# Proposta Experimental
# Projeto Prático 2 - "There and back again"

**Adlla Katarine Aragão Cruz Passos**[1], **Daniel Alves Costa**[1], **Ramon de Cerqueira Silva**[1]

[1]

## 1. Metodologia

Primeiramente, para o desenvolvimento do código, foi escolhida a linguagem de programação *Python*, a mais popular no meio da Ciência de Dados, e a *IDE Spyder*, sendo conhecida por todos integrantes da equipe.

No pré-processamento, será feita a correção de heróis repetidos através da exclusão e/ou mudança de nome; tratamento de *NaN* por classe mais frequente de cada categoria; preenchimento dos valores ausentes dos atributos por negação do atributo e/ou por valor mais frequente de cada categoria; e transformação de valores nominais para numéricos, realizando uma agregação; união das duas bases de dados, classificando-os pelo seus nomes e excluindo heróis que não estão presentes em ambas as bases. O atributo extra escolhido pela equipe foi *Durability* pois contém uma quantidade equilibrada entre valores *True* e *False*.

Em relação aos algoritmos de predição, foram escolhidos três, *Árvore de Decisão*, utilizado no primeiro problema, *Random Forest*, onde combina-se um conjunto de diversas árvores de decisão com diferentes atributos e escolhido para evitar ou anular *overfitting* dos dados, e *Aprendizagem Bayesiana*, relacionando uma ideia de aprendizagem probabilística.

Para se ter uma melhor eficácia, conseguindo assim uma menor taxa de erro e menos *overfitting*, será necessário a combinação de classificadores, ou seja, utilizar métodos de *ensemble*, treinando cada classificador separadamente e combinando-os logo após. Dois métodos de ensemble será utilizado, o *Bootstrap Aggregating(Bagging)* e o *Boosting*. O *Bagging* consistindo em criar n conjuntos de treinamento idênticos e replicar esses dados de treinamento de forma aleatória(*Bootstrap*) e um conjunto para testes, o treinamento é realizado utilizando os dados agrupados pelo *bootstrap*, após isso deve-se utilizar o conjunto de testes, a classificação sendo realizada pela classe que foi mais "votada" pelos classificadores, todos possuindo o mesmo peso. Já o *Boosting* consiste em realizar o *bootstrap*, assim como no *bagging*, porém na sua classificação a importância do voto de cada classificador é dada com base no desempenho de cada modelo. [1]

Para a avaliação dos classificadores e predição, foram escolhidos alguns métodos. Será usada a acurácia para medir a proporção de predições corretas; a matriz de confusão, que é uma tabela que mostra as frequências de classificação para cada classe do modelo classificando-os como verdadeiro positivo, verdadeiro negativo, falso positivo e falso negativo; a validação cruzada, que particiona os conjuntos de dados em subconjuntos para o treinamento, repetindo o processo várias vezes para avaliar a capacidade de generalização de um modelo; e a curva *ROC* que demonstra o desempenho de um classificador binário e possui dois parâmetros: verdadeiro positivo e falso positivo. A curva *AUC* também será usada, sendo ela derivada da curva *ROC*, é a área abaixo da curva.

Caso haja disponibilidade do grupo para uma melhor análise experimental, será utilizado o *Orange* como um kit de visualização de dados para mineração de dados, e que também pode ser usado como uma biblioteca. Assim como mais um ou dois algoritmos de aprendizagem de máquina tais como *KNN* e *Neural Network*, escolhidos para obter uma predição com uma diversidade maior de algoritmos. Além de utilizar outras abordagens de ensemble, como o *Adaptive Boosting(AdaBoost)* e/ou o *Stacking*.

## Referências

[1] Bruno Butilhão C.. Estudo do algoritmo Adaboost de aprendizagem de máquina aplicado a sensores e sistemas embarcados. Disponível em: http://www.teses.usp.br/teses/disponiveis/3/3152/tde-12062012-163740/pt-br.php. Acesso em Agosto de 2019