Determinação de idades de abalones utilizando algoritmos de aprendizagem supervisionada

Renan Augusto Leonel
Departamento de Informática
Universidade Estadual de Maringá
Maringá, Paraná, Brasil
ra115138@uem.br

***Abstract*—This work aimed to evaluate the accuracy and mean absolute error of various classification and regression algorithms for a database containing various physical characteristics of abalones. The algorithm that obtained the best accuracy for classifying instances was the Random Forest, and the algorithm that obtained the lowest mean absolute error in predicting the age of abalones was the SVM algorithm.**

I. Introdução

Este trabalho consiste na análise de uma base de dados que contém características físicas de diversos abalones, onde a partir disso, serão utilizados diversos algoritmos para classificar as idades em 3 grupos, sendo eles jovem, médio e velho, e também realizar uma previsão da idade de abalones utilizando os mesmos algoritmos de forma regressiva. Os algoritmos escolhidos foram: K-Nearest Neighbours (KNN), Decision Tree, Random Forest e Support Vector Machine (SVM).

Ao final, foi apresentada uma tabela mostrando a acurácia de cada algoritmo, e outra contendo seu MAE, evidenciando qual obteve o melhor resultado. Este trabalho foi desenvolvido a partir dos passos representados na imagem abaixo.

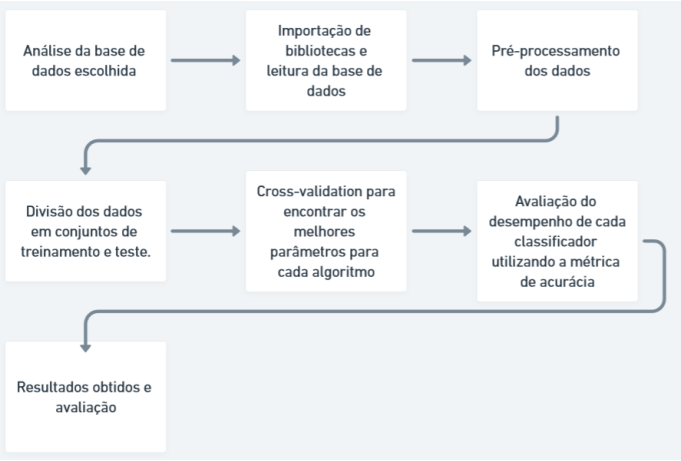

Fig. 1. Sequência de passos.

II. Base de dados

A base de dados escolhida pode ser encontrada em [2], e consiste em diversos atributos físicos de abalones. A idade de abalones pode ser identificada ao retirar sua concha e contar a quantidade de anéis utilizando um microscópio, processo este de extrema lentidão para ser realizado. Sendo assim, a partir das informações provenientes da base de dados escolhida, podemos realizar uma previsão da idade de abalones sem que haja a necessidade de realizar o processo manual descrito anteriormente.

A base de dados escolhida foi extraída a partir de um estudo [3] que não possui relação com a área de inteligência artificial e aprendizagem de máquina, o qual foi direcionado a obter informações qualitativas sobre a situação financeira da pesca de abalones, onde também foi documentado diversas características físicas da população de abalones a partir de análises laboratoriais conduzidas após a obtenção dos moluscos por mergulhadores na costa da Tasmânia.

Sendo assim, os dados resultantes da pesquisa foram categorizados em um arquivo, onde cada linha representa um abalone e suas características físicas. As entradas do arquivo abalone.data estão divididas como mostra a tabela abaixo, onde as unidades de comprimento estão em milímetros, e as unidades de peso estão em gramas.

Sexo	Comprimento	Diâmetro	Altura
M	0.455	0.365	0.095
Peso Total	P. sem concha	P. vísceras	P. concha
0.514	0.2245	0.101	0.15
Anéis			
15			

A base de dados consta em sua totalidade com 4177 entradas, cada uma contendo 8 atributos nos moldes especificados acima.

III. Fundamentação Teórica

Para este trabalho, foram utilizados os algoritmos de aprendizagem supervisionada KNN, Decision Tree, Random Forest e SVM, com o intuito de analisar a acurácia e MAE obtidos para cada um deles e, após a determinação do melhor algoritmo para a base de dados escolhida, analisar quais os

atributos mais importantes da base para a determinação da idade dos abalones.

O método k-Nearest-Neighbours (KNN) é um método amplamente utilizado em problemas de classificação, por se tratar de um método simples e eficiente. É um algoritmo classificador de aprendizagem supervisionada não paramétrico, que utiliza de conceitos como proximidade para agrupar em uma mesma classe as instâncias de dados analisadas, classificando-as a partir do agrupamento resultante [4].

Para problemas de classificação, é atribuída a etiqueta de classe mais frequentemente representada em torno de um determinado ponto de dados, ou seja, o algoritmo calcula a distância entre uma nova amostra e todas as amostras existentes no conjunto de treinamento, selecionando os K vizinhos mais próximos, onde em seguida, a classe da nova amostra é atribuída com base na classe mais comum entre os K vizinhos mais próximos. Para problemas de regressão, um método similar ao de classificação é utilizado, porém, para este tipo de problema, a média dos K vizinhos mais próximos é utilizada para realizar uma previsão sobre uma classificação. Sendo assim, é utilizado classificação para valores discretos, e regressão para valores contínuos [4].

Podemos citar como principais vantagens de se utilizar uma abordagem com kNN a facilidade de implementação e adaptabilidade, além da baixa quantidade de parâmetros necessários para sua execução. Porém, pode-se observar algumas desvantagens em seu uso como uma maior facilidade de gerar overfitting. Além disso, o kNN sofre com problemas de escalabilidade, podendo não ser indicado para bases de dados com um número muito alto de instâncias.

Para este trabalho, foi utilizado o kNN classificador para classificar os exemplares de abalone em um dos 3 grupos de idade previamente definidos. A figura abaixo exemplifica o funcionamento do algoritmo descrito:

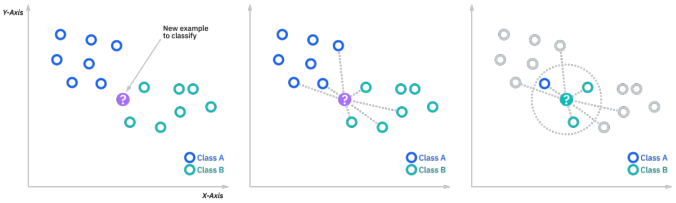

Fig. 2. kNN.

O método de árvore de decisão (Decision Tree) é um algoritmo de aprendizagem supervisionada não paramétrico, que também é utilizado para problemas de classificação e regressão [5]. O algoritmo consiste na criação de uma árvore de decisão com base nas características dos dados de treinamento. A árvore é criada com base em um conjunto de regras, que ajudam a classificar as amostras em uma das várias classes, onde as decisões são tomadas em cada nó da árvore, que representa uma característica ou atributo dos dados.

A aprendizagem da árvore de decisão utiliza a estratégia dividir e conquistar, utilizando uma abordagem gulosa para identificar os melhores pontos de divisão dentro da árvore. Sendo assim, esse processo de divisão é repetido de maneira recursiva até que todos ou a maioria dos registros tenham sido classificados em rótulos de classe específicos. É um método indicado para árvores pequenas, pois para árvores maiores existe um risco maior de ocasionar overfitting [5]. A figura abaixo exemplifica o funcionamento do algoritmo descrito.

Podemos citar como principais vantagens de se utilizar uma abordagem com árvore de decisão sua facilidade de interpretação, ao permitir representações visuais da árvore gerada, e também sua flexibilidade por ser adequada para problemas de classificação e regressão. Porém, pode-se observar algumas desvantagens em seu uso como uma maior facilidade de gerar overfitting e seu custo elevado, resultante da abordagem gulosa proposta pelo algoritmo. Além disso, a árvore de decisão gera uma variância elevada, pois pequenas variações nos dados utilizados pode gerar uma árvore completamente diferente, resultando em uma variação de acurácia.

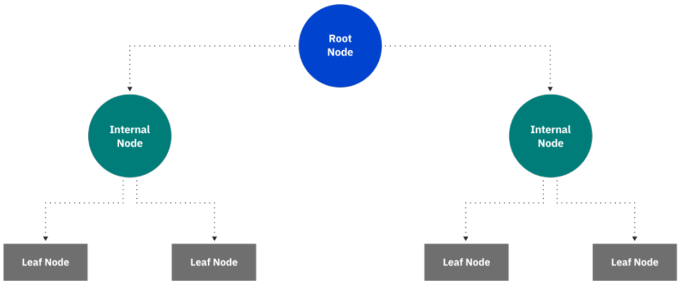

Fig. 3. Decision Tree.

O método Random Forest combina o resultado de diversas execuções de algoritmos de árvores de decisão para obter um único resultado, onde também é utilizado para problemas de classificação e regressão. Este método contribui positivamente para contornar as desvantagens do método de árvore de decisão, como overfitting e viés da árvore, pois a análise de várias árvores em conjunto obtém resultados mais precisos, principalmente quando as árvores não estão correlacionadas entre si. [6]

Este método possui diversos benefícios, como flexibilidade e redução do risco de overfitting por utilizar árvores não correlacionadas entre si, que auxilia na diminuição do erro de predição [6]. Além disso, este método facilita a determinação de importância de cada característica analisada, vantagem esta que será demonstrada ao final deste trabalho. Porém, podemos citar como uma desvantagem de se utilizar uma abordagem com Random Forest o tempo necessário para processar os

dados, resultante do tempo necessário para computar cada árvore de decisão individualmente. A figura abaixo exemplifica o funcionamento do algoritmo descrito:

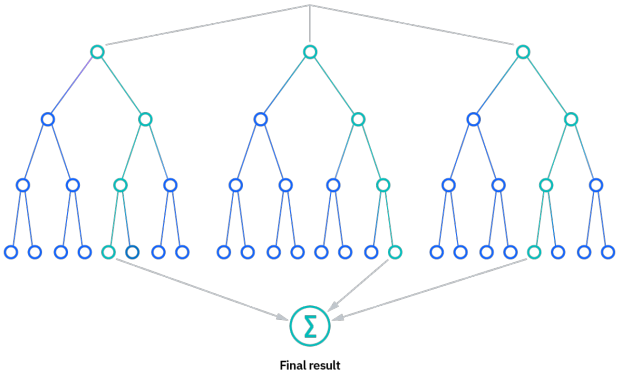

Fig. 4. Random Forest.

Por fim, temos o método Support Vector Machine (SVM), que consiste em uma técnica utilizada para problemas de classificação e regressão. O algoritmo maximiza a predição preditiva de um modelo, sem resultar em overfitting dos dados de treinamento, sendo extremamente indicado para bases de dados com uma grande quantidade de instâncias [7]. O algoritmo consiste em buscar uma reta (também chamada de hiperplano) que separa as amostras de diferentes classes. O algoritmo procura maximizar a margem entre as classes, ou seja, a distância entre o hiperplano e as amostras de cada classe.

A partir desta reta, pode-se prever em qual grupo uma nova instância deve pertencer, dada sua proximidade à reta. Para realizar a transformação, pode-se utilizar as funções matemáticas linear, polinomial, RBF e sigmóide, onde para este trabalho, foram todas consideradas de forma a obter a melhor função para a base de dados escolhida. A figura abaixo exemplifica o funcionamento do algoritmo descrito.

A utilização de uma abordagem com SVM é benéfica para base de dados com uma pequena quantidade de instâncias, onde também se beneficia da generalização do algoritmo, que classifica de maneira adequada novas entradas nunca vistas. Além disso, também é um algoritmo flexível, sendo indicado para problemas de classificação e regressão. Porém, podemos citar como desvantagem o fato de o algoritmo não ser indicado para bases de dados com muitas instâncias, pois pode ser extremamente custoso. Além disso, o algoritmo SVM é extremamente sensível aos parâmetros utilizados, podendo ser uma tarefa complexa determinar os parâmetros ótimos para o problema escolhido.

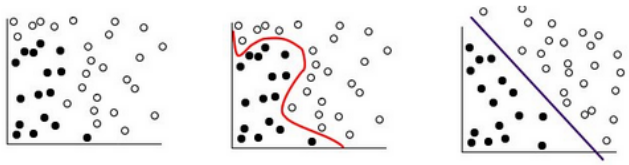

Fig. 5. SVM.

IV. Desenvolvimento

A implementação dos algoritmos para a base de dados deste trabalho foi realizada utilizando como referência o código contido em [1], utilizando a bilbioteca scikit-learn para usufruir dos métodos existentes para a separação dos dados em treino/teste, juntamente com os classificadores para todos os algoritmos utilizados. Para a plotagem e visualização dos gráficos, foram utilizadas as bilbiotecas matplotlib e seaborn. Foram realizados diversos testes para cada algoritmo, alterando os parâmetros

Primeiramente, foi realizada a leitura da base de dados contida em "abalone.data" utilizando a biblioteca pandas, seguida de um pré-processamento para transformar os dados categóricos em binários, onde também foi realizada uma divisão das instâncias em 3 grupos de idade, sendo eles jovem, médio e velho. A partir desta divisão, verifica-se se a base de dados possui algum problema no que diz respeito ao balanceamento de dados, onde pode-se observar que a distribuição é adequada conforme mostra a Figura 6.

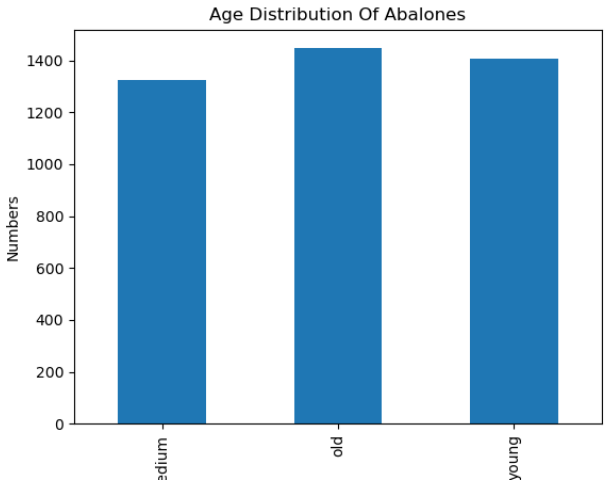

Fig. 6. Avaliação da distribuição dos dados.

Após o processamento, foi realizada uma divisão dos dados em dados de treino e teste, seguindo o padrão 80/20 (80% dos dados para treino, 20% dos dados para teste). A partir desta divisão, foi utilizada a ferramenta GridSearchCV proveniente da biblioteca scikit-learn para realizar uma validação cruzada para cada algoritmo de classificação, com o objetivo de encontrar os melhores parâmetros para cada classificador (por

exemplo, um valor k ótimo para o KNN), que estão disponíveis na Tabela III. Esses valores ótimos foram utilizados tanto para a avaliação de classificação, quanto para a avaliação regressiva. Foram utilizados diversos valores para o parâmetro cv, que determina a estratégia de divisão da validação cruzada, realizando testes com os valores 3, 5 (default), 10 e 15. Após isso, foi analisada a acurácia obtida para cada um dos algoritmos, com o intuito de analisar qual o valor x-fold resulta em uma maior acurácia. Sendo assim, o resultado dessa busca é a combinação de parâmetros que obteve a melhor acurácia de classificação nos dados de treinamento.

Por fim, também foi utilizada a métrica de erro médio absoluto (MAE), que é uma medida de avaliação de desempenho de modelos de regressão que representa a média das diferenças absolutas entre as previsões do modelo e os valores reais [8]. Os resultados dos testes podem ser observados na seção seguinte.

V. Resultados obtidos

Primeiramente, podemos observar os resultados obtidos a partir da validação cruzada utilizando 3-fold, 5-fold, 10-fold e 15-fold, juntamente com os valores dos parâmetros utilizados. Os parâmetros foram obtidos utilizando o método de cross-validation descrito na seção IV, onde após sucessivas execuções, foram encontrados os valores apresentados acima como sendo os melhores parâmetros para a classificação da base. Conforme apresentado na Tabela I, o algoritmo que obteve a maior acurácia para todos os valores de k-fold testados foi o algoritmo Random Forest, onde pode-se observar um aumento na acurácia de todos os algoritmos conforme o número de folds crescia. Além disso, o algoritmo que obteve o menor MAE para todos os valores de k-fold testados foi o algoritmo SVM, conforme mostra a Tabela II.

Podemos concluir que todos os algoritmos obtiveram uma acurácia superior a 60%, onde o algoritmo Random Forest gerou o melhor resultado entre eles com aproximadamente 66% de acurácia. Podemos observar também que utilizar 15-folds resultou em uma maior acurácia, chegando próximo de 67%, onde conclui-se que a acurácia dos algoritmos aumenta conforme o número de folds cresce. Vale a pena ressaltar também que o valor k para o kNN variou conforme o número de folds, assim como o parâmetro ”n-estimators” para o Random Forest. Após isso, avalia-se a acurácia do algoritmo para os dados de teste utilizando o algoritmo que melhor performou, onde obtivemos aproximadamente 63%, evidenciando que não existem problemas de overfitting para este caso e validando os resultados obtidos.

	3-fold	5-fold	10-fold	15-fold
KNN	0.6414	0.6452	0.6466	0.6485
Decision Tree	0.6321	0.6276	0.6257	0.6347
Random Forest	0.6624	0.6635	0.6680	0.6687
SVM	0.6579	0.6616	0.6650	0.6657

TABLE I
Acurácia dos algoritmos utilizando diferentes k-folds.

	3-fold	5-fold	10-fold	15-fold
KNN	1.4564	1.4553	1.4545	1.4896
Decision Tree	1.5457	1.5738	1.5759	1.5759
Random Forest	1.5268	1.4835	1.4679	1.4540
SVM	1.4464	1.4148	1.4152	1.4152

TABLE II
MAE dos algoritmos utilizando diferentes k-folds.

Além disso, podemos analisar quais características físicas de um abalone são mais importante para determinar sua idade. Para isso, os dados de treino e teste foram normalizados, com o intuito de utilizá-los em conjunto com o algoritmo Random Forest, algoritmo este que obteve a maior acurácia dentre os testados, utilizando os parâmetros ótimos obtidos anteriormente. Por fim, utilizando funções auxiliares da biblioteca pandas, podemos gerar um gráfico de barras com as principais características que auxiliam a determinar a idade de um abalone para cada valor de k-fold testados, como mostram as Figuras 7, 8, 9 e 10.

Escolhemos o gráfico de barras para 15-fold, representado na Figura 10, pois foi o parâmetro que obteve a maior acurácia. A partir disso, podemos concluir que os principais atributos físicos de um abalone para a determinação de sua idade estão diretamente relacionados ao seu peso, onde o peso da concha é o principal atributo entre eles. Além disso, as medidas físicas como altura, comprimento e diâmetro podem ser classificadas com grau de importância intermediário, e o atributo ”Sexo” pode ser considerado como o menos determinante, pois ficaram nas últimas posições.

Além disso, as Figuras 11, 12, 13 e 14 apresentam os valores para as idades reais e previstas dos abalones, apresentando os valores previstos por cada algoritmo para todas as variações de k-fold testadas.

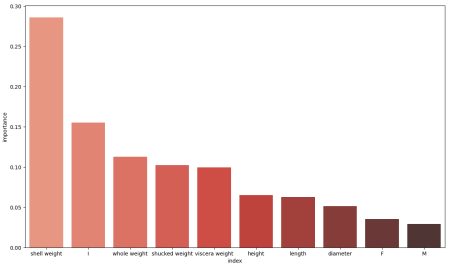

Fig. 7. Atributos mais importantes para 3-fold

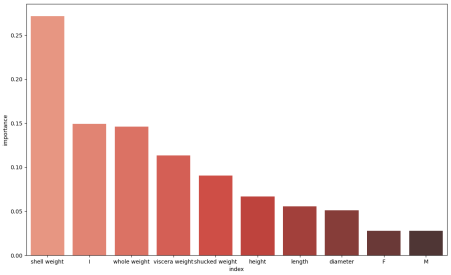

Fig. 8. Atributos mais importantes para 5-fold

Algoritmo	Parâmetros			
	3-fold	5-fold	10-fold	15-fold
KNN	n-neighbours: 17	n-neighbours: 21	n-neighbours: 20	n-neighbours: 30
Decision Tree	max-depth: 5	max-depth: 6	max-depth: 6	max-depth: 6
Random Forest	max-depth: 6	max-depth: 8	max-depth: 10	max-depth: 10
	max-features: 'sqrt'	max-features: 'sqrt'	max-features: 'sqrt'	max-features: 'sqrt'
	n-estimators: 40	n-estimators: 90	n-estimators: 20	n-estimators: 50
SVM	C: 10	C: 10	C: 10	C: 10
	gamma: 2	gamma: 2	gamma: 1	gamma: 1
	kernel: 'poly'	kernel: 'rbf'	kernel: 'rbf'	kernel: 'rbf'

TABLE III
ALGORITMOS E MELHORES PARÂMETROS ENCONTRADOS PARA CADA K-FOLD.

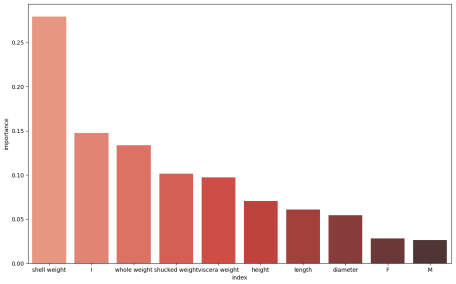

Fig. 9. Atributos mais importantes para 10-fold

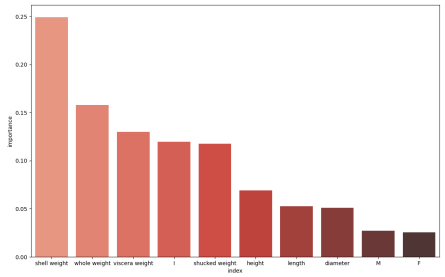

Fig. 10. Atributos mais importantes para 15-fold

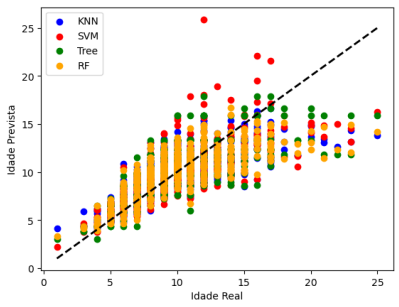

Fig. 11. Previsão da idade utilizando 3-folds

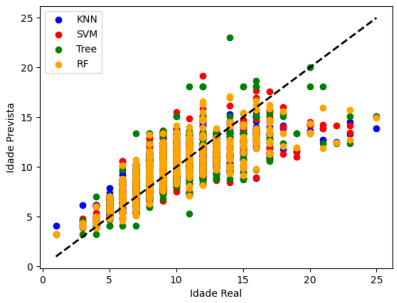

Fig. 12. Previsão da idade utilizando 5-folds

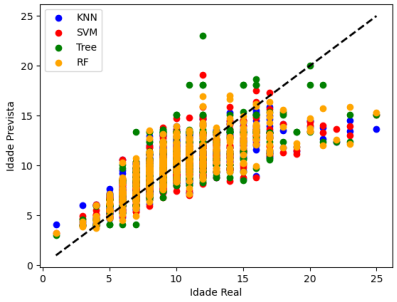

Fig. 13. Previsão da idade utilizando 10-folds

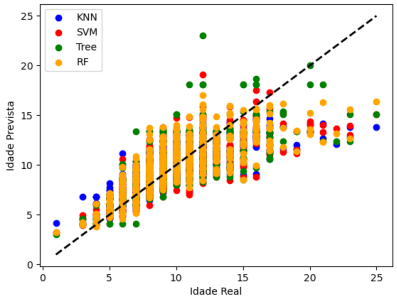

Fig. 14. Previsão da idade utilizando 15-folds

Com base no que foi apresentado, podemos concluir que, para a base de dados escolhida, o algoritmo que melhor classifica os dados é o algoritmo Random Forest com uma acurácia de aproximadamente 67% para 15-folds, onde o atributo mais importante para a classificação é o peso da concha de um abalone. Além disso, o algoritmo que realiza uma melhor previsão da idades dos abalones é o SVM, pois possui o menor MAE de aproximadamente 1.414 para 5-folds.

Por fim, como trabalhos futuros, pode-se realizar mais testes aumentanto os valores de k-fold para cada algoritmo, além de realizar avaliações combinando os algoritmos apresentados neste trabalho com o objetivo de aumentar a acurácia e diminuir o erro absoluto médio dos resultados observados.

REFERENCES

[1] https://github.com/huiminren/AbalonesAgeClassification
[2] https://archive.ics.uci.edu/ml/datasets/abalone
[3] NASH, Warwick J. et al. The population biology of abalone (haliotis species) in tasmania. i. blacklip abalone (h. rubra) from the north coast and islands of bass strait. Sea Fisheries Division, Technical Report, v. 48, p. p411, 1994.
[4] https://www.ibm.com/topics/knn
[5] https://www.ibm.com/in-en/topics/decision-trees
[6] https://www.ibm.com/topics/random-forest
[7] https://www.ibm.com/docs/en/spss-modeler/saas?topic=models-about-svm
[8] https://www.ibm.com/docs/en/cloud-paks/cp-data/3.0.1?topic=overview-mean-absolute-error