

Inteligência Artificial

2º trabalho prático

Grupo A2_1

- Francisco da Ana, up202108762
- José Pedro Evans, up202108818
- Tomás Vicente, up202108717

Descrição do problema

- **Objetivo:** Estudar vários modelos de *machine learning* com a finalidade de identificar se um ficheiro pdf é malicioso ou benigno.
- **Dataset:** Dados de 500 000 PDFs (dos quais 90% são malignos).
 - Variáveis independentes (20): tamanho do ficheiro, número de páginas, imagens, cross references, objetos de JavaScript, metadados, ...
 - Variável dependente (a ser prevista pelo modelo): classificação como maligno (0/1)
 - Dataset disponível [aqui](#).



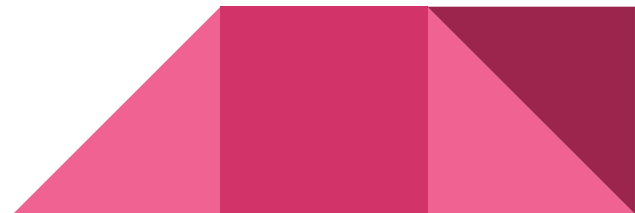
Trabalhos relacionados

- *“Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection”*

Maiorca, D., Corona, I., & Giacinto, G. (2013, May). Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security (pp. 119-130). ([Ligação](#))

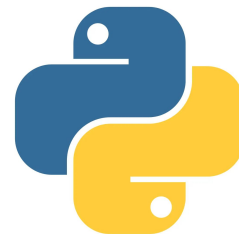
- *“Evasion attacks against machine learning at test time”*

Trad, F., Hussein, A., & Chehab, A. (2023). Leveraging Adversarial Samples for Enhanced Classification of Malicious and Evasive PDF Files. *Applied Sciences*, 13(6), 3472. ([Ligação](#))



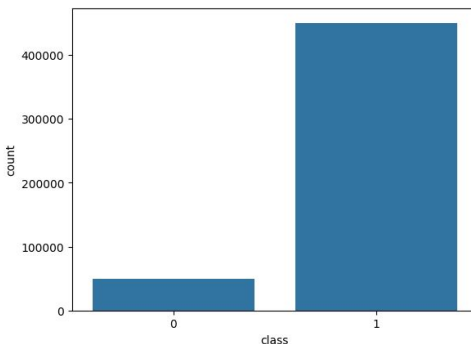
Recursos para implementação

- **Linguagem de programação:** Python (Jupyter Notebook)
- **Bibliotecas:**
 - **Pandas** - representação dos dados
 - **Seaborn** - visualização e estudo estatístico dos dados
 - **Matplotlib** - visualização de dados
 - **Numpy** - operações numéricas
 - **Sklearn** - treino dos modelos de *machine learning* e respetiva avaliação



Pré-processamento dos dados

- No pré processamento dos dados realizámos apenas um **balanceamento** dos mesmos, já que não se registaram *missing values*, *outliers* ou outro tipo de anomalias que requeressem transformações prévias. Desta forma aplicámos SMOTE (*Synthetic Minority Over-sampling Technique*) para balancear os dados.
- Além disso, separámos ainda os dados entre conjunto de teste e de treino.




90% da amostra era positiva

Algoritmos

- **Árvores de Decisão** - Sujeitas a *overfitting*, podem ter mau desempenho quando expostas a novos dados.
- **Random Forest** - Reduz o *overfitting*, mas é computacionalmente mais exigente e podem ser necessários ajustes.
- **Regressão Logística** - Assume relações lineares, sensível a outliers, requer normalização dos dados.
- **K-Nearest Neighbors** - Computacionalmente exigente para grandes conjuntos de dados, sensível ao valor de K, escala dos dados e variáveis irrelevantes.
- **Support Vector Machine** - Seleção complexa de kernel e parâmetros de regularização, computacionalmente caro, menos interpretável.
- **Gradient Boosting** - Propenso a *overfitting* sem regularização adequada, requer ajuste cuidadoso de hiperparâmetros, computacionalmente intensivo para grandes conjuntos de dados.
- **AdaBoost** - Sensível a outliers e dados ruidosos, menos eficaz com dados de alta dimensionalidade, o desempenho depende do classificador base e de ajustes extensivos de hiperparâmetros.
- **Gaussian Naive Bayes** - Assume independência condicional, funciona melhor com variáveis contínuas com distribuição normal, menos eficaz com variáveis categóricas ou relações complexas.

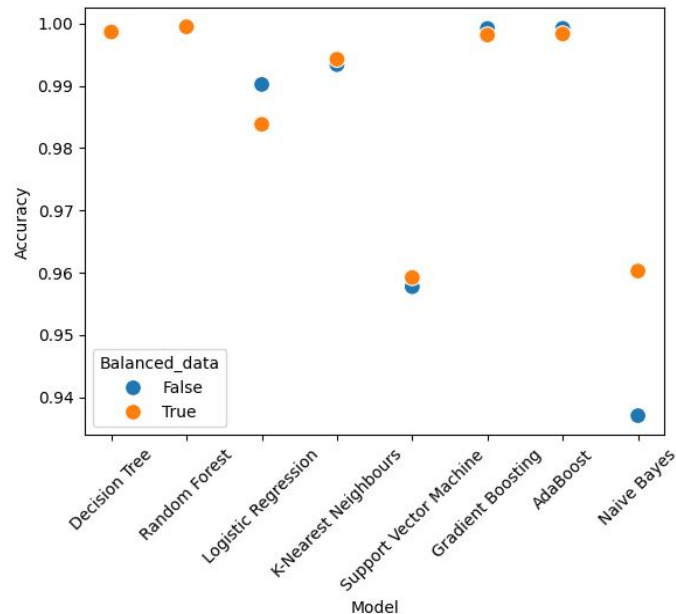
Métricas

- Accuracy
 - Precision
 - Recall
 - F1 score
 - Training time
- 

Resultados

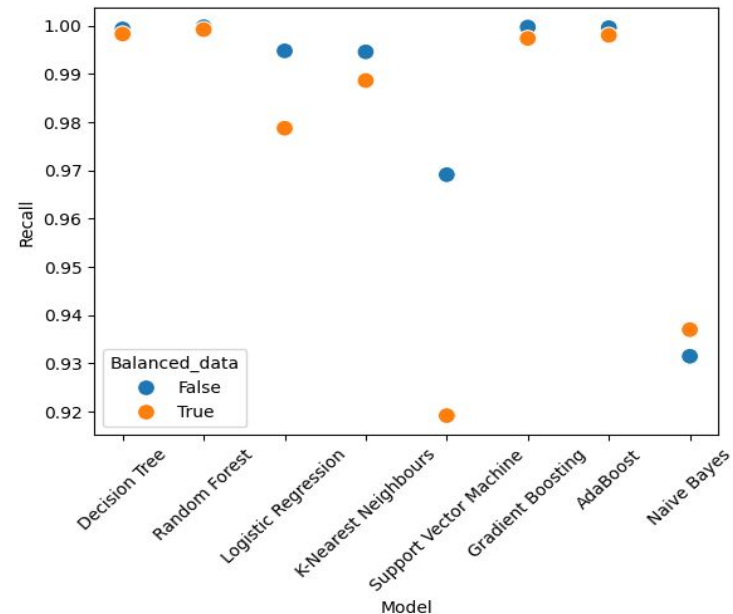
	Model	TN	FP	FN	TP	Accuracy	Precision	Recall	F1	Time	Balanced_data
0	Decision Tree	11988	85	76	112431	0.998708	0.999245	0.999324	0.999285	1.455493	False
1	Random Forest	12031	42	24	112483	0.999470	0.999627	0.999787	0.999707	17.052069	False
2	Logistic Regression	11440	633	587	111920	0.990207	0.994376	0.994783	0.994579	1.241654	False
3	K-Nearest Neighbours	11860	213	610	111897	0.993394	0.998100	0.994578	0.996336	56.617533	False
4	Support Vector Machine	10283	1790	3476	109031	0.957730	0.983848	0.969104	0.976420	3584.387166	False
5	Decision Tree	112173	116	194	112460	0.998622	0.998970	0.998278	0.998624	2.826006	True
6	Random Forest	112258	31	91	112563	0.999458	0.999725	0.999192	0.999458	39.647590	True
7	Logistic Regression	111039	1250	2395	110259	0.983796	0.988790	0.978740	0.983740	3.070830	True
8	K-Nearest Neighbours	112277	12	1281	111373	0.994252	0.999892	0.988629	0.994229	179.578789	True
9	Support Vector Machine	112222	67	9110	103544	0.959203	0.999353	0.919133	0.957566	11664.144424	True
10	Gradient Boosting	12004	69	31	112476	0.999197	0.999387	0.999724	0.999556	54.950413	False
11	AdaBoost	12021	52	46	112461	0.999213	0.999538	0.999591	0.999564	12.933841	False
12	Naive Bayes	11940	133	7712	104795	0.937028	0.998732	0.931453	0.963920	0.116272	False
13	Gradient Boosting	112167	122	297	112357	0.998137	0.998915	0.997364	0.998139	97.402189	True
14	AdaBoost	112134	155	225	112429	0.998311	0.998623	0.998003	0.998313	23.290460	True
15	Naive Bayes	110448	1841	7101	105553	0.960248	0.982858	0.936966	0.959363	0.251786	True

Accuracy



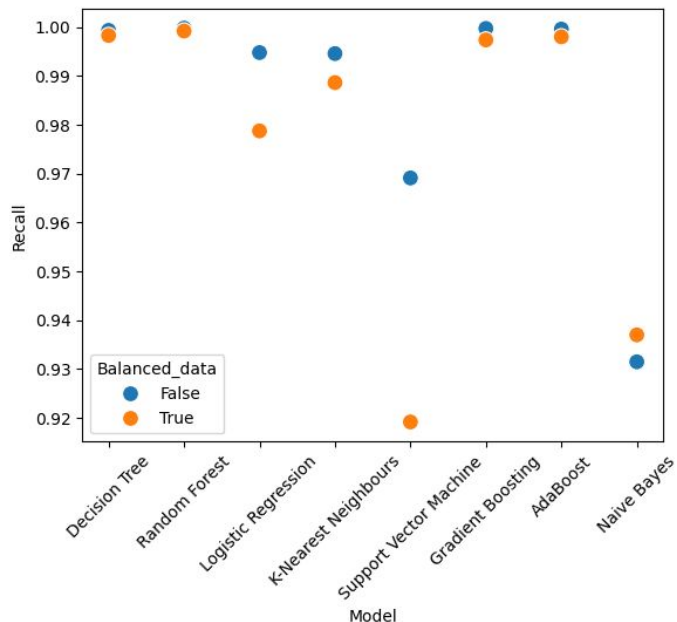
Quanto à *accuracy*, todos os modelos tiveram um desempenho bom, com destaque para o **Random Forest** e **Decision Trees**. Por outro lado o **SVM** e **Naive Bayes** registarem os piores valores nesta métrica.

Precision



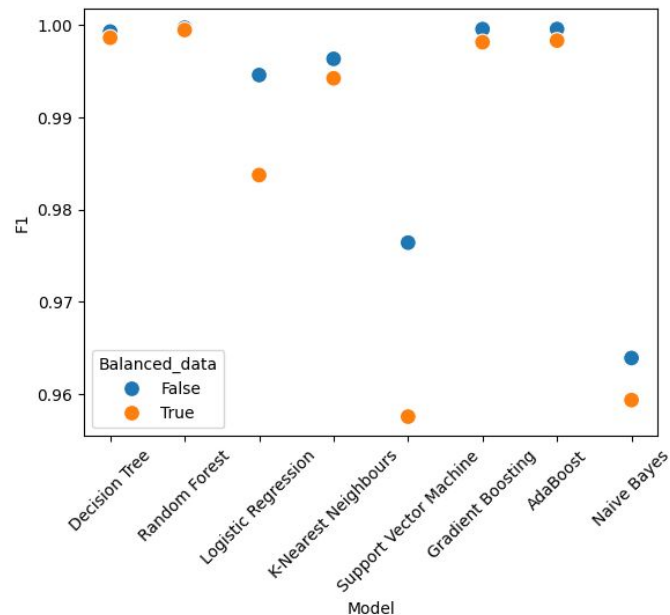
Quanto à precisão, obtivemos também bons resultados transversalmente. Registrou-se valores mais baixos em **Naive-Bayes**, tendo o balanceamento dos dados mostrado influência expressiva apenas no **SVM**.

Recall



Quanto ao *recall*, o **Random Forest** é o algoritmo que apresenta maiores valores, assim como o **Gradient Boosting**. O **SVM** tem o pior desempenho para dados balanceados, e o **Naive Bayes** para dados desbalanceados.

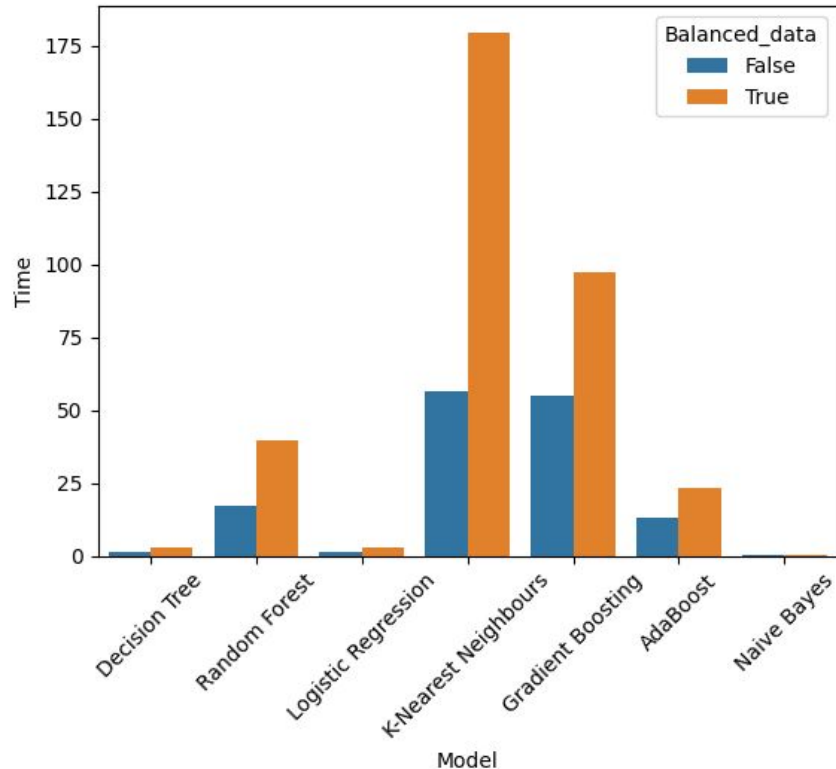
F1 score



O **Random Forest** apresenta-se uma vez mais com os melhores resultados, seguido das **Decision Trees**. **Naive Bayes** revelou os piores resultados.

Tempo de treino

Valores em **segundos**.



O gráfico apresenta os tempos de execução dos diferentes modelos, excluindo o **SVM**. Isto porque o tempo registado pelo mesmo foi excessivamente alto (1h - 3h), tornando-o irrelevante como termo de comparação. Desta forma, **Naive Bayes** foi o algoritmo com menor tempo de treino, quase instantâneo, seguido da **Logistic Regression** e da **Decision Tree**, com tempos também relativamente rápidos de treino.

Conclusões

- Em suma, é possível concluir que o algoritmo **Random Forest** é o **mais consistente** ao longo de todas as métricas, tanto para dados balanceados como não balanceados. As **Decision Trees** e o **Gradient Boosting** mostraram-se também escolhas sólidas.
- Quanto ao **Naive Bayes**, apesar do seu treinamento evidentemente rápido, mostrou menos accuracy e F1 scores do que os restantes modelos.
- **Support Vector Machine** mostrou longos tempos de treino, elevada precisão mas pouco recall. Foi assim o modelo **menos adequado**.
- A escolha última do modelo a aplicar será dependente da prioridade dada a uma determinada métrica em detrimento de outras.