UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
MAC0460 - Introdução ao aprendizado de máquina

SABRINA ARAÚJO DA SILVA, NUSP 12566182

EP2: Relatório

São Paulo
2023

1. Introdução

O objetivo deste relatório é descrever a análise realizada para resolver o problema de classificação de pokémons dos tipos aquático e normal encontrados durante a expedição do professor Carvalho. O problema consiste em classificar corretamente novos pokémons com base em suas características, contando com a ajuda do Pikachu para identificar vantagens e desvantagens contra pokémons do tipo elétrico. Com base nisso, foram construídos classificadores binários em diferentes modelos, como Regressão Logística (RL), Support Vector Machine (SVM), Decision Tree (DT) e Random Forest (RF), para ajudar a classificação de novos pokémons. Além dos modelos adicionais Perceptron, K-Nearest Neighbors (KNN) e Naive Bayes.

2. Metodologia

O dataset utilizado foi importado do arquivo "pokemon.csv"e passou por algumas etapas de tratamento antes de ser utilizado na análise. Primeiramente, algumas colunas consideradas irrelevantes para a análise ou com valores ausentes foram excluídas do dataset. As colunas "type2", "abilities"e "classfication"foram removidas de acordo com as especificações do enunciado do EP. Além disso, as colunas "is_legendary", "percentage_male", "japanese_name", "name", "weight_kg"e "height_m"também foram excluídas devido a conflitos no código, como valores iguais a zero, dados ausentes ou dados iguais a NaN.

Após a limpeza dos dados, o dataset foi filtrado para manter apenas as linhas com o tipo "normal"ou "water"na coluna "type1". Em seguida, o conjunto de dados foi dividido em duas partes: 75% foi alocado para o conjunto "atual"e 25% para o conjunto "final". O conjunto "final"será usado para testar todos os modelos após o treinamento de todos. O conjunto "atual"foi novamente dividido em 70% para o conjunto de treinamento (X_train e y_train) e 30% para o conjunto de teste (X_test e y_test).

Os modelos utilizados foram Regressão Logística, Support Vector Machine (SVM), Decision Tree e Random Forest. Além dos modelos adicionais Perceptron, K-Nearest Neighbors e Naive Bayes. Cada modelo foi treinado com os conjuntos de treinamento e, em seguida, testado com os conjuntos de teste.

Para cada modelo, foram realizadas buscas de parâmetros utilizando k-cross validation e grid search para encontrar os melhores valores.

Para a Regressão Logística, foi feita a busca para os melhores valores do parâmetro de regularização, com base no parâmetro C. O modelo SVM também passou por uma busca dos melhores valores para os parâmetros C, kernel, gamma e grau do polinômio. O modelo Decision Tree também realizou a busca para encontrar os melhores valores para os parâmetros "maximum depth of the tree"e "minimum number of samples required to be at a leaf node". Por fim, o modelo Random Forest passou pelo mesmo processo de busca para os parâmetros "number of estimators", "maximum depth of the tree"e "minimum number of samples required to be at a leaf node".

Os modelos adicionais tiveram seus parâmetros ajustados, exceto Naive Bayes. Perceptron buscou os melhores valores para "alpha"e "max_iter". KNN buscou os melhores valores para "n_neighbors"e "weights".

Após o treinamento e teste dos modelos, foi utilizado o conjunto de dados separado previamente (X_final e y_final) para testar a acurácia de todos os modelos. Com base na maior acurácia final, foi identificado o melhor modelo.

3. Resultados

Temos os seguintes valores finais de acurácia para cada modelo:

Tabela 1 – Valores Finais de Acurácia dos Modelos

Modelo	**Acurácia Final**
Regressão Logística	0.80
SVM	0.80
Decision Tree	0.85
Random Forest	0.87
Perceptron	0.60
K-Nearest Neighbors	0.75
Naive Bayes	0.75

E as seguintes matrizes de confusão para cada modelo:

- Matriz de confusão para Regressão Logística

  [[23 7]

  [4 21]]

- Matriz de confusão para SVM

[[23 7]

[4 21]]

- Matriz de confusão para Decision Tree

  [[24 6]

  [2 23]]

- Matriz de confusão para Random Forest

  [[24 6]

  [1 24]]

- Matriz de confusão para Perceptron:

  [[15 15]

  [7 18]]

- Matriz de confusão para K-Nearest Neighbors:

  [[20 10]

  [4 21]]

- Matriz de confusão para Naive Bayes:

  [[22 8]

  [6 19]]

Comparando as acurácias finais dos modelos e as matrizes de confusão, podemos concluir que a Random Forest obteve o melhor desempenho em relação aos demais modelos testados. E o modelo Perceptron obteve o pior desempenho entre os modelos.

4. Discussão

Podemos utilizar o atributo "feature_importances_"da Random Forest para obter uma estimativa de quais features foram mais relevantes para a classificação dos pokémons. Com base nisso, foi obtido o seguinte resultado:

É possível observar que a feature "against_electric"se destacou como mais importante na classificação, tendo relação direta com a ajuda do Pikachu durante a expedição. Além de que as seguintes features mais relevantes para a classificação foram as habilidades e estatísticas de batalha dos pokémons.

Os resultados obtidos podem ajudar o professor, uma vez que ele pode escolher o modelo com maior acurácia para obter classificações mais precisas.

Tabela 2 – Importância das Features

Feature	**Importance**
against_electric	0.243598
sp_attack	0.109782
base_total	0.076333
base_egg_steps	0.074573
attack	0.073187
speed	0.065950
sp_defense	0.064077
pokedex_number	0.063630
defense	0.063455
hp	0.054710
experience_growth	0.038081
capture_rate	0.034194
generation	0.030888
base_happiness	0.007543

No entanto, para aprimorar ainda mais a classificação, é preciso que o professor Carvalho colete mais dados, pois quanto mais informações e conhecimento ele tiver disponíveis, maior será a precisão do modelo de classificação.

5. Conclusão

Ao aplicar diferentes modelos de aprendizado de máquina, como Regressão Logística, SVM, Decision Tree, Random Forest, Perceptron, K-Nearest Neighors e Naive Bayes, foi possível identificar diferentes valores de acurácia.

Foi possível perceber que a Random Forest apresentou os melhores resultados, obtendo uma acurácia superior aos demais modelos. Além disso, através da matriz de confusão, foi possível avaliar o desempenho dos modelos na classificação dos Pokémons, identificando os acertos e erros em cada categoria.

A importância das features também foi avaliada por meio da Random Forest, destacando a feature "against_electric"como uma das mais relevantes para a classificação.

Para aprimorar a análise dos Pokémons, seria recomendado coletar um conjunto de dados maior, que inclua uma variedade maior de tipos e características dos Pokémons para que o modelo de aprendizado tenha uma melhor precisão.

Referências bibliográficas

Scikit-learn (2021). Supervised Learning. Disponível em: https://scikit-learn.org/stable/supervis
learning.

Kaggle. Tuning Parameters for Logistic Regression. Disponível em: https://www.kaggle.com/cod
tuning-parameters-for-logistic-regression.