Documentação do Projeto de Inteligência Artificial

Técnicas de Machine Learning para inferência de uma variável target.

Faculdade e Campus: USJT - Butantã

Andrew Augusto Matos Silva – RA: 822150527

Eduarda de Oliveira – RA: 822129744

João Henrique Pacheco de Oliveira – RA: 822154533

João Pedro Deiroz Fecchio – RA: 822222470

Thiago Rodrigues Marinho – RA: 822146121

A base de dados escolhida se chama “Zoo Animal Classification”, possui dados referentes às características de diversos animais e de diferentes classes. Nessa base de dados as variáveis referentes às características dos animais são importantes para o que será feito futuramente, treinar o modelo de machine learning para prever a variável target (classe do animal). Esta base de dados é encontrada no site Kaggle (em https://www.kaggle.com/).

Base de dados escolhida (link do site Kaggle): https://www.kaggle.com/datasets/uciml/zoo-animal-classification.

Base de dados escolhida (link do Planilhas Google): https://docs.google.com/spreadsheets/d/1SHVTNV3VTI7wN47hi8G2Ks8b2XzS0PM2/edit?usp=sharing&ouid=113224702602120506490&rtpof=true&sd=true.

Tabela complementar para o projeto (link do Planilhas Google): https://docs.google.com/spreadsheets/d/1JQ_YdqSvacAWNBksdSLIsgZ-n9HVnrsz/edit?usp=sharing&ouid=113224702602120506490&rtpof=true&sd=true.

Na base de dados “DataSetAnimaisCaracteristicas” (segundo link acima) é apresentado os nomes dos animais na primeira coluna, nas demais colunas/variáveis estão as características dos animais e na última coluna é apresentado a respectiva classe de cada animal.

Detalhando cada coluna/variável:

1. **animal_name:** Variável do tipo String que identifica o nome do animal, esta variável será ignorada para o treinamento do modelo, pois nosso modelo utilizará diferentes técnicas que irão analisar as características representadas por valores numéricos para prever a classe do animal, que também será representada por um valor numérico. Além disso, não queremos que nosso sistema de machine learning descubra de qual classe o animal é com base em seu nome, e sim com base em suas características biológicas/físicas.
2. **hair:** Variável do tipo int (inteiros) que identifica se o animal tem pelos ou cabelos, que será representada por 0 (não tem cabelo e/ou pelo) ou 1 (tem cabelo ou pelo).
3. **feathers:** Variável do tipo int (inteiros) que identifica se o animal tem penas, que será representada por 0 (não tem penas) ou 1 (tem penas).
4. **eggs:** Variável do tipo int (inteiros) que identifica se o animal bota ovo, que será representada por 0 (não bota ovo) ou 1 (bota ovo).
5. **milk:** Variável do tipo int (inteiros) que identifica se o animal amamenta, que será representada por 0 (não amamenta) ou 1 (amamenta).
6. **airborne:** Variável do tipo int (inteiros) que identifica se o animal voa, que será representada por 0 (não voa) ou 1 (voa).
7. **aquatic:** Variável do tipo int (inteiros) que identifica se o animal é aquático, que será representada por 0 (não é aquático) ou 1 (é aquático).
8. **predator:** Variável do tipo int (inteiros) que identifica se o animal é predador, que será representada por 0 (não é predador) ou 1 (é predador).
9. **toothed:** Variável do tipo int (inteiros) que identifica se o animal possui dentes, que será representada por 0 (não possui dentes) ou 1 (possui dentes).
10. **backbone:** Variável do tipo int (inteiros) que identifica se o animal tem coluna vertebral, que será representada por 0 (não tem coluna vertebral) ou 1 (tem coluna vertebral).

11. **breathes:** Variável do tipo int (inteiros) que identifica se o animal faz respiração branquial, que será representada por 0 (faz respiração branquial) ou 1 (não faz respiração branquial).
12. **venomous:** Variável do tipo int (inteiros) que identifica se o animal é venenoso, que será representada por 0 (não é venenoso) ou 1 (é venenoso).
13. **fins:** Variável do tipo int (inteiros) que identifica se o animal tem barbatanas ou nadadeiras, que será representada por 0 (não tem barbatanas ou nadadeiras) ou 1 (tem barbatanas ou nadadeiras).

14. **legs:** Variável do tipo int (inteiros) que identifica quantas pernas tem o animal, na qual os números 0, 2, 4, 5, 6 ou 8 representam a quantidade de pernas do animal.
15. **tail:** Variável do tipo int (inteiros) que identifica se o animal tem cauda, que será representada por 0 (não tem cauda) ou 1 (tem cauda).
16. **domestic:** Variável do tipo int (inteiros) que identifica se o animal é doméstico, que será representada por 0 (não é doméstico) ou 1 (é doméstico).
17. **catsize:** Variável do tipo int (inteiros) que identifica se o animal é maior que o tamanho de um gato, que será representada por 0 (não é maior que um gato) ou 1 (é maior que um gato).
18. **class_type:** Essa é a variável target, ou seja, nessa coluna estão os valores que nosso modelo de machine learning irá tentar prever. Nessa coluna/variável estarão valores do tipo int (inteiros) que vão de 1 a 7 (no qual cada número representa uma classe dos animais).

Detalhando a variável target (class_type):

Na tabela complementar do projeto (link mencionado acima) é explicado as informações sobre as classes utilizadas nessa base de dados. Nela está presente uma coluna com os números que cada classe representa (importante para entender a coluna class_type da base de dados escolhida), sendo 1 (os mamíferos), 2 (as aves), 3 (os repteis), 4 (os peixes), 5 (os anfíbios), 6 (os insetos), 7 (os invertebrados). Também há uma coluna indicando o número de

espécies em cada classe (Number_Of_Animal_Species_In_Class) e uma coluna com todos os nomes de animais pertencentes à uma determinada classe (Animal_Names).

Variável Target:

Como já foi mencionado, a variável target será a classe dos animais (na tabela é identificada como “class_type”. O motivo dessa escolha é simples, queremos que a as técnicas de Machine Learning consigam prever de qual classe é o animal de acordo com as suas características (variáveis preditoras).

Variáveis transformadas:

Nenhuma das variáveis da Base de dados precisará ser transformada, pois tanto as variáveis preditoras quanto a variável target são do tipo inteiro, ou seja não apresentam outros tipos de dados que seriam difíceis de manipular (String seria um exemplo). Entretanto a variável dos nomes dos animais será desconsiderada para a aplicação das técnicas de ML, pois não é considerada uma característica essencial para a IA utilizar para prever a classe do animal, e também o tipo dessa variável é String, diferente das demais que é int.

Primeiro método de IA (Árvores de decisão):

Nesta primeira etapa, aplicamos e testamos o método de árvores de decisão ao nosso projeto. Árvores de decisão é muito utilizado em tarefas de classificação, por isso decidimos testar no nosso projeto que visa prever a classe dos animais.

```
[61] #Criando uma matriz de confusão
     print('\nMatriz de confusão detalhada:\n',
           pd.crosstab(y_test, predictions, rownames = ['Real'],
                       colnames = ['Previsto'],
                       margins = True, margins_name = 'Todos'))
```

```
Matriz de confusão detalhada:
 Previsto  1  2  3  4  5  6  7  Todos
Real
1         5  0  0  0  0  0  0      5
2         0  7  0  0  0  0  0      7
3         0  0  1  0  0  0  0      1
4         0  0  0  8  0  0  0      8
5         0  0  0  0  1  0  0      1
6         0  0  0  0  0  4  1      5
7         0  0  0  0  0  0  4      4
Todos     5  7  1  8  1  4  5     31
```

Os resultados obtidos foram satisfatórios, na imagem acima é possível perceber que o modelo previu com 100% de êxito em praticamente todas as classes, com exceção da classe número 7 que representa os invertebrados e teve um acerto de 80%.

```
#Apresentando a avaliação do modelo de machine learning aplicado a esse caso
import sklearn.metrics as metrics
print( 'Relatório sobre a qualidade:\n')
print(metrics.classification_report(y_test, predictions,
                                    target_names = ['Mammal', 'bird',
                                                    'Reptile', 'Fish',
                                                    'Amphibian', 'Bug',
                                                    'Invertebrate']))
```

```
Relatório sobre a qualidade:

              precision    recall  f1-score   support

      Mammal       1.00      1.00      1.00         5
        bird       1.00      1.00      1.00         7
     Reptile       1.00      1.00      1.00         1
        Fish       1.00      1.00      1.00         8
   Amphibian       1.00      1.00      1.00         1
         Bug       1.00      0.80      0.89         5
Invertebrate       0.80      1.00      0.89         4

    accuracy                           0.97        31
   macro avg       0.97      0.97      0.97        31
weighted avg       0.97      0.97      0.97        31
```

A imagem acima contém os resultados, indicando principalmente a precisão de acerto que o modelo obteve. Utilizando o método de IA de árvores de decisão nosso modelo conseguiu prever a variável target muito bem, pois conseguiu

atingir uma média de 97,14% considerando a porcentagem de acertos de cada resultado possível da variável target (classes dos animais).

Segundo método de IA (KNN):

Agora para a segunda etapa, aplicamos e testamos o método KNN (K – Nearest Neighbors) em nosso projeto. Esse é um método que pode ser usado para casos de classificação e de regressão. Esse método de IA se baseia na similaridade de um dado com o outro, analisando a “distância” entre os dados e a quantidade de dados de cada classe. Para o nosso projeto, escolhemos esse método, pois pode ser utilizado para problemas de classificação, e com isso, iremos avaliar seu desempenho na previsão das classes dos animais e comparar com o método anterior.

```
#Criando uma matrix de confusão
print('\nMatrix de confusão detalhada:\n',
      pd.crosstab(y_test, predictions, rownames = ['Real'],
                  colnames = ['Previsto'],
                  margins = True, margins_name = 'Todos'))
```

```
Matrix de confusão detalhada:
 Previsto  1  2  4  5  6  7  Todos
Real
1         5  0  0  0  0  0      5
2         0  7  0  0  0  0      7
3         0  0  1  0  0  0      1
4         0  0  8  0  0  0      8
5         0  0  0  1  0  0      1
6         0  0  0  0  5  0      5
7         0  0  0  1  2  1      4
Todos     5  7  9  2  7  1     31
```

```
[14] #Criando uma matrix de confusão através da função confusion_matrix importada do pacote sklearn.metrics para o código
     confusion_matrix(y_test, predictions)

array([[5, 0, 0, 0, 0, 0, 0],
       [0, 7, 0, 0, 0, 0, 0],
       [0, 0, 0, 1, 0, 0, 0],
       [0, 0, 0, 8, 0, 0, 0],
       [0, 0, 0, 0, 1, 0, 0],
       [0, 0, 0, 0, 0, 5, 0],
       [0, 0, 0, 0, 1, 2, 1]])
```

Os resultados obtidos utilizando o KNN não foram tão satisfatórios, principalmente ao comparar com os resultados utilizando árvores de decisão. O modelo acertou 100% a previsão das classes Mammal, Bird e Invertebrate, porém não acertou nenhuma previsão da classe Reptile, acertou 89% das previsões da classe Fish, 50% da classe Amphibian e 71% da classe Bug.

O KNN errou previsões em 4 classes, enquanto o modelo anterior que utilizou árvores de decisão errou apenas na classe Invertebrate, porém mantendo uma previsão de 80%.

```
#Apresentando a avaliação do método KNN aplicado nessa situação
import sklearn.metrics as metrics
print( 'Relatório sobre a qualidade:\n')
print(metrics.classification_report(y_test, predictions,
                                    target_names = ['Mammal', 'bird',
                                     'Reptile', 'Fish',
                                     'Amphibian', 'Bug',
                                     'Invertebrate']))
```

```
Relatório sobre a qualidade:

              precision    recall  f1-score   support

      Mammal       1.00      1.00      1.00         5
        bird       1.00      1.00      1.00         7
     Reptile       0.00      0.00      0.00         1
        Fish       0.89      1.00      0.94         8
   Amphibian       0.50      1.00      0.67         1
         Bug       0.71      1.00      0.83         5
Invertebrate       1.00      0.25      0.40         4

    accuracy                           0.87        31
   macro avg       0.73      0.75      0.69        31
weighted avg       0.88      0.87      0.84        31
```

Utilizando o método KNN, o modelo parece ter tido dificuldade em prever as classes dos animais, pois 4 classes tiveram uma precisão menor que 100% e 3 classes uma precisão menor que 75%. A média de precisão obtida foi de 72,85%, que comparando com os resultados aplicando o método de Árvores de decisão (97,14%), a precisão média do método KNN foi 24,29% menor, uma diferença considerável.