# A A3

## Inteligência Artificial

***Professor:*** Marcelo Paoli

***Alunos responsáveis:***

Vinicius Pereira de Sousa | RA: 12523163043

Kauan Guilherme dos Santos Silva | RA: 1252229754

Thiago Ferreira da Silva | RA: 1252225451

Gustavo Batista | RA: 12523137227

***Objetivo do trabalho:***

Relatório de um trabalho de análise de dados com 4 diferentes modelos em cima de um dataset que traz informações sobre acidentes de trânsito da Australia, vindo direto do site do governo.

Essa base contém registros de acidentes fatais de trânsito ocorridos na Austrália ao longo de mais de três décadas, com diversas variáveis relevantes como localização, horário, tipo de estrada, envolvimento de veículos, entre outras. Com isso em mente, buscamos explicar o tipo de acidente (***Crash Type***) que tende a acontecer em determinada rua/situação ou transeuntes ali presentes, o retorno é uma categórica que retorna se o acidente envolveu múltiplos indivíduos, ou se foi um acidente sem vítimas.

***Modelos utilizados***:

- Random Forest

- Logistic Regression

- SVM

- KNN (Bônus)

Para avaliação dos modelos, foram usadas as métricas:

* **Acurácia**
* **Precisão**
* **Recall**
* **F1-Score**
* **Matriz de confusão**

Dissertaremos sobre os modelos aplicados, seu desempenho e levantaremos hipóteses sobre os resultados levantados por eles.

***Random Forest***:

Ao aplicarmos o modelo de Random Forest, recebemos o seguinte output:

(Já com os melhores parâmetros ao utilizarmos o GridSearch)

Texto

O conteúdo gerado por IA pode estar incorreto.

- Acurácia: 72% (Nada ruim, porém nada impressionante.)

- F1-Score Médio Ponderado: 0.70

- Desempenho:

- Excelente Recall para Single (0.87)

- Baixo Recall para Multiple (0.52)

Análise:

O modelo Random Forest foi o que retornou um melhor desempenho dentre todos os aplicados. Seu poder de generalização e sua resistência à ruídos poderia ser facilmente um fator determinante para isso. No entanto, nota-se que no cálculo de acidentes múltiplos, possivelmente devido a um desbalanceamento de classes ou sobreposição de características, o modelo não conseguiu ir tão bem, ou achar um padrão.

***Logistic Regression:***

Um modelo mais simples, porém, extremamente funcional na grande maioria das análises.

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

- Acurácia: 71%

- F1-Score médio Ponderado: 0.70

- Desempenho:

- Melhor recall para acidentes do tipo Single (0.79)

- Desempenho moderado para Multiple (F1-score = 0.64)

Análise:

Como um modelo linear, a Regressão Logística tende a ser mais simples e eficaz em problemas com dados linearmente separáveis. Aqui, ela teve um desempenho razoável, mas menor capacidade de capturar relações mais complexas nos dados. O dataset tinha diversas colunas categóricas, onde o modelo de regressão logística poderia brilhar um pouco mais, o que não foi tão longe. Perdeu apenas na precisão para a Random Forest. Mesmo sendo mais simples.

***SVM:***

Tentando dividir em 2 grupos o dataset e traçando uma linha que traça uma distância que melhor os explicam e os separam

*Calendário

O conteúdo gerado por IA pode estar incorreto.*

- Acurácia: 71%

- F1-Score médio Ponderado: 0.70

- Desempenho:

- Recall para Single foi 0.86

- Para Multiple foi apenas 0.53

Análise:

O SVM mandou bem para captar padrões não lineares, porém foi bem mediano para reconhecer padrões Multiple. O SVM tem uma sensibilidade à padrões de regularização e à distribuição de classes, o que pode ter explicado a discrepância de ambos os resultados.

**KNN:**

Como bônus, resolvemos testar como o KNN se sairia nesse dataset

Calendário

O conteúdo gerado por IA pode estar incorreto.

- Acurácia: 71%

- F1-Score médio Ponderado: 0.70

- Desempenho:

- Bom recall para Single (0.85)

- Baixo recall para Multiple (0.51)

Análise:

O output de KNN foi muito parecido com o do SVM e da Regressão Logística, boas previsões da single, porém muita dificuldade com a Multiple.

O que até é esperado, tendo em vista que o KNN é altamente sensível à distribuição dos dados e à presença de ruídos. Se os dados de treino tiverem regiões densas de uma classe (como Single), o modelo tende a favorecer essa classe, pois ela começa a ter um peso estatístico para o modelo.

Além disso, o **KNN sofre em datasets com muitas dimensões** (maldição da dimensionalidade), o que pode ter contribuído para seu desempenho limitado.

# Conclusões

Aplicando esses 4 modelos no dataset de acidentes automobilísticos, podemos, podemos levantar o seguinte sobre os modelos;

Random Forest foi o que teve o melhor resultado, trazendo um ‘equilíbrio’ entre as classes. Tem grande força contra overfitting, que possivelmente foi o fator determinante de todo o resto dos modelos não ter trazido um resultado satisfatório. Seus usos ideais seriam datasets com muitos atributos, ruído ou relações complexas entre variáveis. Aplicações incluem detecção de fraude, sistemas de recomendação e diagnósticos médicos.

Logistic Regression teve uma boa performance, especialmente na single (assim como todos). Seus altos são que sua aplicação e leitura são simples e é facilmente interpretável. Esse modelo não lida bem com decisões não lineares, o que, no caso do nosso dataset, definitivamente não ajudou. Exemplos de melhores usos desse modelo seria em situações com necessidade de explicação clara e relações aproximadamente lineares, como crédito bancário, análises epidemiológicas e classificações binárias básicas.

SVM seguindo o padrão de todos os outros, manteve uma boa acurácia, porém, apanhando para as análises de decisões não lineares.

O SVM funciona bem para situações de alta dimensionalidade e separações complexas, porém, se limita na sensibilidade à escalabilidade de dados e o desbalanceamento das classes, além de poder trazer gargalos em grandes datasets. Idealmente usado em problemas com conjuntos de dados bem estruturados, mas não necessariamente lineares, como reconhecimento de padrões, bioinformática e classificação de texto.

KNN obteve a menor capacidade de identificar a classe Multiple.

É um modelo simples de entender e implementar, porém é também sensível a ruído, escala de variáveis e à maldição de dimensionalidade.

Use em Problemas com poucos atributos, dados bem distribuídos e sem muito ruído. Ex: reconhecimento de dígitos manuscritos ou recomendadores baseados em similaridade.