Este projeto é parte do meu portfólio de exercícios de aplicações práticas de ML e IA visando praticar e internalizar conceitos, ferramentas e técnicas vistas em cursos e conteúdo acadêmico

Visando experimentar os principais algoritmos de Machine Learning optei como conjunto de dados a base de jogadores do FIFA, escolha motivada por minha familiaridade com o jogo e paixão por futebol que facilitam a compreensão dos dados em questão, assimilação de conceitos e resolução de problemas

Fiz o carregamento do Dataset de jogadores do FC 24 por meio de um Web Scraper escrito em Java utilizando as bibliotecas HtmlUnit e Commons CSV da Apache para extração dos dados do site SoFIFA e escrita do CSV respectivamente

O projeto responsável por isso pode ser visitado em https://github.com/RenanFR/sofifa-web-scraping

Como primeiro desafio escolhi a classificação da posição do jogador com base em seus atributos e parti para a Engenharia de Features seguindo as etapas abaixo:

* Redução de dimensionalidade agregando as avaliações específicas de cada habilidade em colunas com a média para cada aspecto do jogo (Ataque, habilidade técnica, movimentação, força, mentalidade, defesa e goleiro). Isso, pois, existem 34 atributos distintos para cada valência (Como chute, cruzamento, drible, aceleração, visão, salto, etc.) e a tese inicial é de que as médias por posição podem ser o suficiente para classificar a posição de um jogador

|  |  |
| --- | --- |
| Crossing, Finishing, Heading accuracy, Short passing, Volleys | ATTACKING |
| Dribbling, Curve, FK Accuracy, Long passing, Ball control | SKILL |
| Acceleration, Sprint speed, Agility, Reactions, Balance | MOVEMENT |
| Shot power, Jumping, Stamina, Strength, Long shots | POWER |
| Aggression, Interceptions, Positioning, Vision, Penalties, Composure | MENTALITY |
| Defensive awareness, Standing tackle, Sliding tackle | DEFENDING |
| GK Diving, GK Handling, GK Kicking, GK Positioning, GK Reflexes | GOALKEEPING |

* Remoção de outros atributos sem correlação com a posição como ID
* Como a idade do jogador em atividade pode ser um previsor útil para descobrir a posição (Visto que atletas mais longevos geralmente estarão nas posições de goleiro e zagueiro), transformei a data de nascimento na idade numérica criando uma coluna que subtrai o ano presente nela do atual
* Conversão de atributos financeiros do jogador como salário, valor de mercado e clausula de rescisão de String (Como foram originalmente capturados) para numérico
* Foi necessário realizar a codificação (Encoding) da variável de pé dominante (Destro ou canhoto) usando a estratégia de One-hot

Em seguida executei o treinamento do modelo de árvore de decisão utilizando a biblioteca sklearn

A acurácia obtida foi de apenas 0.51 e a aderência aos dados de treinamento é de 0.92, mostrando a necessidade de refinamento e otimização tanto do pré-processamento quanto dos parâmetros do algoritmo, o que farei em exercícios posteriores

**Problemas detectados**

* O SoFIFA armazena várias versões de sua base de dados de jogadores com base nas atualizações periódicas da EA, na data da extração dos dados foi utilizada como base a versão de 26/09/2023, ao fazer porém, uma comparação das pontuações de ataque do jogador Bukayo Saka usando CSV e site retroagindo a versão para a que foi utilizada há algumas discrepâncias como 82 para cruzamento no CSV, enquanto no site está como 83. Outros atributos e registros foram impactados como por exemplo, a clausula de liberação do mesmo jogador consta como €158M no CSV e €195.5M no site
  + Não se sabe se se trata de um bug na navegação por histórico de versão no site ou se foi algo do lado do código do Web Scraper

Como forma de reduzir a complexidade do problema e otimizar os resultados aceitei a redução da especificidade da classificação por meio da diminuição de possíveis valores para a classe. No caso da classificação de posição dos jogadores ao invés de tentar prever uma dentre as 17 posições específicas, utilizei o agrupamento por função em campo que contém 4 categorias. Aplicando essa generalização simplificamos o problema, apesar de abrir mão da categorização da posição final em específico, ainda nos beneficiamos e extraímos valor ao obter a classificação automática da função do jogador (Tipo de posição) e aumentamos significativamente a acurácia dos algoritmos e modelos testados (Devido à redução do número de classes). Abaixo o mapeamento feito

|  |  |
| --- | --- |
| GK | GOALKEEPER |
| CB, RB, LB, RWB, LWB | DEFENDER |
| CM, CDM, CAM, RM, LM | MIDFIELDER |
| ST, CF, RF, LF, RW, LW | FORWARD |

No python foi criada uma função de mapeamento conforme a tabela acima na etapa de pré-processamento e aplicada ao conjunto de dados antes da divisão entre treinamento e validação

**Etapas comuns de pré-processamento**

* Remoção das colunas não relacionadas
* Exclusão dos não numéricos na coluna número de camisa
* Conversão para numérico das colunas monetárias
* Criação da coluna idade a partir da data de nascimento
* Agrupamento das 17 posições específicas em 4 categorias

**Experimentos**

|  |  |  |  |
| --- | --- | --- | --- |
| **Algoritmo/Modelo** | **Resultado** | **Pré-processamentos** | **Hiper parâmetros** |
| Árvore de decisão | 0.7889978413099563 | * Agrupamento das 34 avaliações específicas de habilidade em suas 7 categorias de valência | {'criterion': 'gini', 'max\_depth': 10, 'min\_samples\_split': 50} |
| KNN | 0.8618421052631579 | * Escalonamento dos atributos numéricos * Loop e gráficos específicos para testar uma faixa de valores distintos para o parâmetro k e aferição visual do melhor número (HPO) | n\_neighbors = 50 |
| SVM | 0.875 | * Escalonamento dos valores numéricos | kernel='poly', C=2.0 |