Comparação entre métodos de classificação aplicados à classes de preços de venda de imóveis

Comparação entre métodos de classificação de aprendizado de máquina com o objetivo de classificar o valor de imóveis dadas suas características

Gustavo Ficher Catarino
IGCE
Universidade Estadual Paulista
Rio Claro, Brazil
gustavo.ficher@unesp.br

Filipe Antonio Constantino Mouro
IGCE
Universidade Estadual Paulista
Rio Claro, Brazil
filipe.mouro@unesp.br

Denis Henrique Pinheiro Salvadeo
IGCE
Universidade Estadual Paulista
Rio Claro, Brazil
denis.salvadeo@unesp.br

***Resumo* — Existem vários métodos de classificação em aprendizado de máquina disponíveis na academia. Este artigo compara quatro desses métodos usando um banco de dados contendo características e preços de imóveis. Cada método possui suas próprias características que determinam a precisão de acerto em dados após o treinamento.**

Keywords* — *machine learning, classification, real estate, random forest classifier, k-nearest neighbors, naive bayes, multilayer perceptron

I. Introdução

Este artigo tem como objetivo analisar diferentes algoritmos de classificação usando técnicas de aprendizado de máquina, que é uma subciência de inteligência artificial.

Para tanto, foi levado em conta um dataset que contém algumas informações de mercado imobiliário, que é um dos setores mais importantes da economia. A acessibilidade dos preços dos imóveis e dos aluguéis e as mudanças nesses preços têm um impacto direto sobre a riqueza dos proprietários e inquilinos e sobre os gastos dos consumidores.

Com isso em mente, foram selecionados quatro métodos de classificação de forma que eles não sejam muito parecidos entre si, que agrupam imóveis de acordo com a faixa de preço de venda. Os métodos selecionados foram Random Forest Classifier, K-Nearest Neighbors, Naïve Bayes e Multilayer Perceptron nos quais foram programados para classificar os imóveis em custo baixo, médio, alto e altíssimo.

II. Introdução aos métodos de classificação empregados

A. *Random Forest Classifier (RFC):*

O RFC é um classificador ensemble que constrói diversas árvores de decisão para mitigar o problema de overfitting. Cada uma delas produz um resultado independente, mas o conjunto gera um resultado melhor que uma única árvore. Para obter o resultado final, considera-se a moda do conjunto de resultados [2].

B. *K-Nearest Neighbors (K-NN):*

O K-NN baseia seu aprendizado na similaridade de um dado de teste aos dados de treino. O algoritmo mede a distância do dado de teste em relação aos dados de treino e decide a classe resultante com base nos k vizinhos mais próximos. Neste método, a forma de medida de distância depende do problema e é importante encontrar um valor k mais próximo possível do ideal (geralmente baixo) e a influência dos k vizinhos pode ser ponderada. [3]

C. *Naïve Bayes:*

Este método encontra a distribuição de probabilidade de cada valor de um atributo ser de uma determinada classe e, cruzando tais informações, escolhe para o dado teste a classe que é mais provável a pertencer. [4]

D. *Multilayer Perceptron (MLP):*

É uma classe de rede neural que consiste em pelo menos três camadas (Entrada, Saída e camadas ocultas), treinado comumente com um algoritmo de retropropagação. O treino ocorre quando os pesos dos conectores entre os neurônios se altera após cada dado ser processado. [5]

III. Descrição da base de dados

A base de dados escolhida chama-se Melbourne Housing Snapshot[1], contém diversas informações sobre imóveis vendidos na cidade de Melbourne (EUA), tais como preço, localização, ano de construção, distância do centro da cidade e área de construção. A base foi disponibilizada publicamente a fim de apoiar uma competição de classificação no site Kaggle.[6]

IV. Metodologia utilizada

Antes de treinar os modelos de classificação, o dataset passou por um pré-processamento para eliminar campos não relacionados ao objetivo final. Além disso, algumas colunas

foram transformadas para representar melhor o valor contido.

Algumas das alterações foram:

- A partir do atributo ‘Price’, criou-se 4 classes que representam diferentes faixas de valor: baixo, médio, alto e altíssimo. Essas serão as classes utilizadas pelos algoritmos

- considerados desnecessários para os aprendizados foram removidos: 'Address', 'Method', 'Lattitude', 'Longtitude', 'Date' e 'Seller', além de 'Price', que não pode ser visto pelos algoritmos.

- Os atributos categóricos foram convertidos em variáveis indicativas.

Inicialmente, mais atributos estavam sendo retirados, mas a ação foi desfeita porque percebeu-se uma diminuição na acurácia de alguns métodos. Também foram realizados testes com a normalização dos dados com o objetivo de melhorar a precisão dos métodos (principalmente o K-NN), mas não foi atingida melhoria nos resultados.

A divisão da base de dados foi feita com o método holdout com 25% do dataset sendo usado para testes e 75% usado para treinamento.

Da biblioteca scikit learn, foram utilizados os classificadores Random Forest Classifier, K-Neighbors Classifier e GaussianNB, enquanto que o multilayer Perceptron foi implementado com uma junção de métodos da biblioteca keras e scikit learn.

Para análise e comparação dos classificadores, utilizou-se a matriz de confusão, o valor de precisão e uma comparação de curvas ROC.

V. Configurações experimentais

O método random forest é utilizado com 100 florestas internas e com estado randômico fixo (7) para padronizar a saída dentro de múltiplas execuções

O método K-NN está utilizando k=3. tal valor foi decidido após uma análise de diferentes valores.

Foi utilizada a implementação gaussiana do método Naïve Bayes.

O Perceptron tem 339 neurônios de entrada, 8 internos com ativação ReLU e 4 saídas (referentes às 4 classes) com ativação softmax. Sua função de perda é a entropia cruzada categórica com otimizador adam e métrica de precisão. Outras configurações de neurônios internos foram testadas, mas houve grande aumento no tempo de execução sem acrescentar mais precisão.

VI. Resultados gerados

O método de classificação Random Forest Classifier apresentou a maior precisão de aproximadamente 75.27%, mas reconhece-se que este valor não é o suficiente para confiar no modelo. Para a análise, foram geradas as matrizes de confusão e a matriz de confusão normalizada.

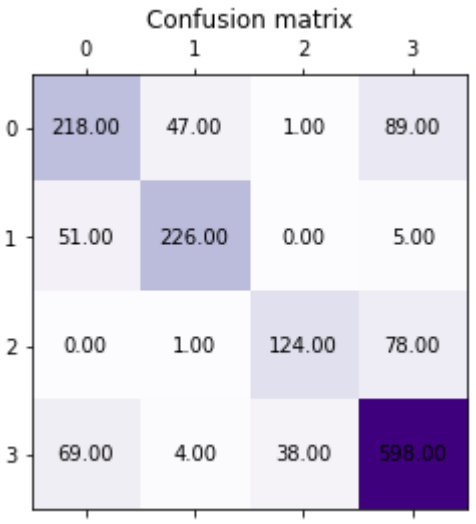

Fig. 1. Matriz de confusão de Random Forest Classifier

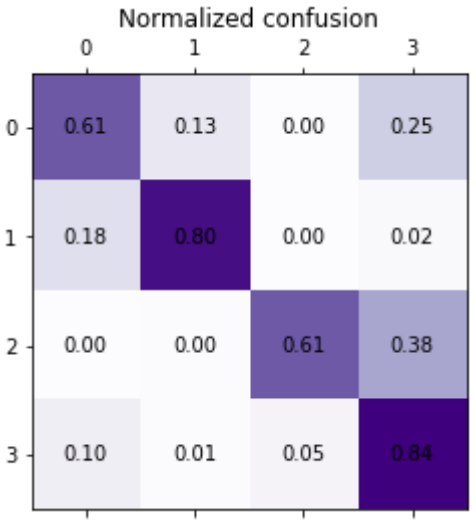

Fig. 2. Matriz de confusão normalizada de Random Forest Classifier

O método K-Nearest Neighbors ficou em segundo lugar porque teve precisão de 65.59% e as matrizes de confusão e a matriz de confusão normalizada geradas podem ser visualizadas a seguir

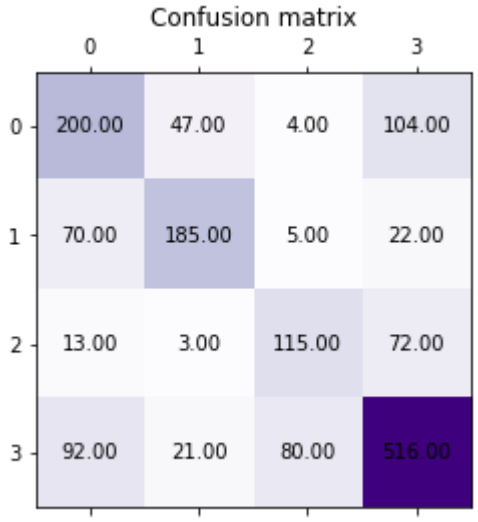

Fig. 3. Matriz de confusão de K-Nearest Neighbors

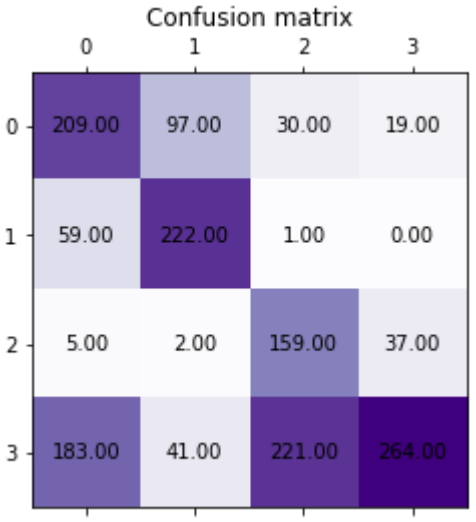

Fig. 5. Matriz de confusão de Naïve Bayes

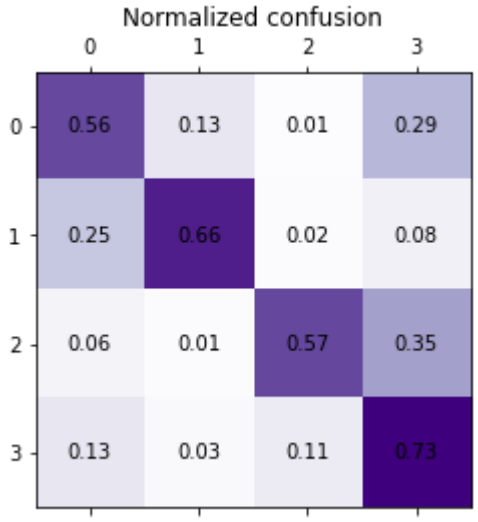

Fig. 4. Matriz de confusão normalizada de K-Nearest Neighbors

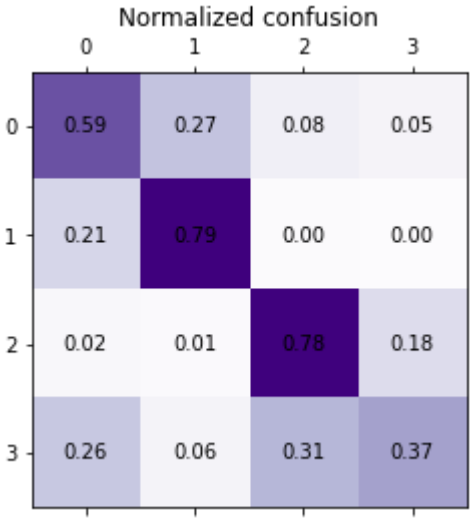

Fig. 6. Matriz de confusão normalizada de Naïve Bayes

O método Multilayer Perceptron ficou em terceiro lugar com a precisão de 57.56. As matrizes de confusão e a matriz de confusão normalizada não puderam ser geradas porque o esse método usa uma rede neural de três camadas e não foi possível encontrar uma forma de obter esses resultados usando as matrizes.

Por fim, o método Naïve Bayes teve a menor precisão de 55.13% entre todos os quatros métodos. As matrizes de confusão e a matriz de confusão normalizada geradas podem ser visualizadas a seguir

VII. Conclusão

A partir dos resultados gerados pelos classificadores, pode-se comparar a facilidade de obter um nível razoável de precisão em cada um deles. Também observou-se que cada método possui suas próprias características que devem ser ajustadas dependendo do tipo dos dados de entrada e que a quantidade de dados de treinamento também varia. Alguns métodos demandam maior poder computacional e diminuem a facilidade de testar diferentes configurações.

References

[1] D. Becker, Melbourne Housing Snapshot. https://www.kaggle.com/dansbecker/melbourne-housing-snapshot, 2018.

[2] T. Yiu, Understanding Random Forest. https://towardsdatascience.com/understanding-random-forest-58381e0602d2, junho 2019

[3] D. Salvadeo, Classificação: Parte 1. UNESP, Mar 2021, pp.52–71.

[4] D. Salvadeo, Classificação: Parte 2. UNESP, Mar 2021, pp.4–30.

[5] D. Salvadeo, Classificação: Parte 2. UNESP, Mar 2021, pp.52–71.

[6] J. Brownlee "Multi-Class Classification Tutorial with the Keras Deep Learning Library," in Magnetism, vol. III, https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/, jan 2021.