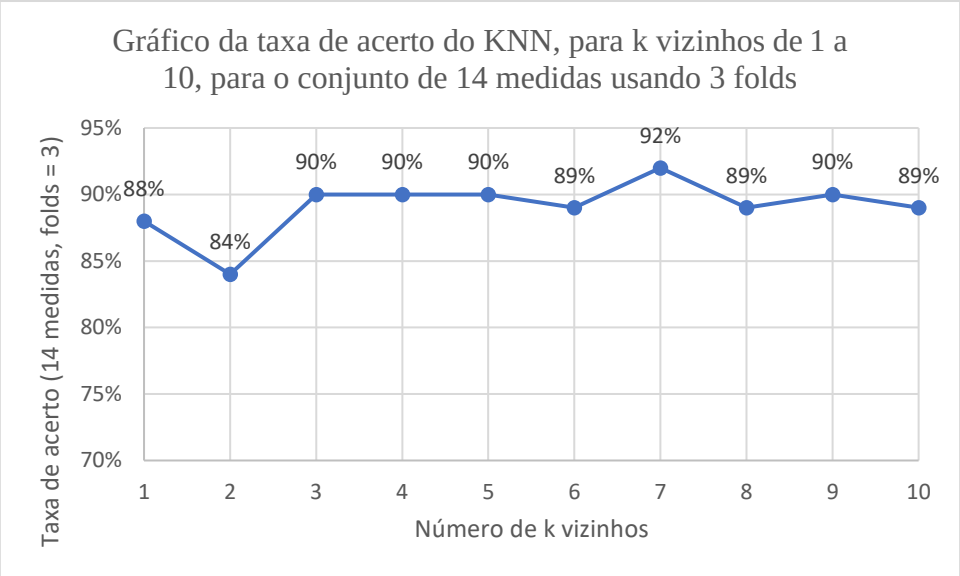


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 3 folds
Taxa de acerto (14 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
88%
84%
90%
90%
90%
89%
92%
89%
90%
89%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

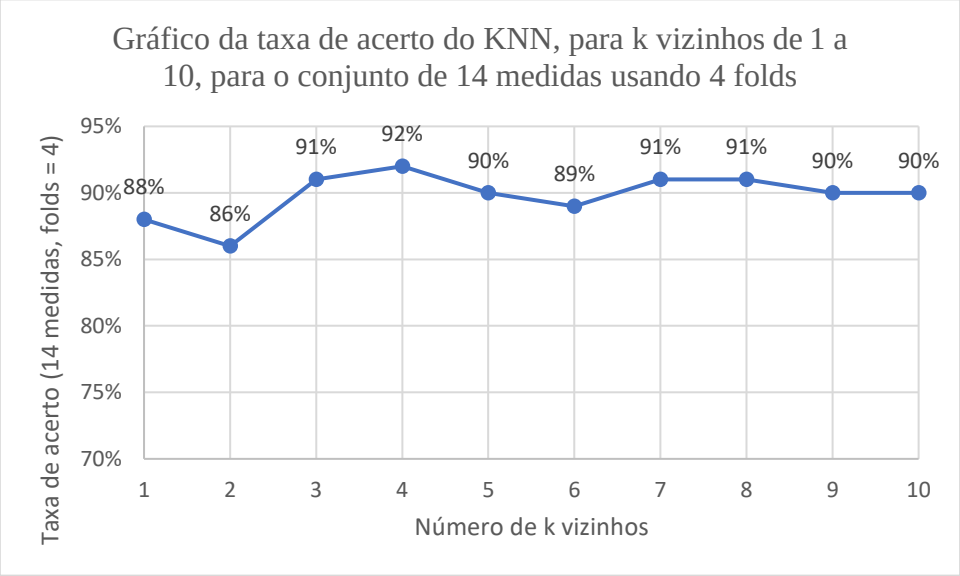


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 4 folds
Taxa de acerto (14 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
88%
86%
91%
92%
90%
89%
91%
91%
90%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

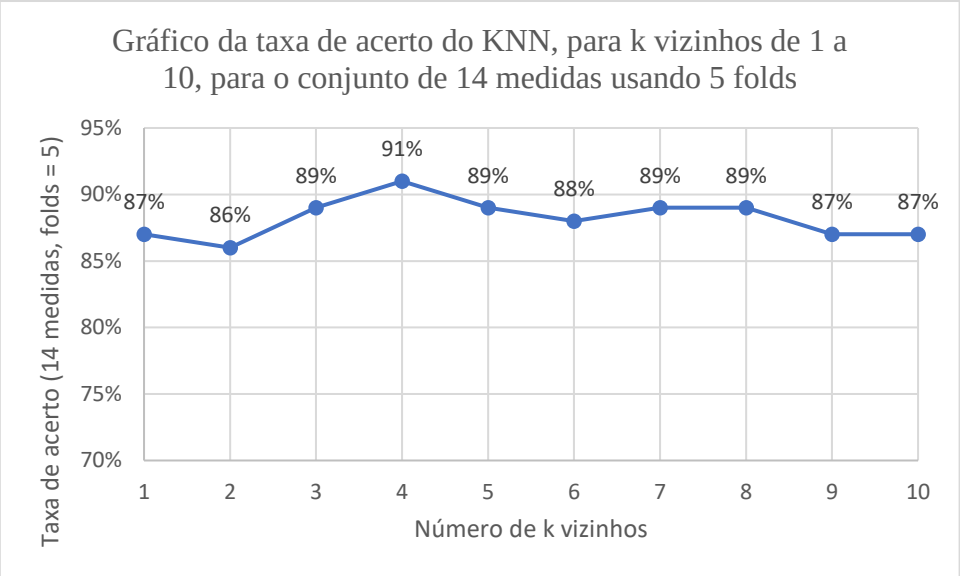


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 5 folds
Taxa de acerto (14 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
87%
86%
89%
91%
89%
88%
89%
89%
87%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

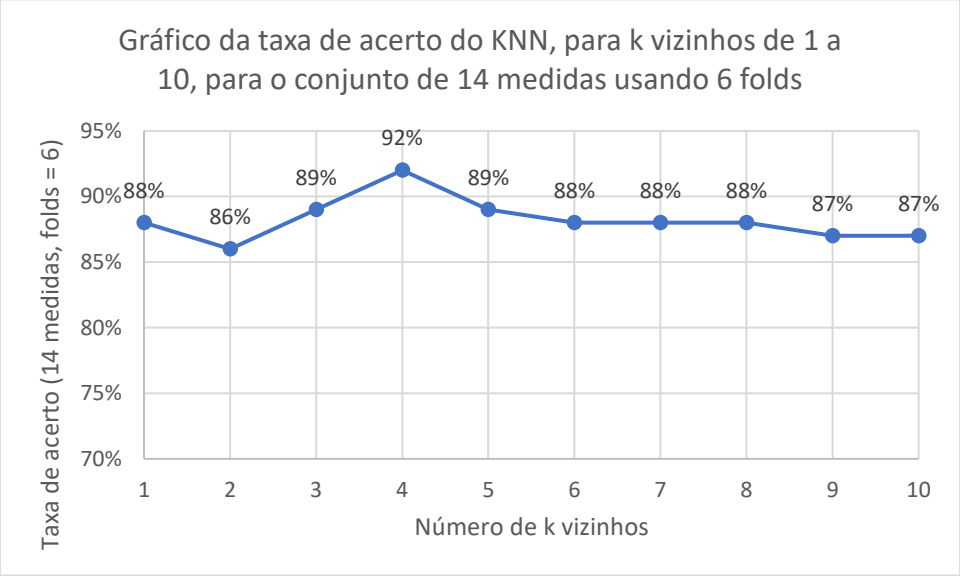


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 6 folds
Taxa de acerto (14 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
88%
86%
89%
92%
89%
88%
88%
88%
87%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

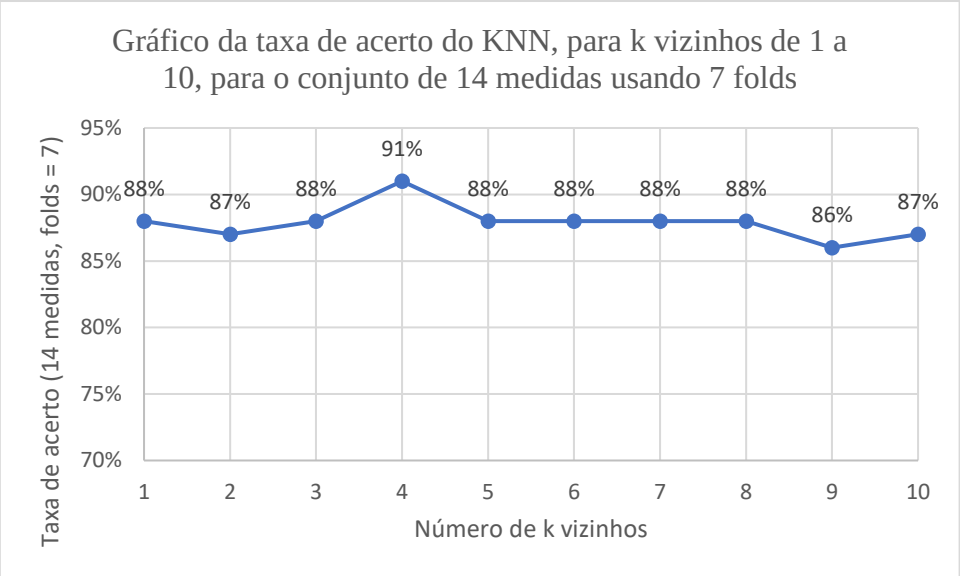


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 7 folds
Taxa de acerto (14 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
88%
87%
88%
91%
88%
88%
88%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

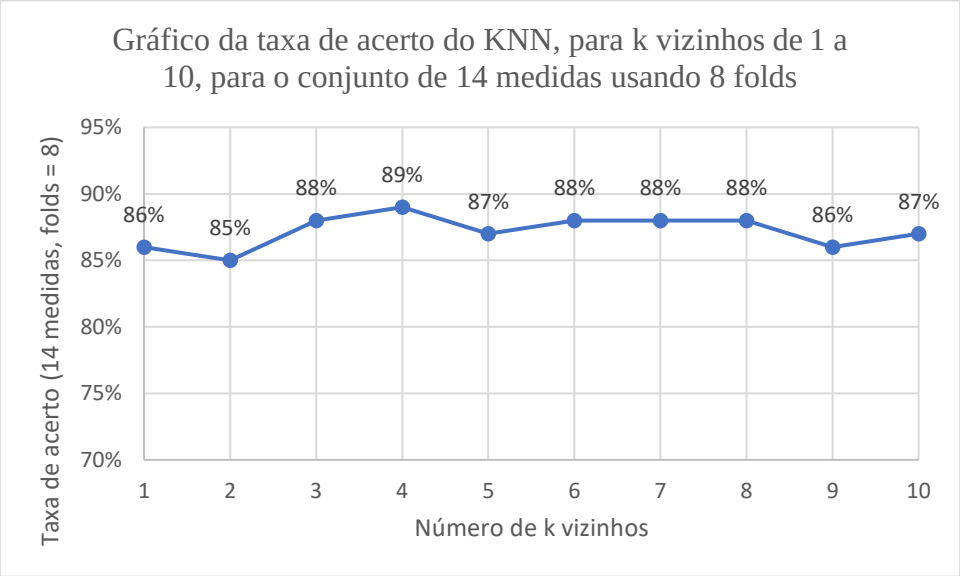


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 8 folds
Taxa de acerto (14 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
86%
85%
88%
89%
87%
88%
88%
88%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

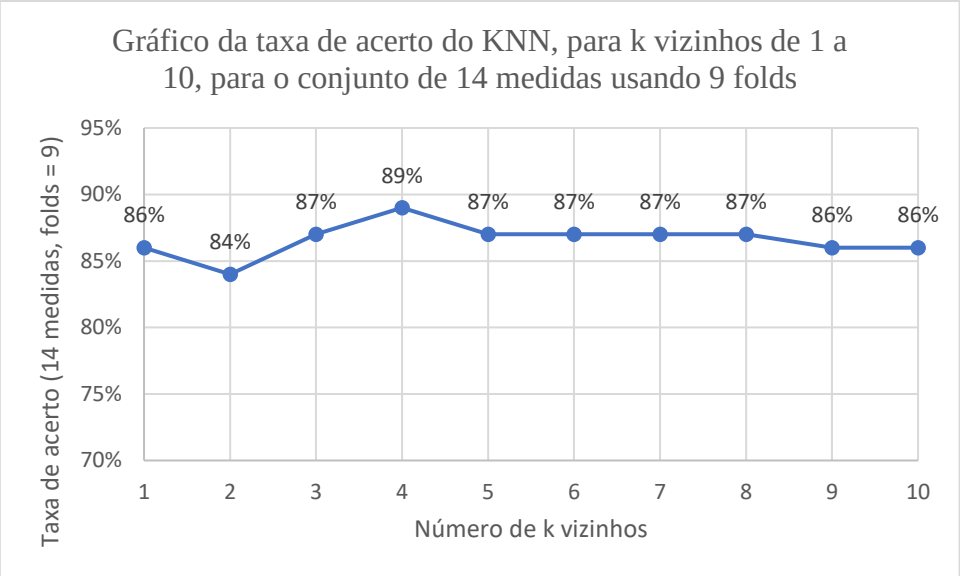


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 9 folds
Taxa de acerto (14 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
86%
84%
87%
89%
87%
87%
87%
87%
86%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

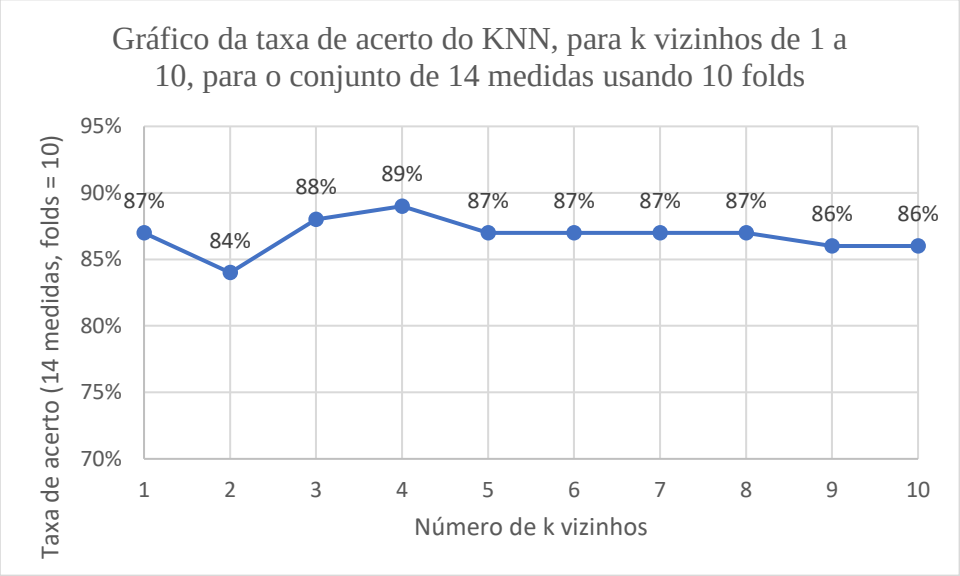


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 14 medidas usando 10 folds
Taxa de acerto (14 medidas, folds = 10)
95%
90%
85%
80%
75%
70%
87%
84%
88%
89%
87%
87%
87%
87%
86%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

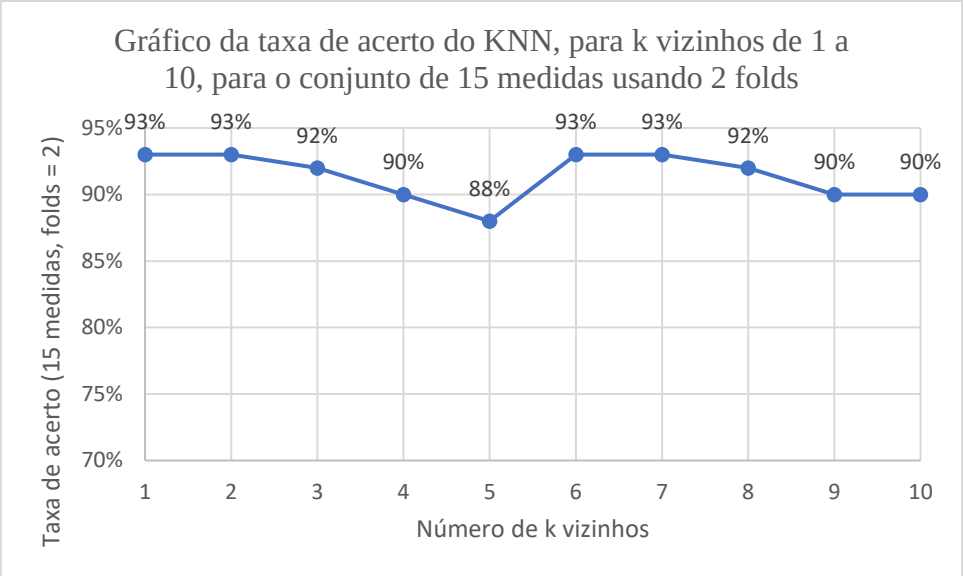


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 2 folds
Taxa de acerto (15 medidas, folds = 2)
95%
90%
85%
80%
75%
70%
93%
93%
92%
90%
88%
93%
93%
92%
90%
90%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

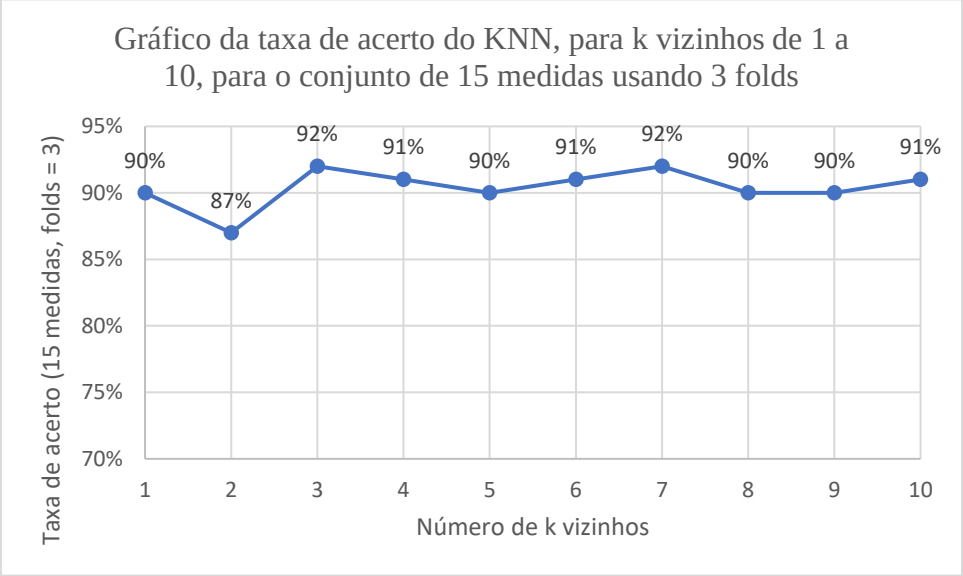


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 3 folds
Taxa de acerto (15 medidas, folds = 3)
95%
90%
85%
80%
75%
70%
90%
87%
92%
91%
90%
91%
92%
90%
90%
91%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

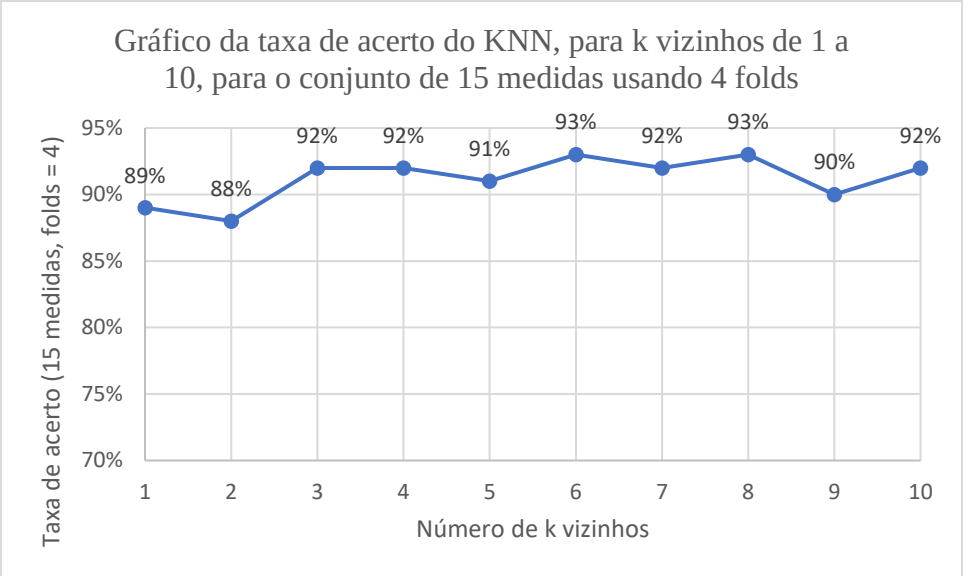


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 4 folds
Taxa de acerto (15 medidas, folds = 4)
95%
90%
85%
80%
75%
70%
89%
88%
92%
92%
91%
93%
92%
93%
90%
92%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

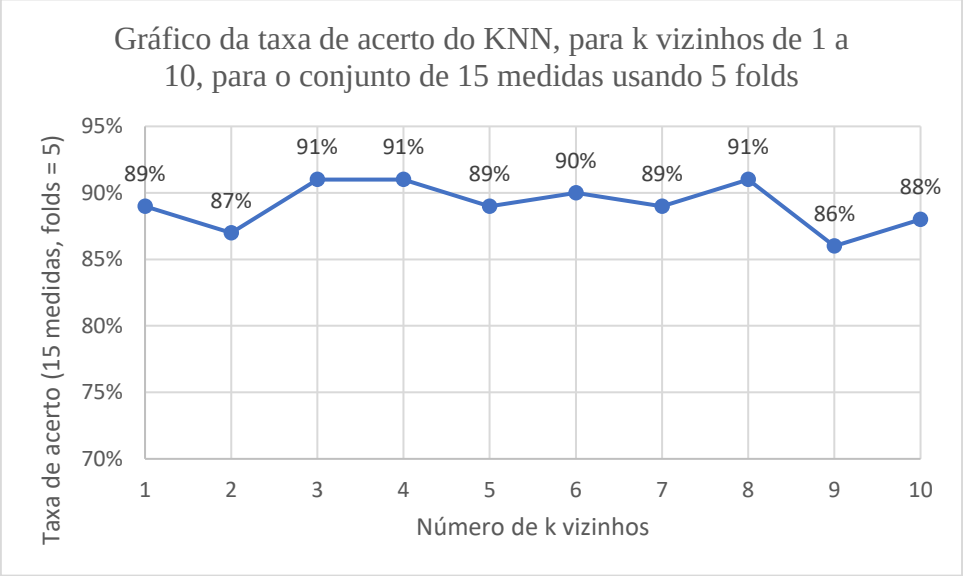


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 5 folds
Taxa de acerto (15 medidas, folds = 5)
95%
90%
85%
80%
75%
70%
89%
87%
91%
91%
89%
90%
89%
91%
86%
88%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

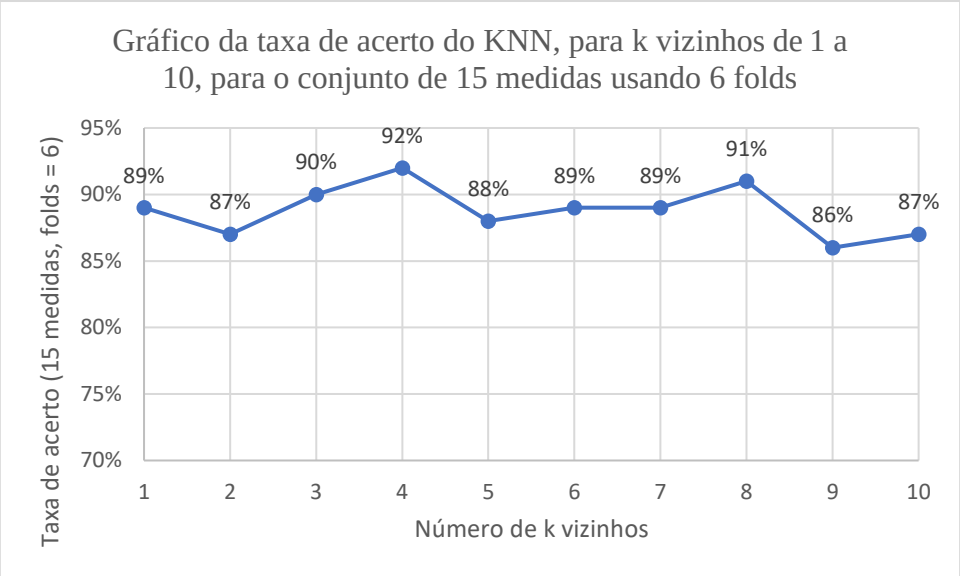


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 6 folds
Taxa de acerto (15 medidas, folds = 6)
95%
90%
85%
80%
75%
70%
89%
87%
90%
92%
88%
89%
89%
91%
86%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

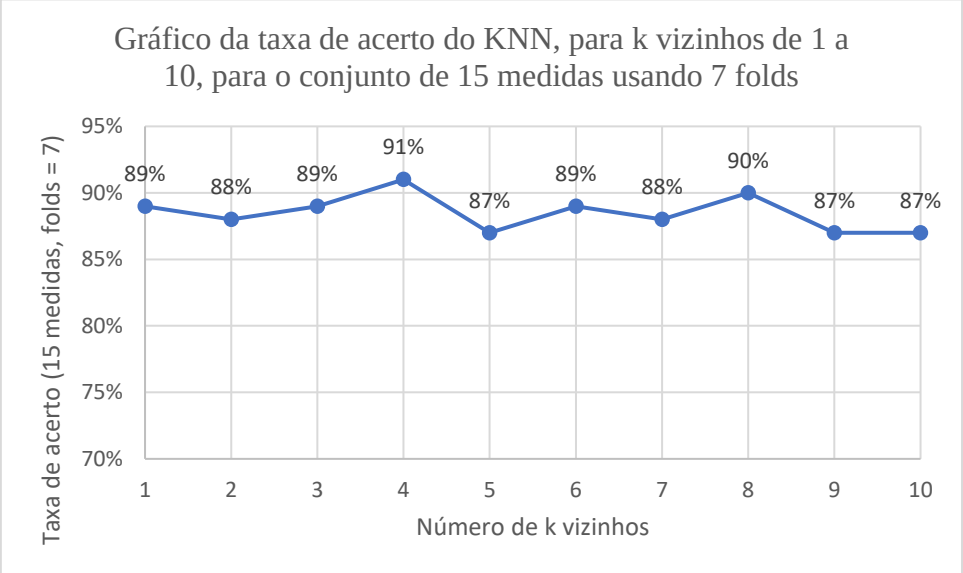


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 7 folds
Taxa de acerto (15 medidas, folds = 7)
95%
90%
85%
80%
75%
70%
89%
88%
89%
91%
87%
89%
88%
90%
87%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

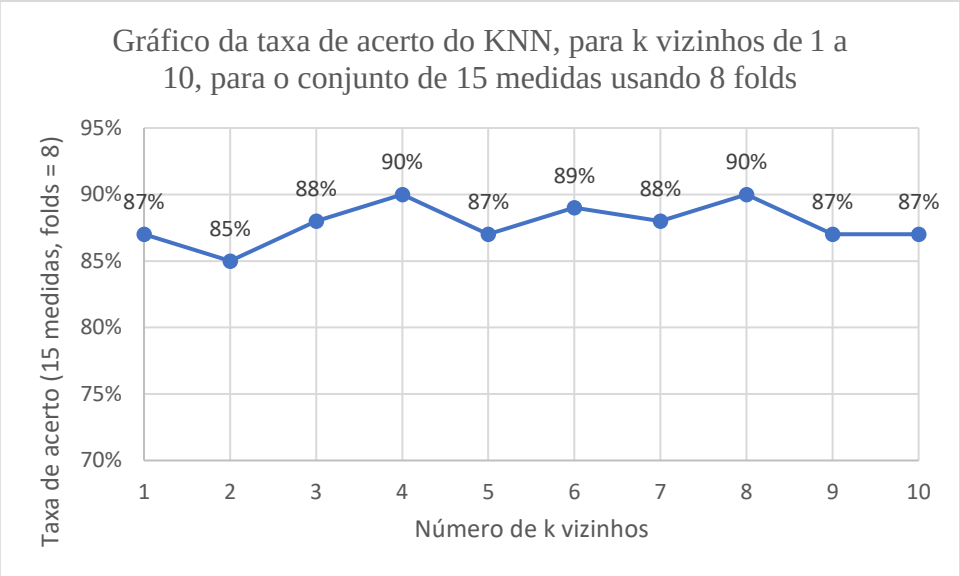


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 8 folds
Taxa de acerto (15 medidas, folds = 8)
95%
90%
85%
80%
75%
70%
87%
85%
88%
90%
87%
89%
88%
90%
87%
87%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos

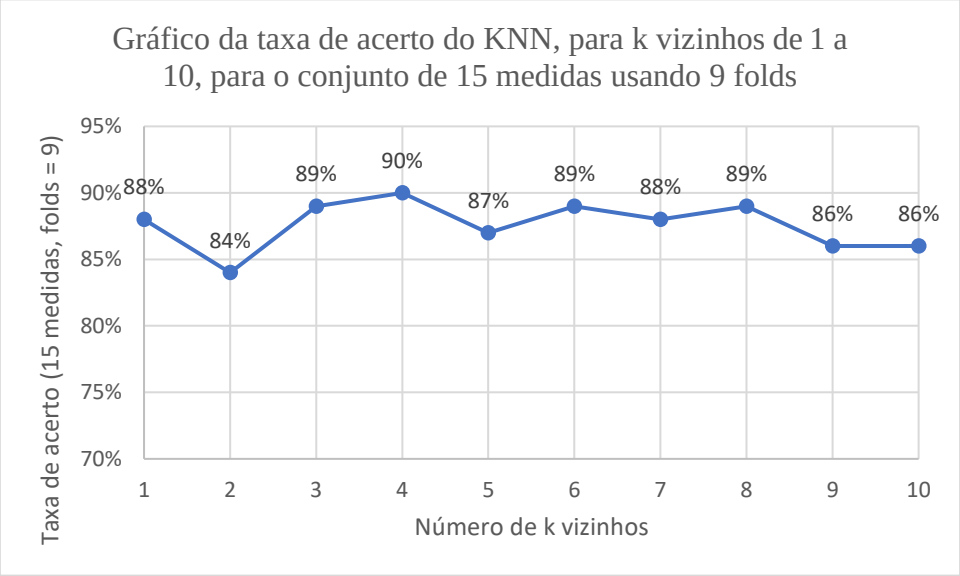
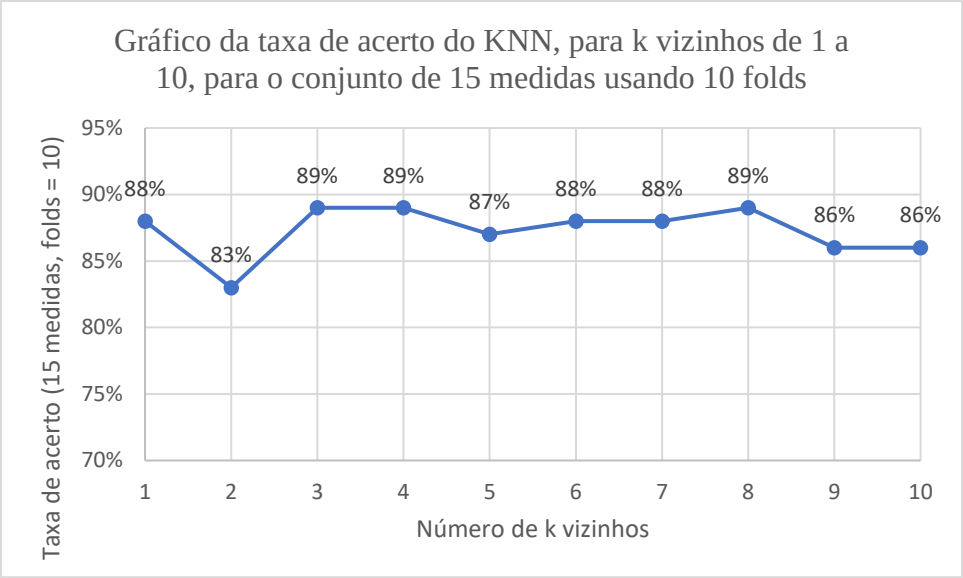
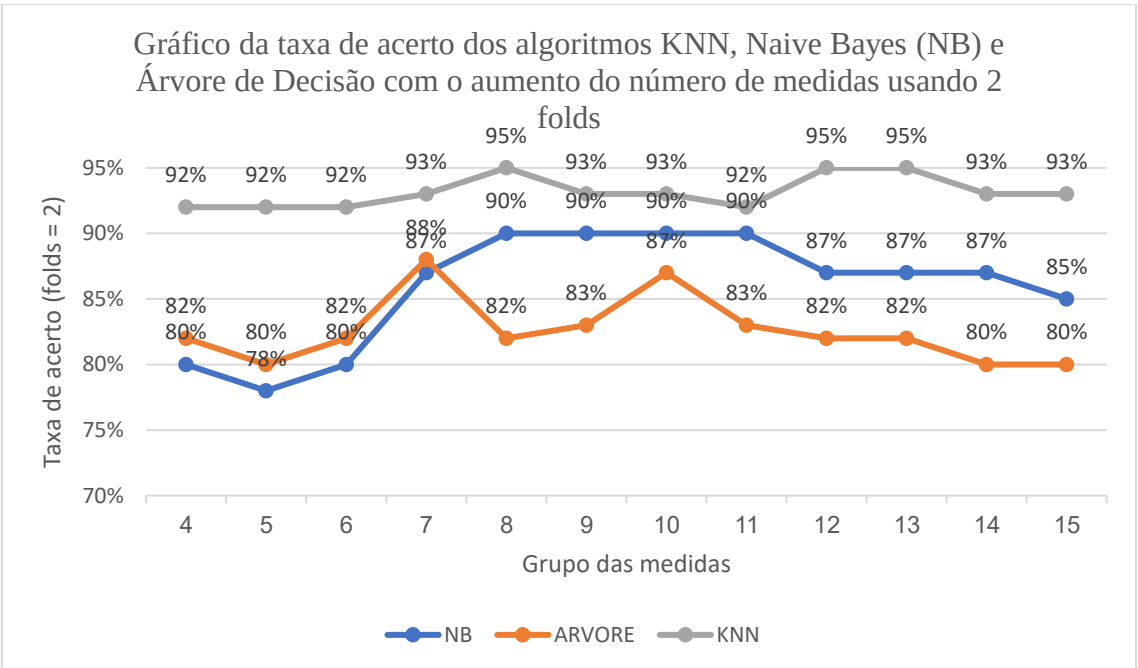


Gráfico da taxa de acerto do KNN, para k vizinhos de 1 a 10, para o conjunto de 15 medidas usando 9 folds
Taxa de acerto (15 medidas, folds = 9)
95%
90%
85%
80%
75%
70%
88%
84%
89%
90%
87%
89%
88%
89%
86%
86%
1
2
3
4
5
6
7
8
9
10
Número de k vizinhos



APÊNDICE B - Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando de 2 a 10 folds



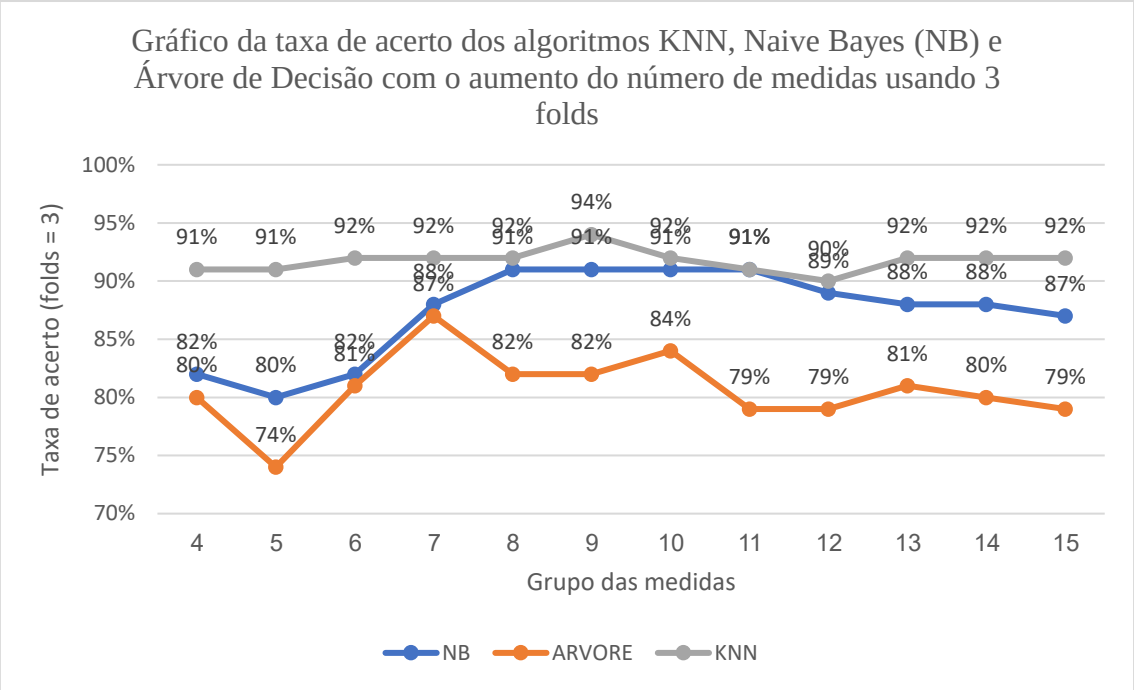


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 3 folds
Taxa de acerto (folds = 3)
100%
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

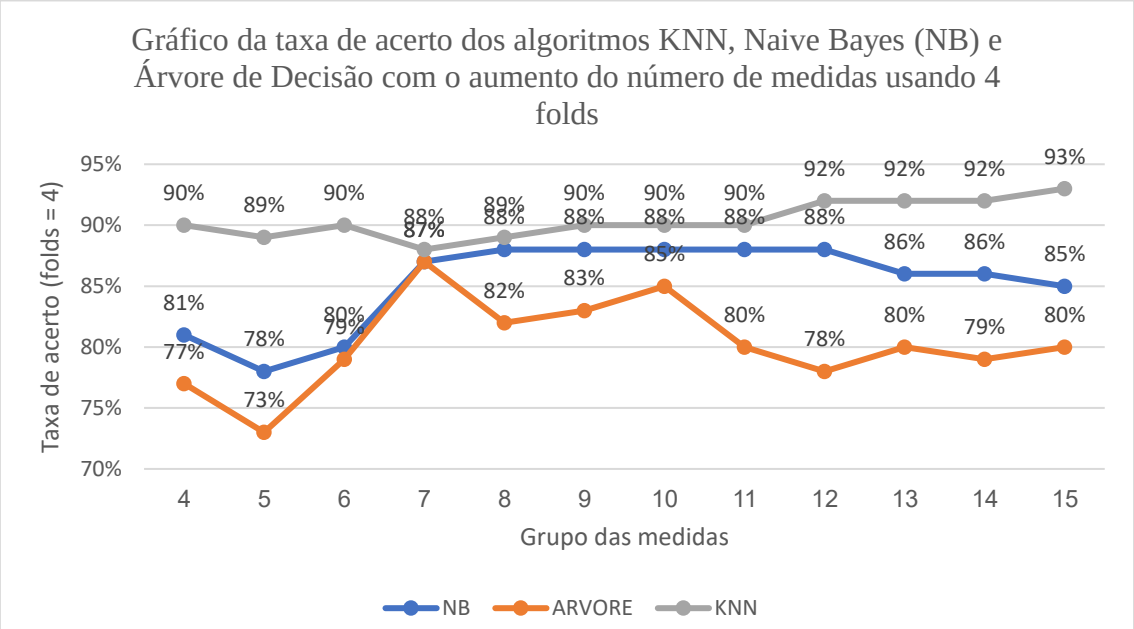


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 4 folds
Taxa de acerto (folds = 4)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

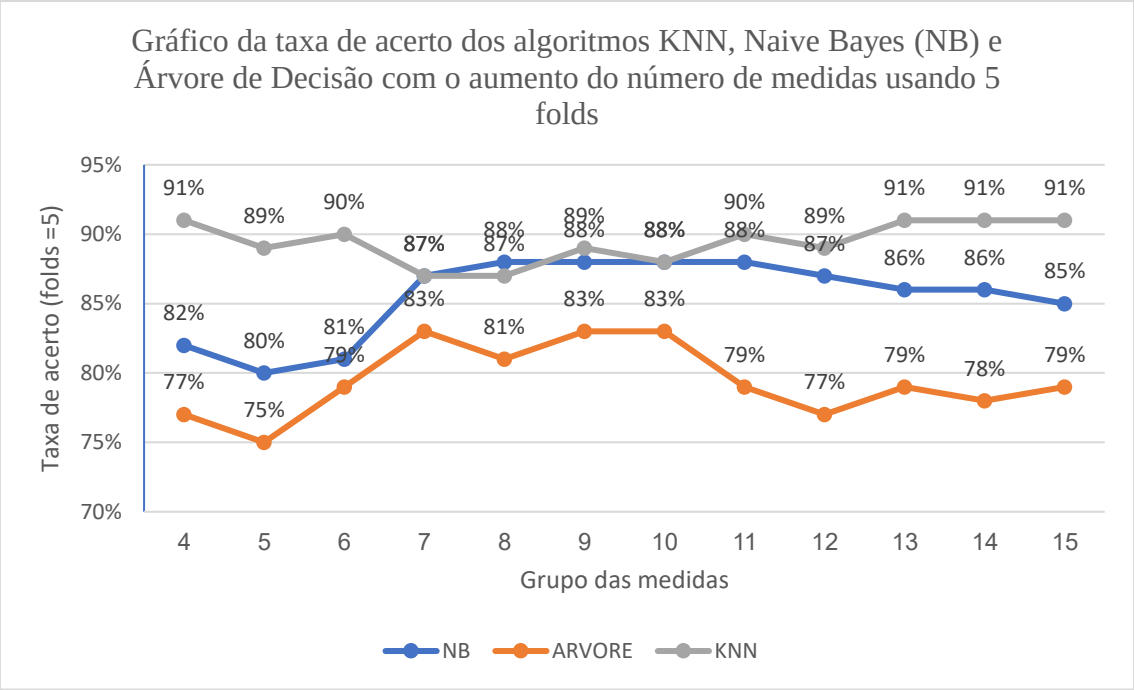


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 5 folds
Taxa de acerto (folds =5)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

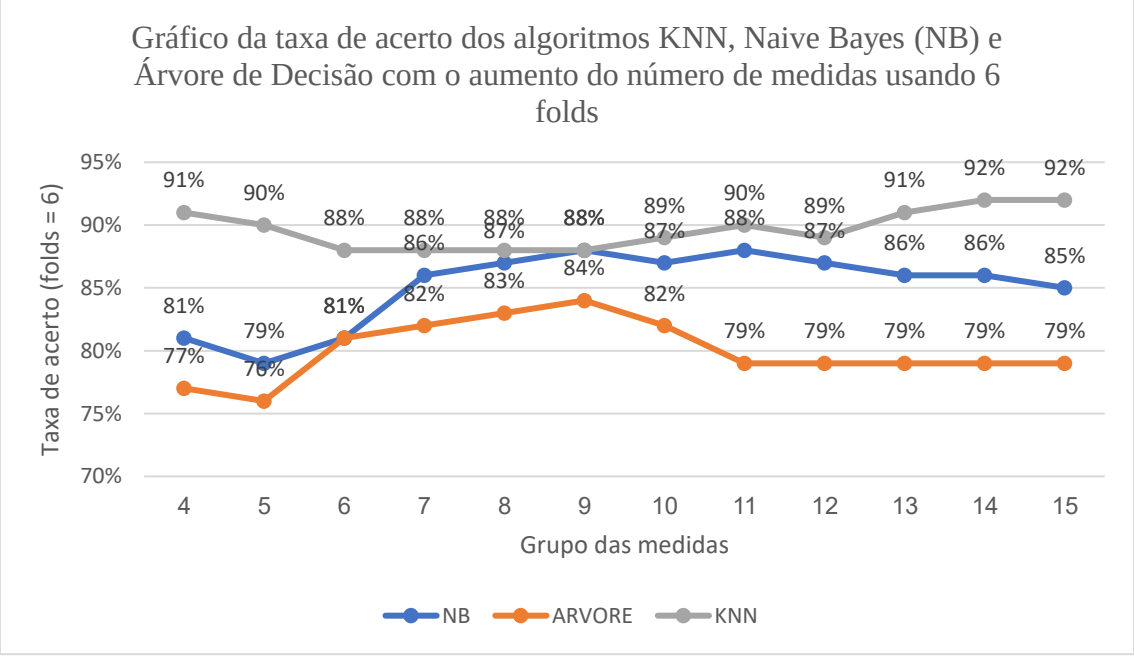


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 6 folds
Taxa de acerto (folds = 6)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

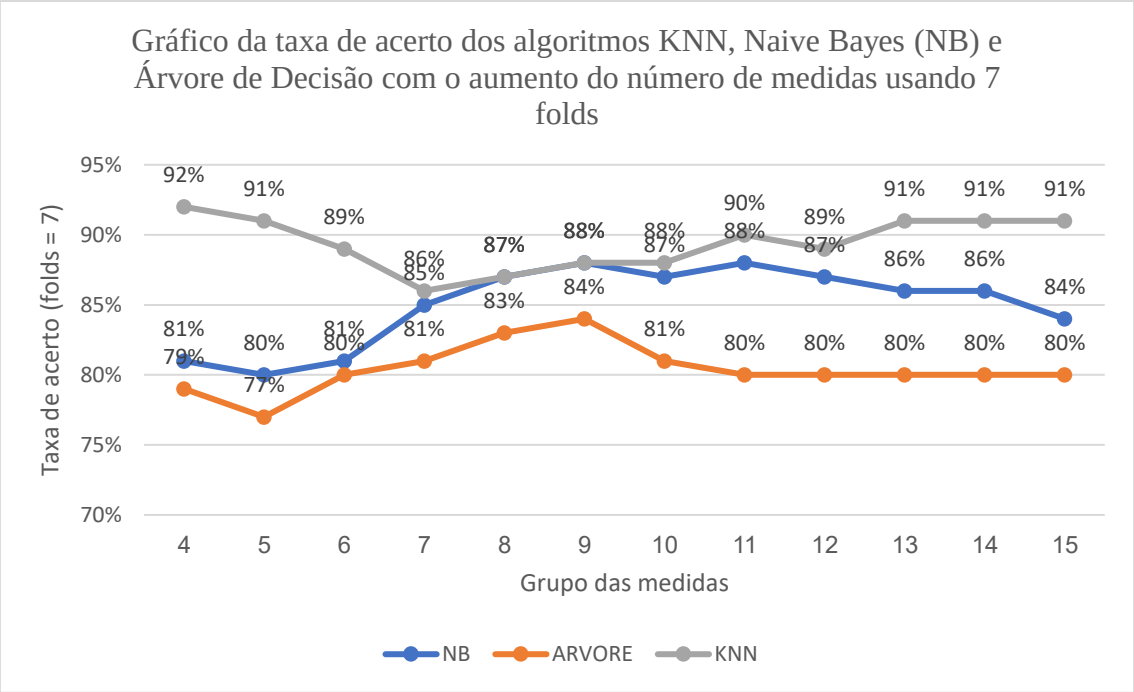


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 7 folds
Taxa de acerto (folds = 7)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

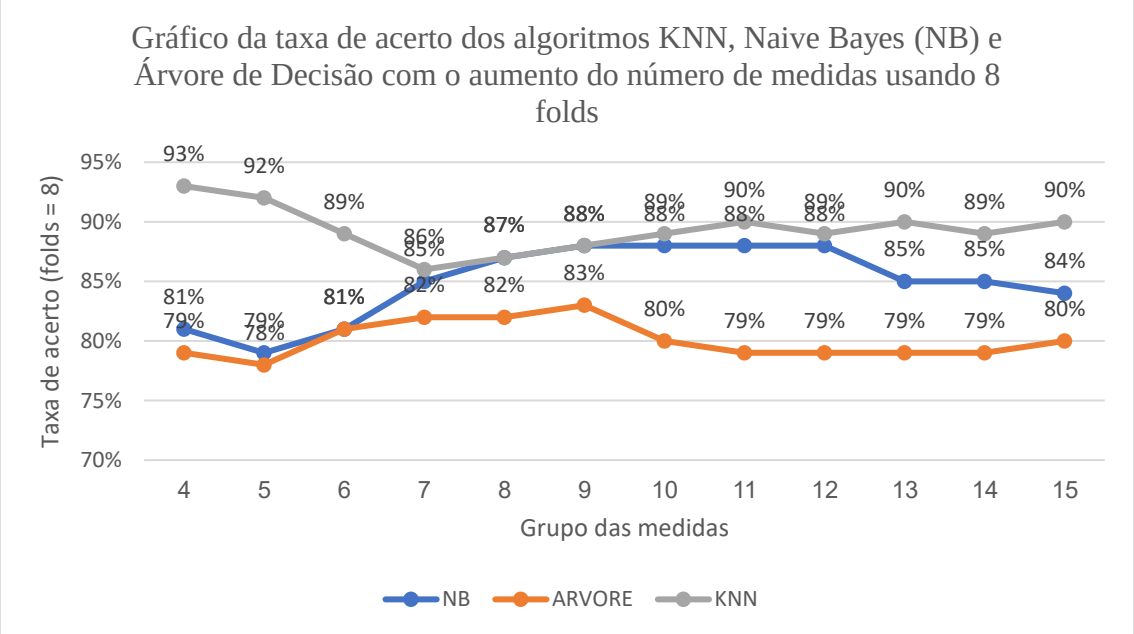


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 8 folds
Taxa de acerto (folds = 8)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

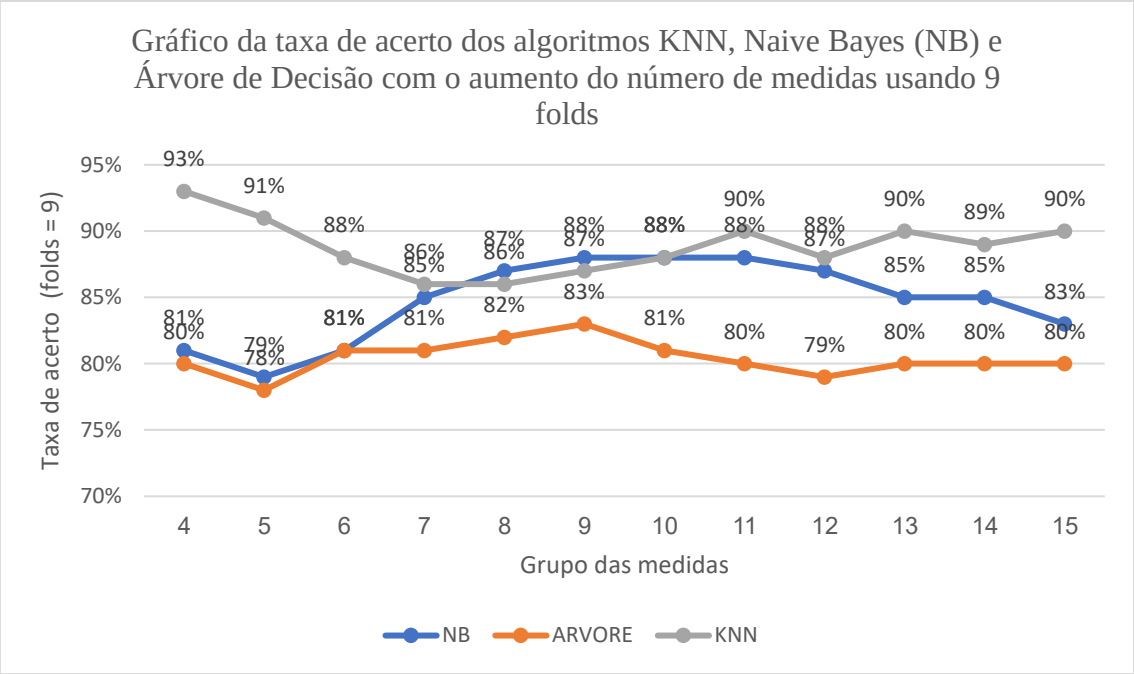


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 9 folds
Taxa de acerto (folds = 9)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

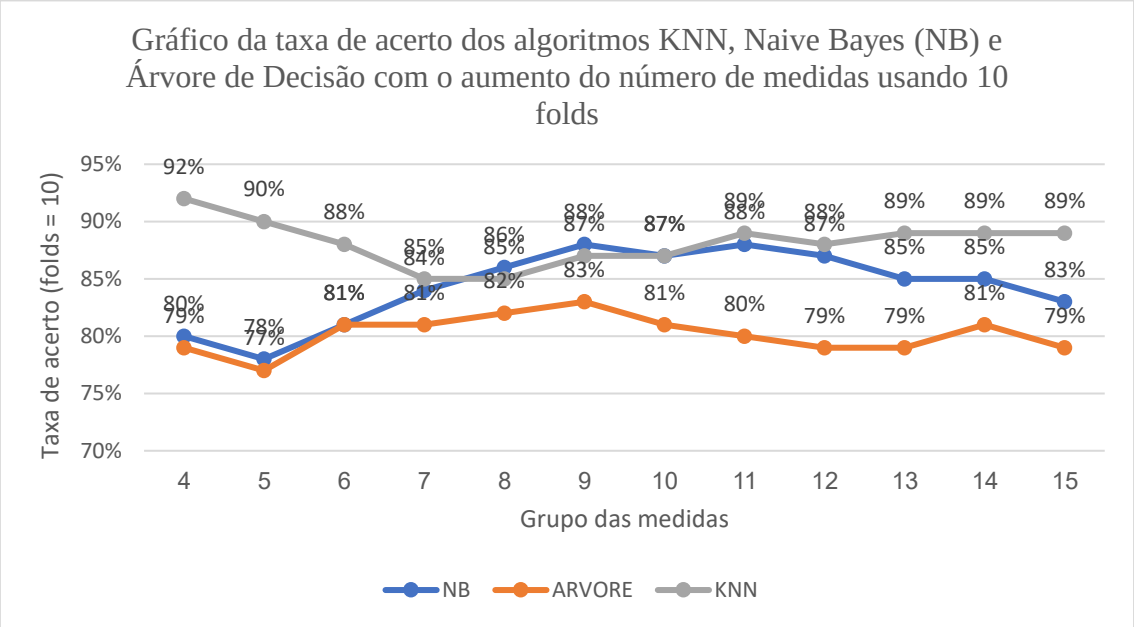


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão com o aumento do número de medidas usando 10 folds
Taxa de acerto (folds = 10)
95%
90%
85%
80%
75%
70%
4
5
6
7
8
9
10
11
12
13
14
15
Grupo das medidas
NB
ARVORE
KNN

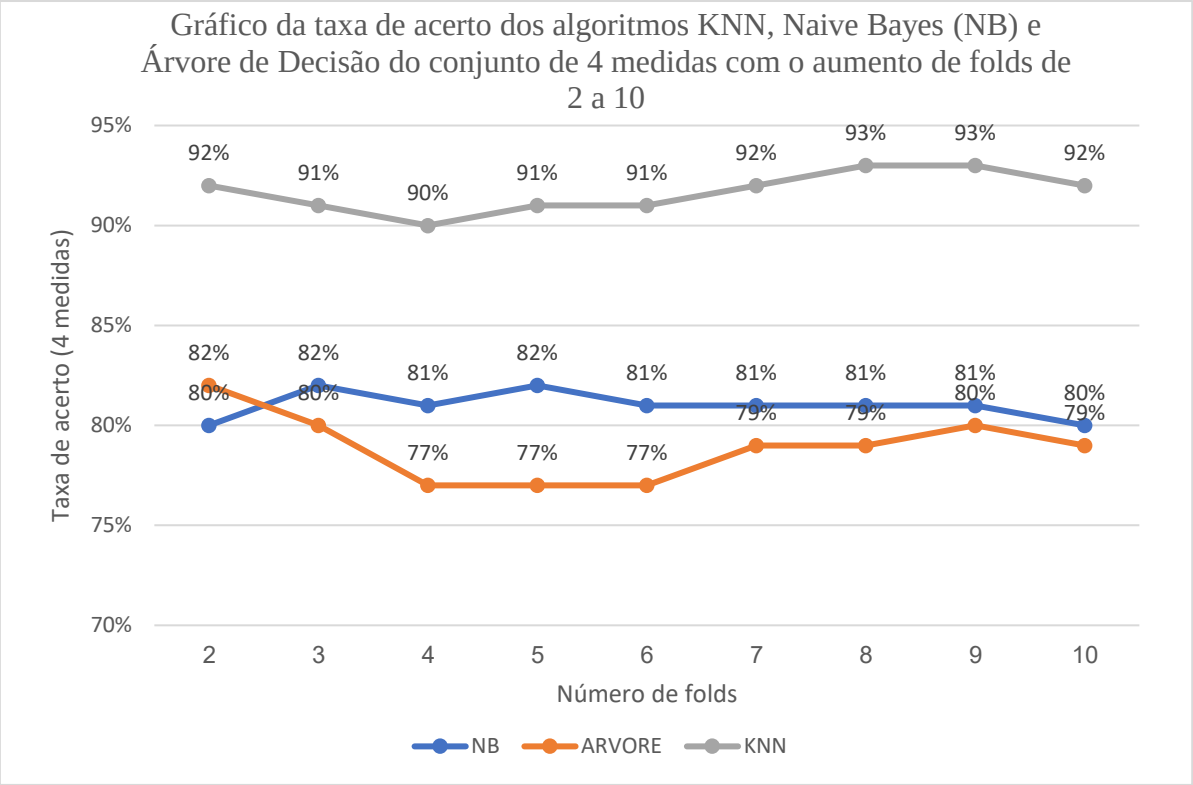


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 4 medidas com o aumento de folds de 2 a 10
Taxa de acerto (4 medidas)
95%
90%
85%
80%
75%
70%
92% 91% 90% 91% 91% 92% 93% 93% 92%
82% 82% 81% 82% 81% 81% 81% 81% 80%
80% 80% 77% 77% 77% 79% 79% 80% 79%
2 3 4 5 6 7 8 9 10
Número de folds
NB
ARVORE
KNN

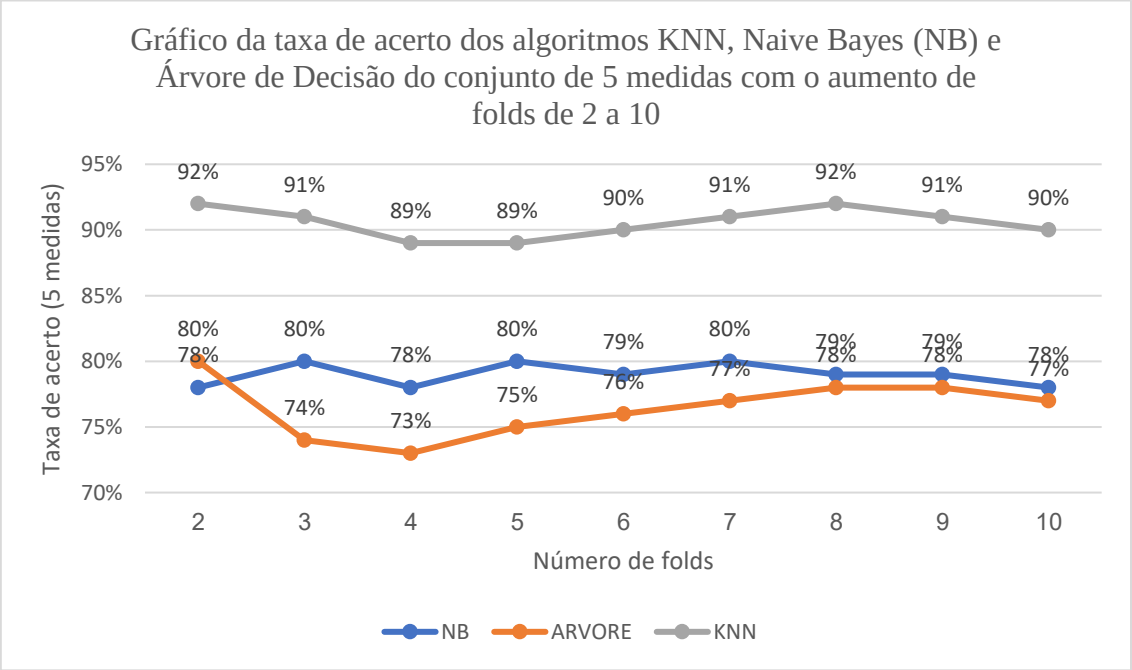


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 5 medidas com o aumento de folds de 2 a 10
Taxa de acerto (5 medidas)
95%
90%
85%
80%
75%
70%
92% 91% 89% 89% 90% 91% 92% 91% 90%
80% 80% 78% 80% 79% 80% 79% 79% 78%
78% 74% 73% 75% 76% 77% 78% 78% 77%
2 3 4 5 6 7 8 9 10
Número de folds
NB
ARVORE
KNN

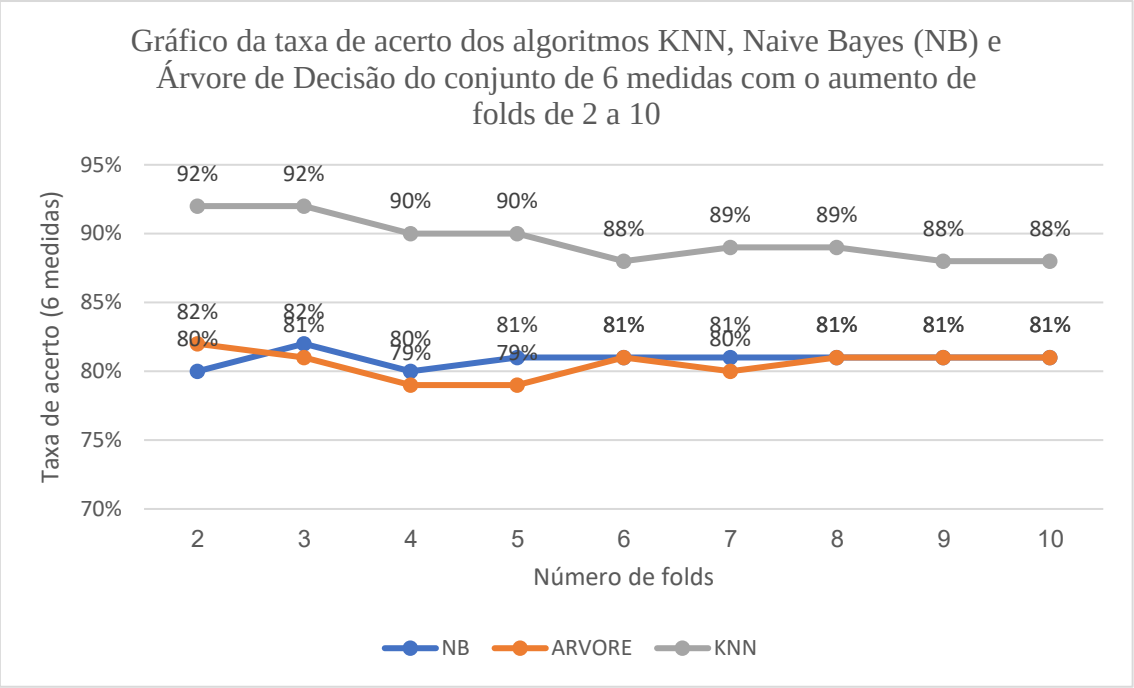


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 6 medidas com o aumento de folds de 2 a 10
Taxa de acerto (6 medidas)
95%
90%
85%
80%
75%
70%
92% 92% 90% 90% 88% 89% 89% 88% 88%
82% 81% 80% 81% 81% 81% 81% 81% 81%
80% 82% 79% 80% 81% 80% 81% 81% 81%
2 3 4 5 6 7 8 9 10
Número de folds
NB
ARVORE
KNN

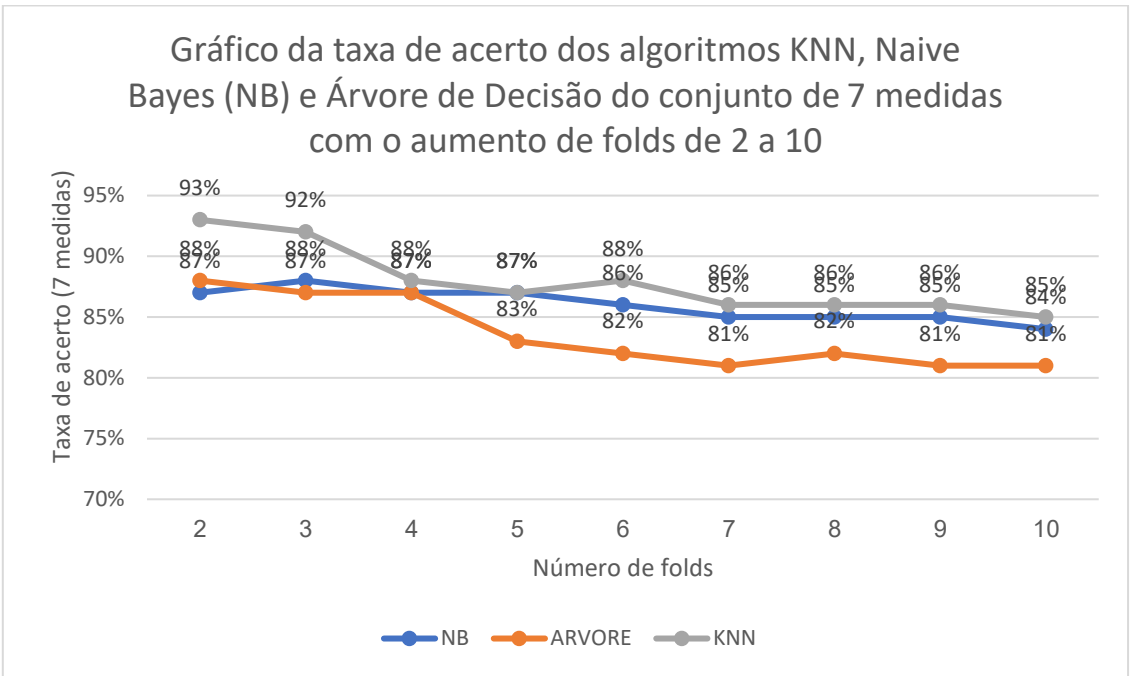


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 7 medidas com o aumento de folds de 2 a 10
Taxa de acerto (7 medidas)
95%
90%
85%
80%
75%
70%
93% 92% 88% 87% 88% 87% 88% 87% 87% 88% 86% 86% 85% 86% 85% 86% 85% 85% 84%
87% 88% 87% 87% 83% 82% 81% 82% 81% 81%
2 3 4 5 6 7 8 9 10
Número de folds
NB
ARVORE
KNN

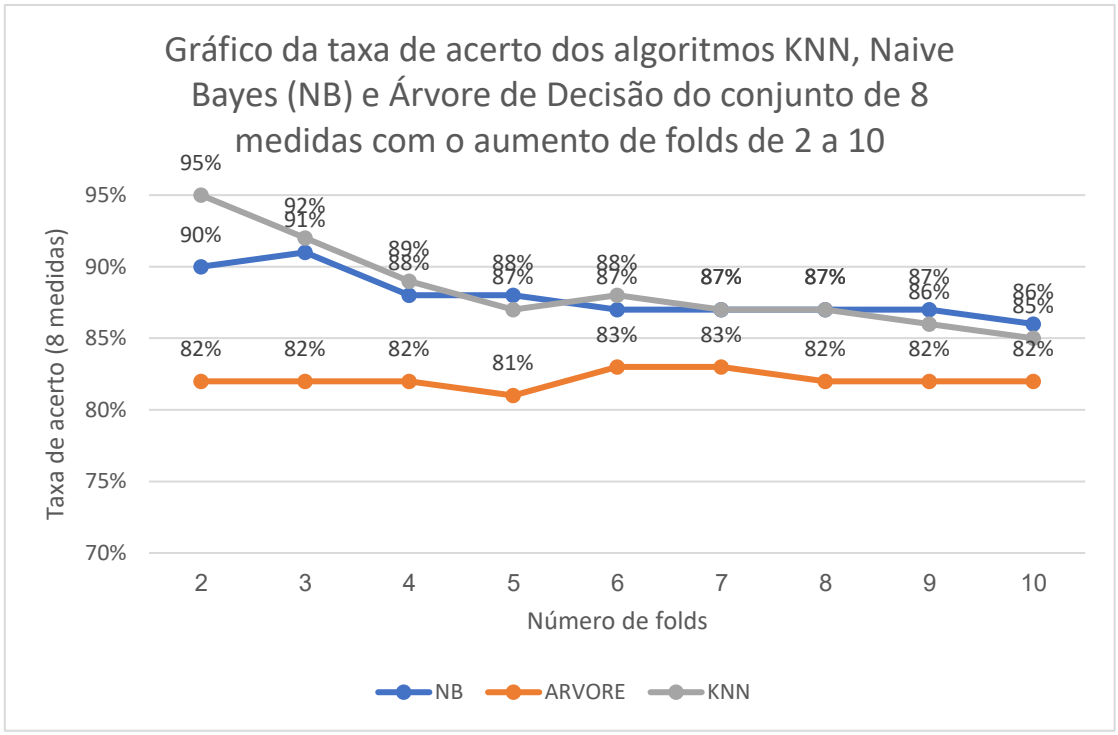


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 8 medidas com o aumento de folds de 2 a 10
Taxa de acerto (8 medidas)
95%
90%
85%
80%
75%
70%
2
3
4
5
6
7
8
9
10
Número de folds
NB
ARVORE
KNN

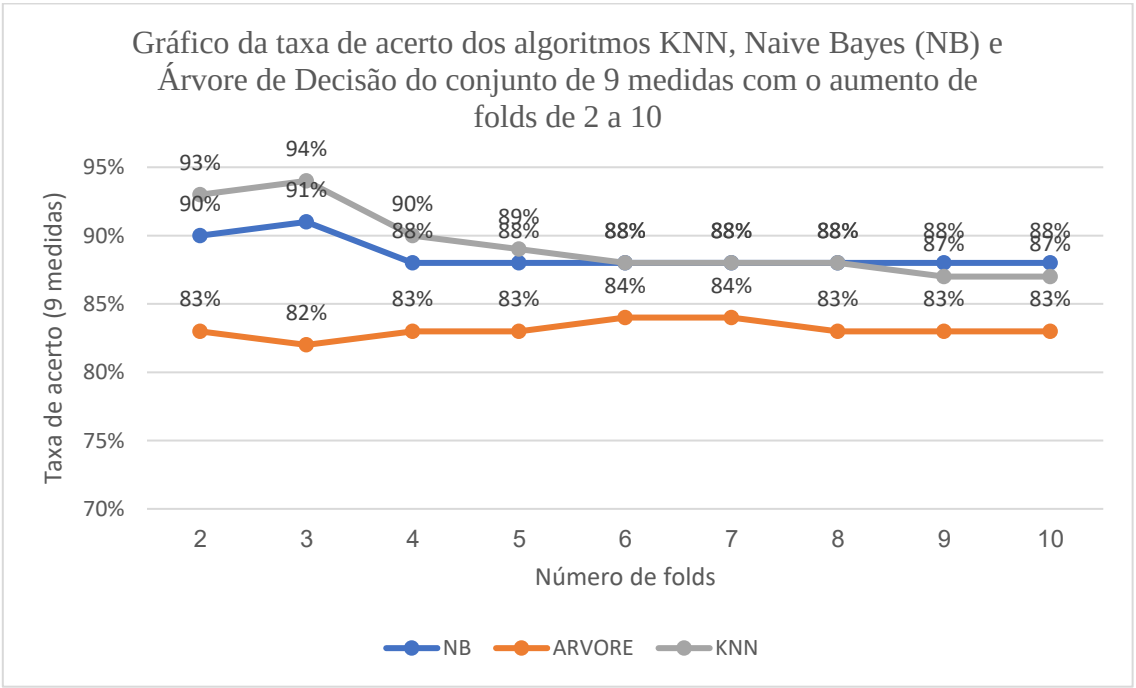


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 9 medidas com o aumento de folds de 2 a 10
Taxa de acerto (9 medidas)
95%
90%
85%
80%
75%
70%
2
3
4
5
6
7
8
9
10
Número de folds
NB
ARVORE
KNN

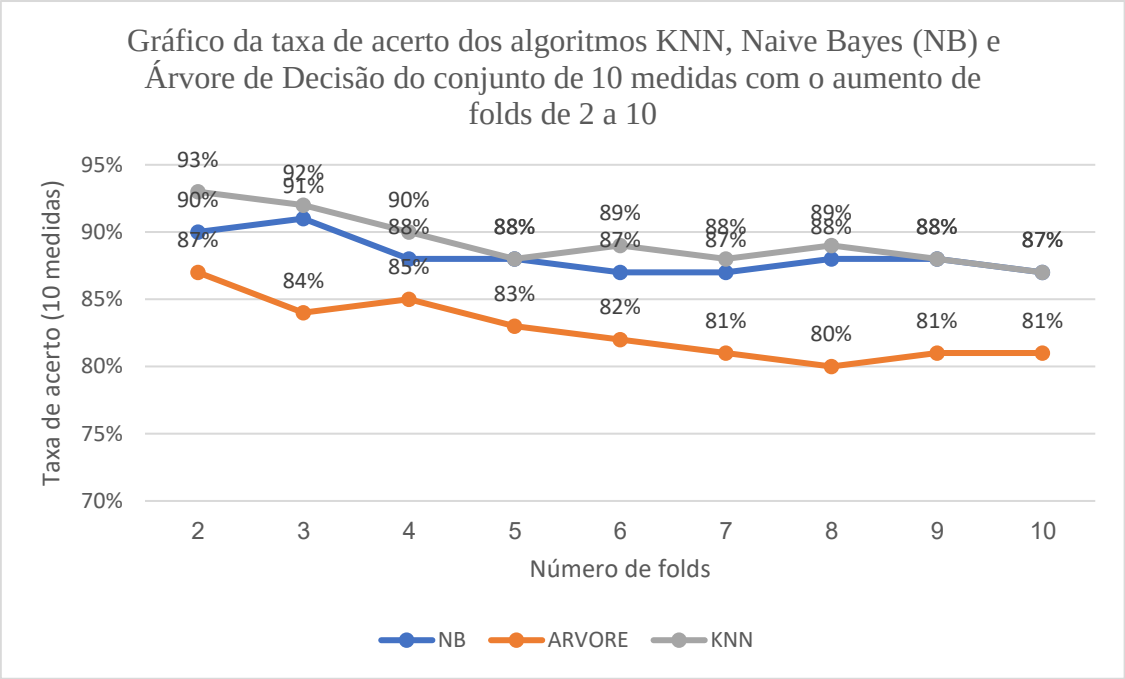


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 10 medidas com o aumento de folds de 2 a 10
Taxa de acerto (10 medidas)
95%
90%
85%
80%
75%
70%
2
3
4
5
6
7
8
9
10
Número de folds
NB
ARVORE
KNN

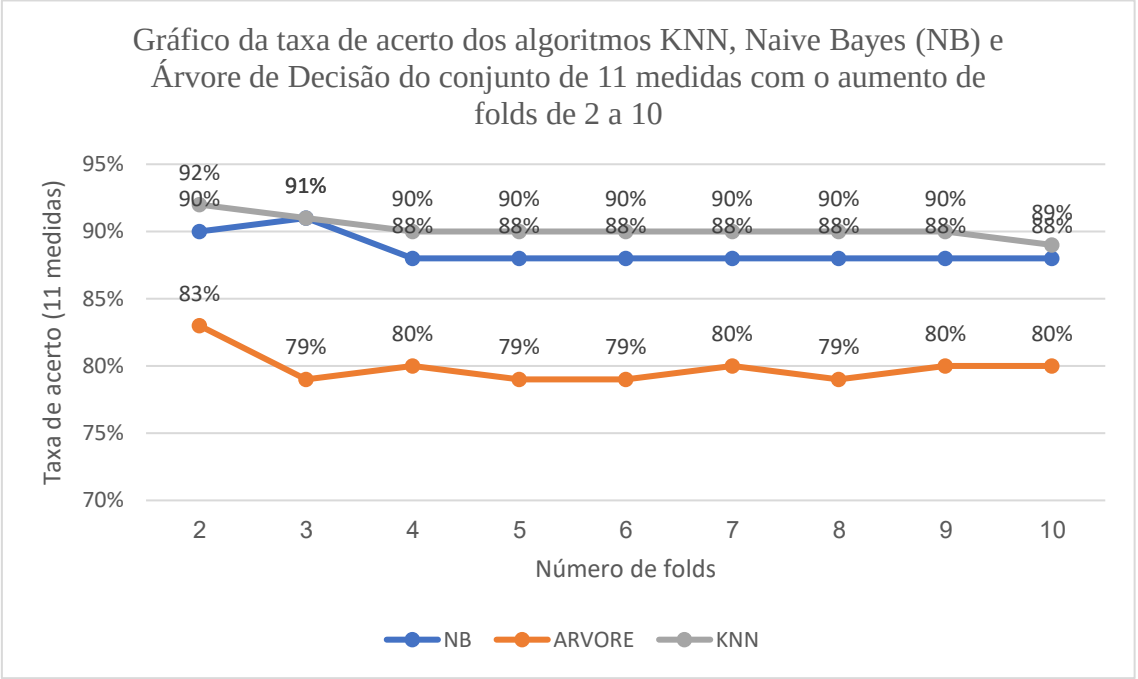


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 11 medidas com o aumento de folds de 2 a 10
Taxa de acerto (11 medidas)
95%
90%
85%
80%
75%
70%
2
3
4
5
6
7
8
9
10
Número de folds
NB
ARVORE
KNN

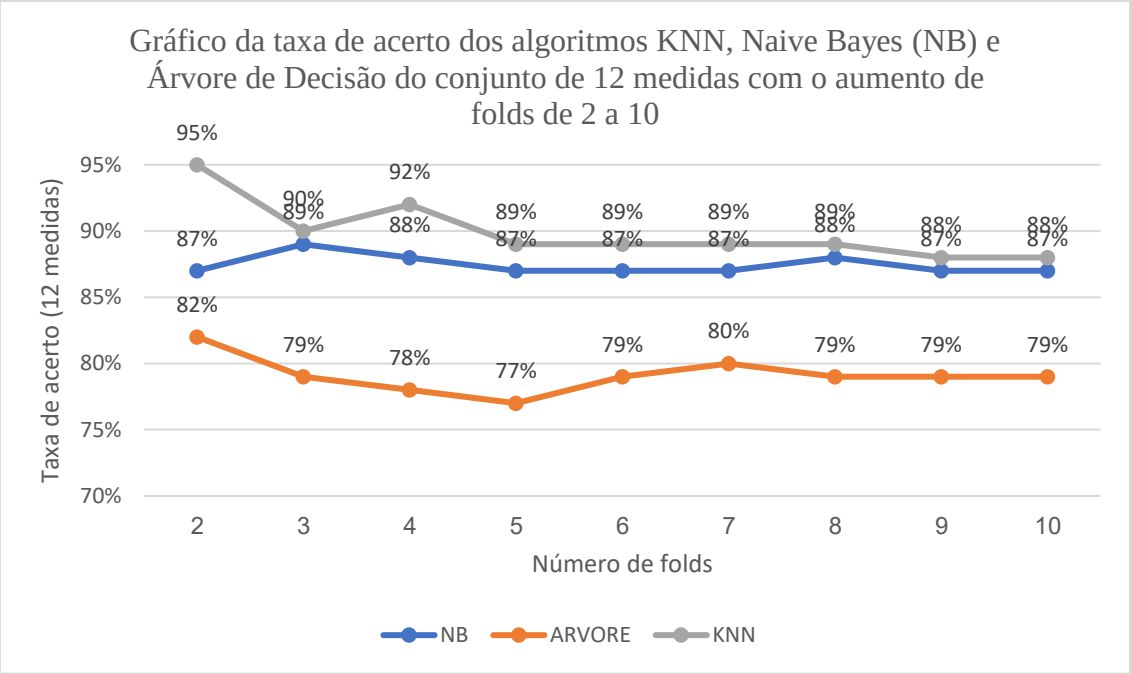


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 12 medidas com o aumento de folds de 2 a 10
Taxa de acerto (12 medidas)
95%
90%
85%
80%
75%
70%
2
3
4
5
6
7
8
9
10
Número de folds
NB
ARVORE
KNN

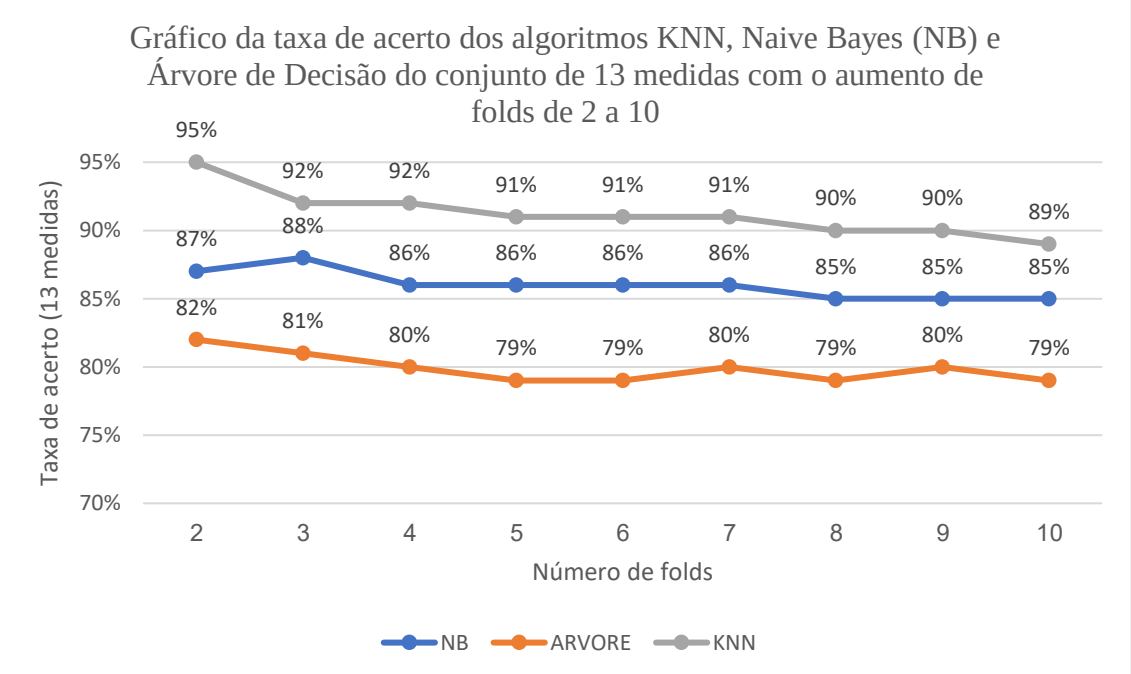


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 13 medidas com o aumento de folds de 2 a 10
Taxa de acerto (13 medidas)
95%
90%
85%
80%
75%
70%
2
3
4
5
6
7
8
9
10
Número de folds
NB
ARVORE
KNN

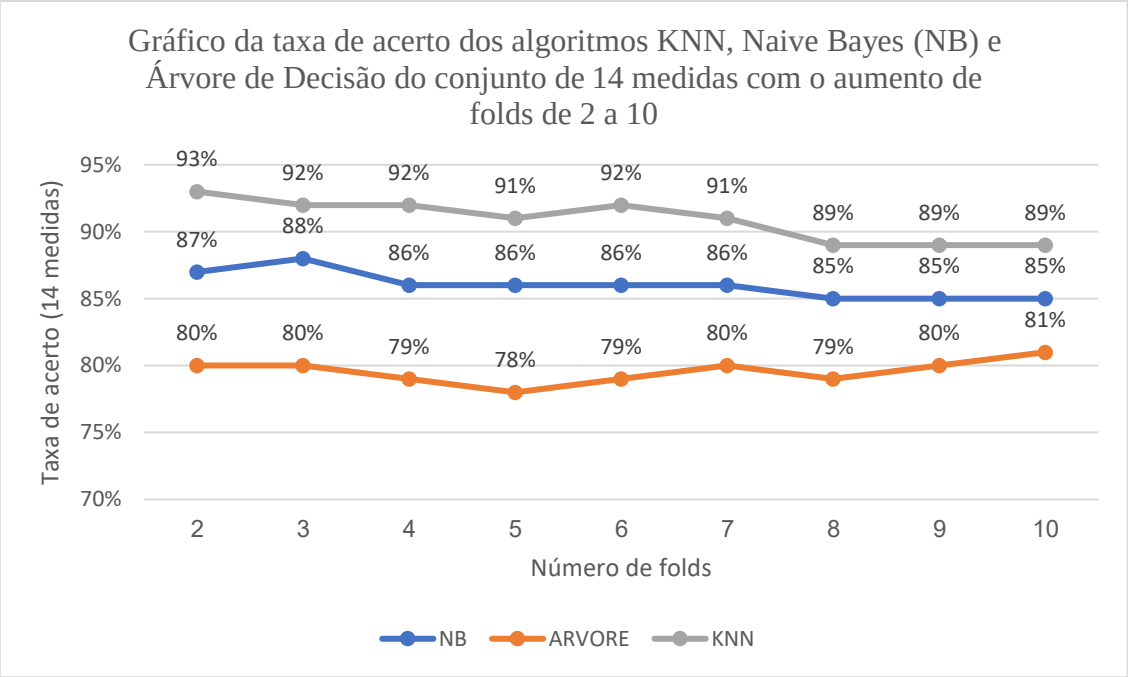


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 14 medidas com o aumento de folds de 2 a 10
Taxa de acerto (14 medidas)
95%
90%
85%
80%
75%
70%
93% 92% 92% 91% 92% 91% 89% 89% 89%
87% 88% 86% 86% 86% 86% 85% 85% 85%
80% 80% 79% 78% 79% 80% 79% 80% 81%
2 3 4 5 6 7 8 9 10
Número de folds
NB
ARVORE
KNN

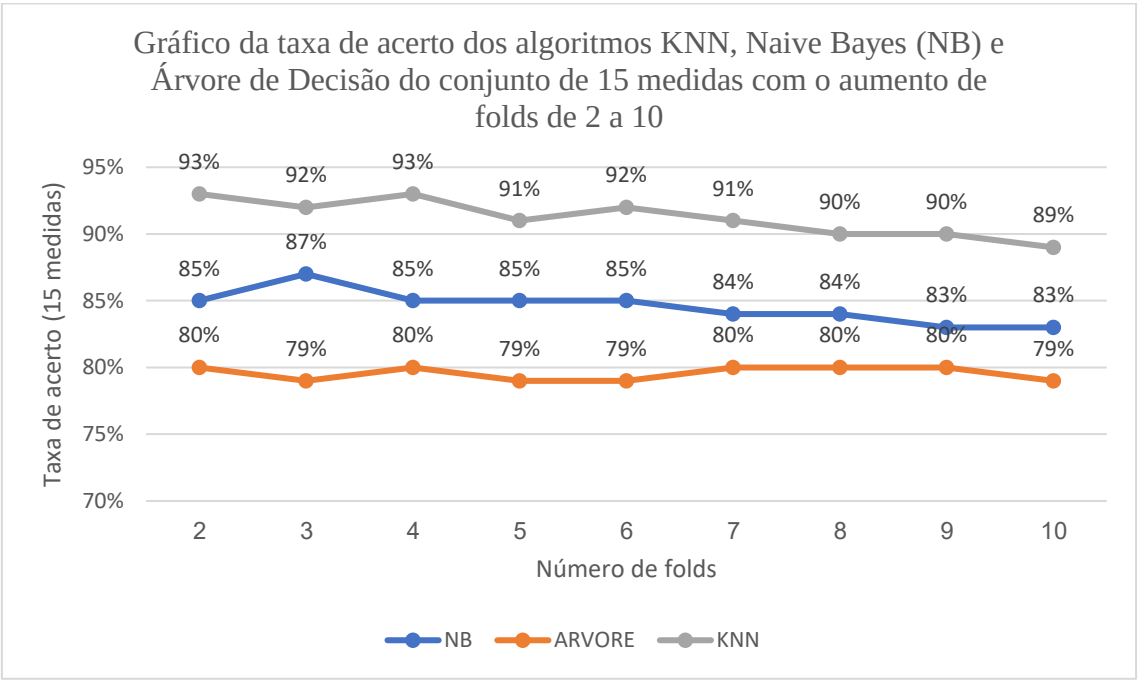


Gráfico da taxa de acerto dos algoritmos KNN, Naive Bayes (NB) e Árvore de Decisão do conjunto de 15 medidas com o aumento de folds de 2 a 10
Taxa de acerto (15 medidas)
95%
90%
85%
80%
75%
70%
93% 92% 93% 91% 92% 91% 90% 90% 89%
85% 87% 85% 85% 85% 84% 84% 83% 83%
80% 79% 80% 79% 79% 80% 80% 80% 79%
2 3 4 5 6 7 8 9 10
Número de folds
NB
ARVORE
KNN