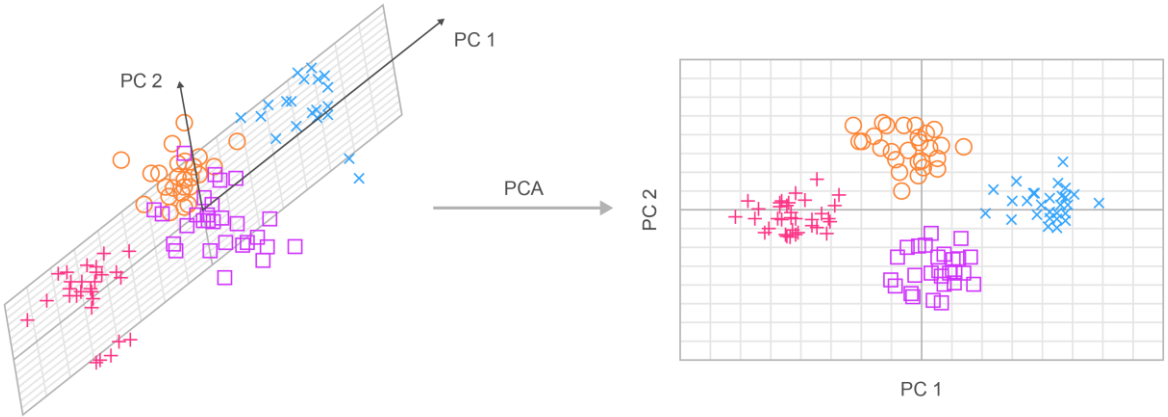

FIAP

O PROBLEMA DA ALTA **DIMENSIONALIDADE...**

ANÁLISE **PCA**

O objetivo da análise é encontrar um meio de condensar a informação contida em várias variáveis originais em um conjunto menor de variáveis estatísticas (componentes) com uma perda mínima de informação (Adaptado Operdata - 2019)



https://medium.com/analytics-vidhya/understanding-principle-component-analysis-pca-step-by-step-e7a4bb4031d9

FEATURE **SELECTION**

Análise de Componentes Principais

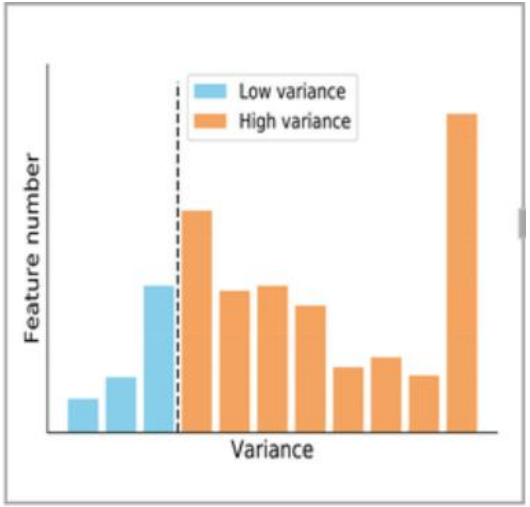
- Mas e se temos características inúteis/ruins em nosso Dataset?

FEATURE **SELECTION**

- Métodos de Filtro (*filter methods*)
- Métodos Embutidos (*embedded methods*)
- Métodos Invólucros (*wrapper methods*)

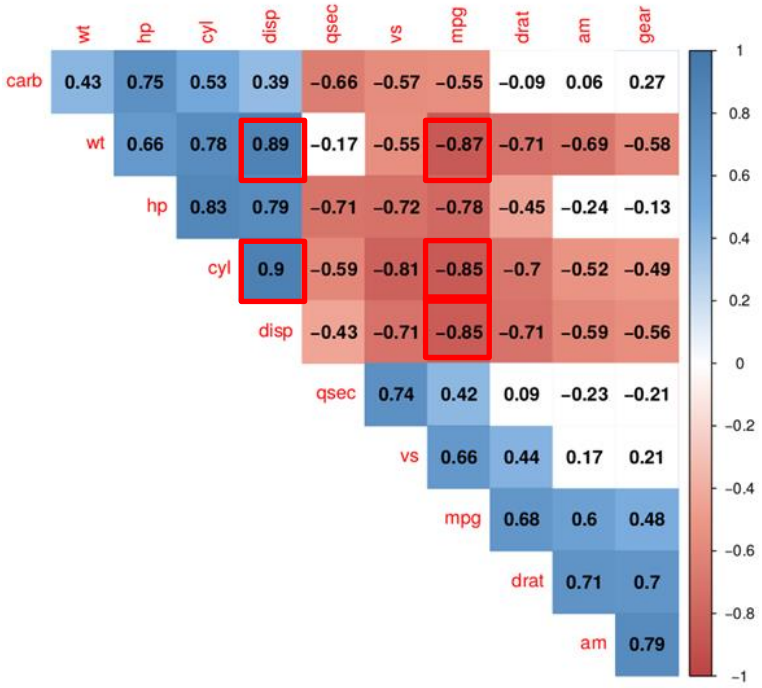
MÉTODOS DE **FILTRO**

Baixa Variâncica

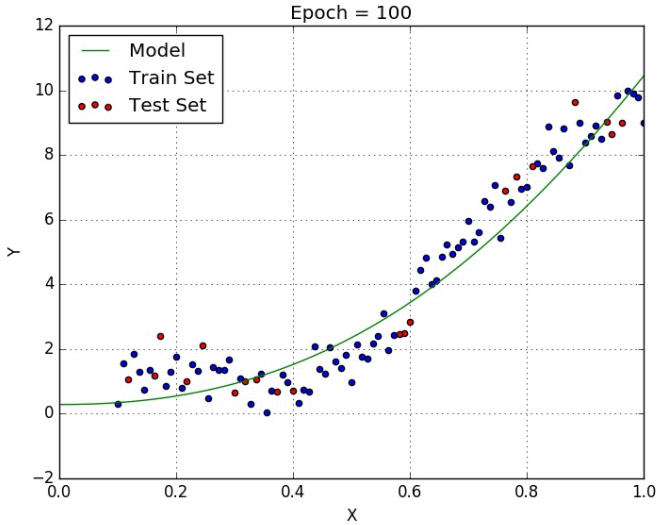


MÉTODOS DE **FILTRO**

Correlação de Pearson



MÉTODOS **EMBUTIDOS**



LASSO, Elastic Net, Ridge Regression

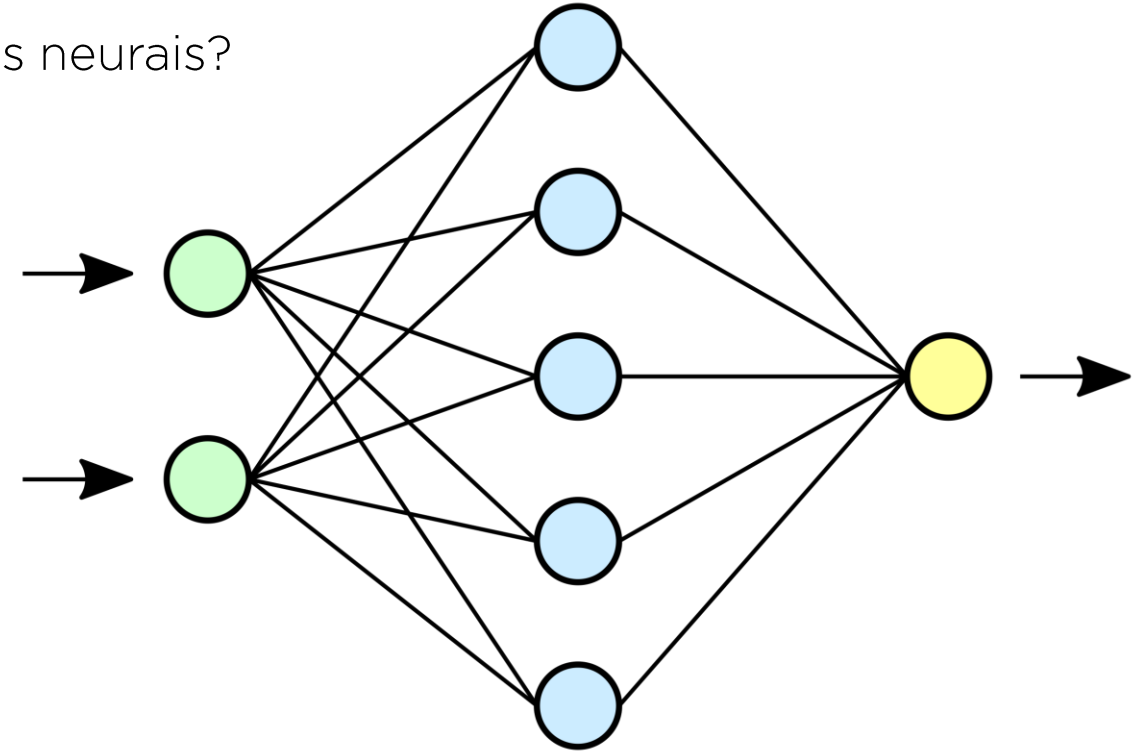
RANDOM FOREST

LightGBM

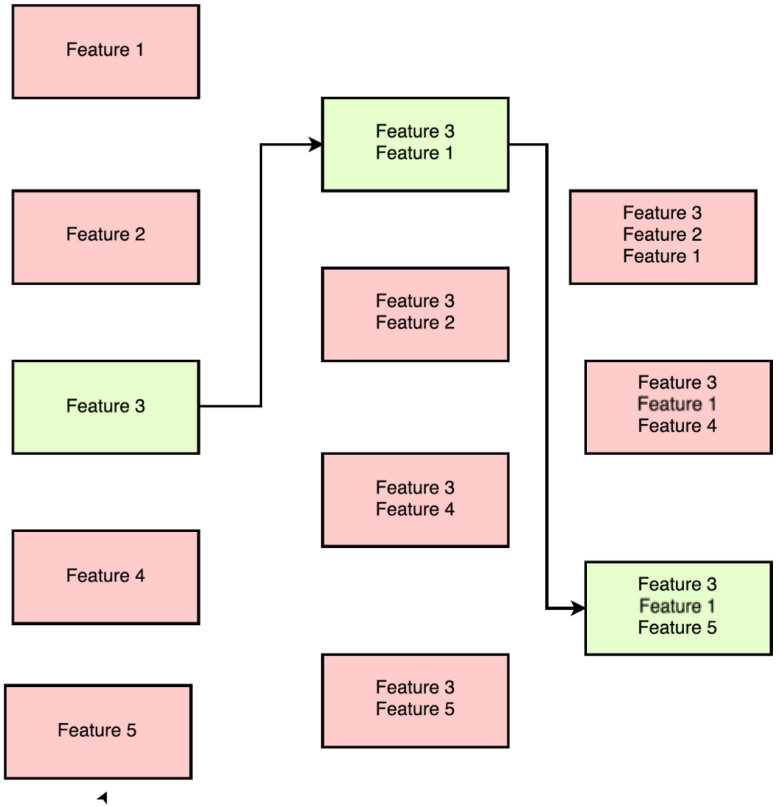
XGBoost

MÉTODOS **EMBUTIDOS**

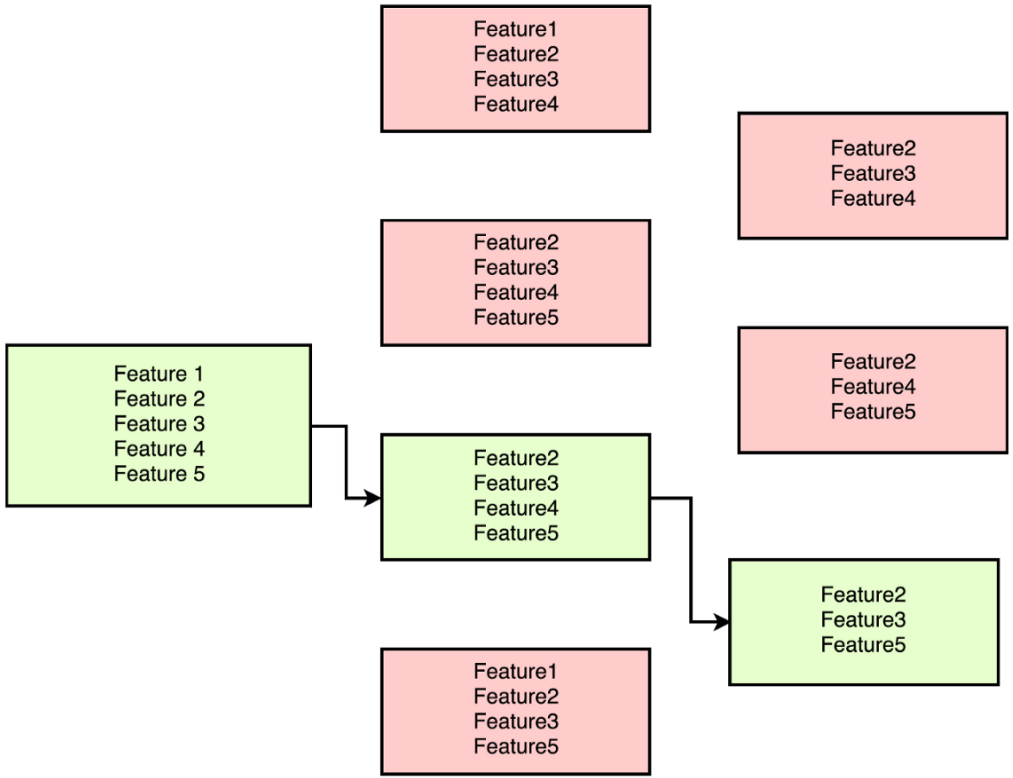
E as redes neurais?



MÉTODOS **WRAPPER** - Forward Selection



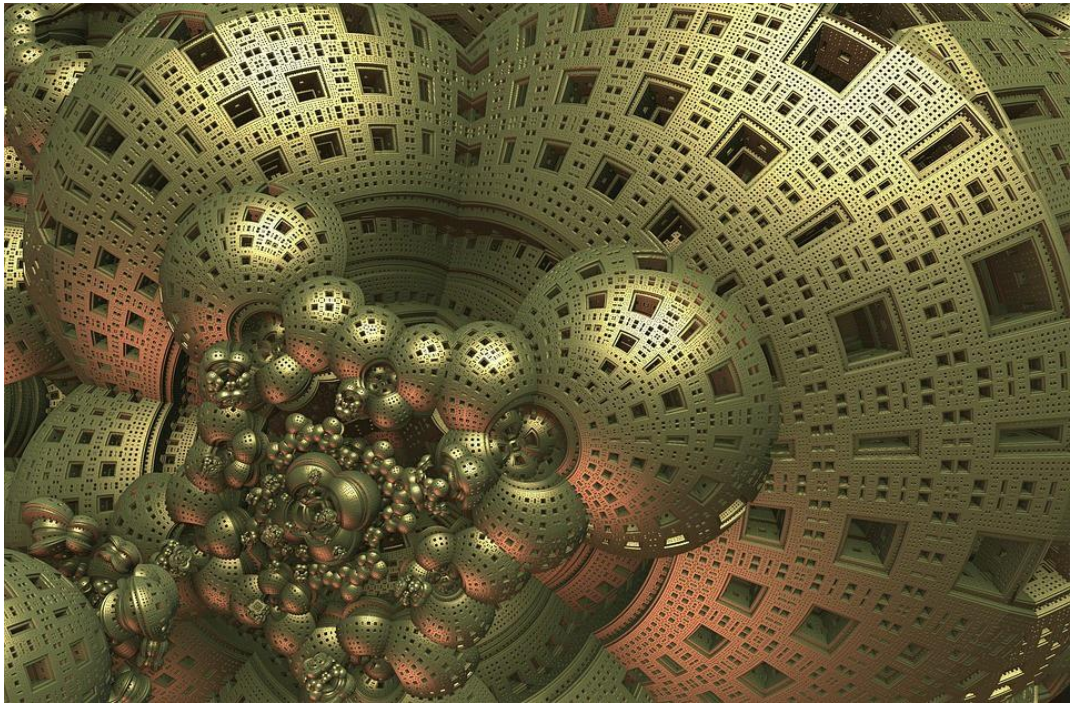
MÉTODOS **WRAPPER** – Backward Elimination



MÉTODOS **WRAPPER – BUSCAS EXAUSTIVAS**

Recursive Feature Elimination e Exhaustive Feature Selection

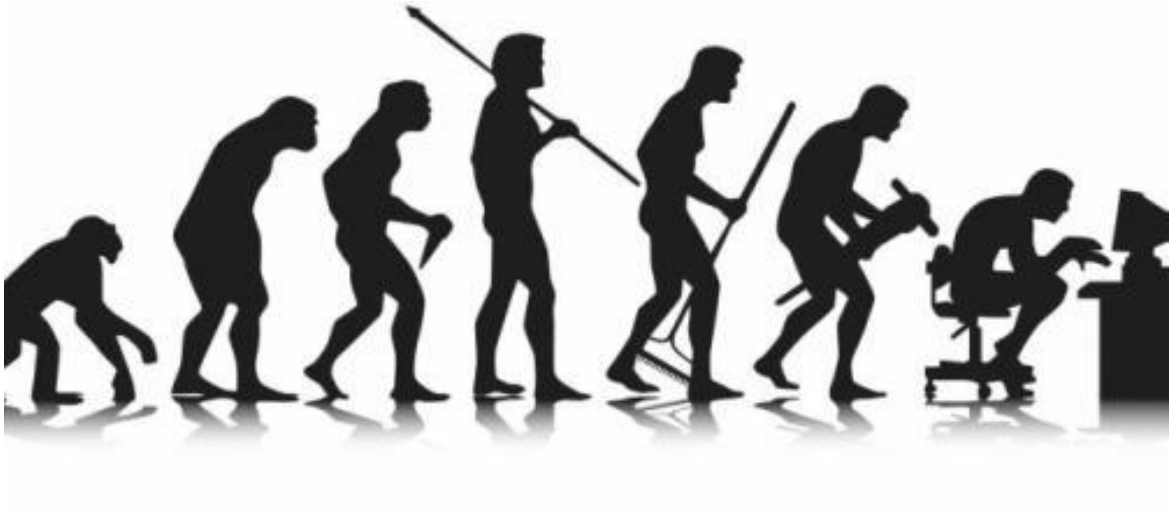
- Para *n* features temos 2^n combinações!



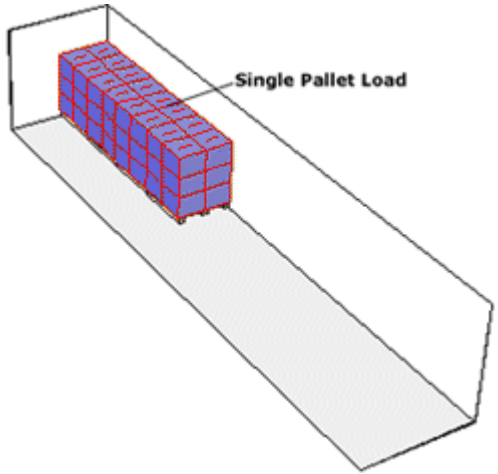
E agora?

Heurísticas!

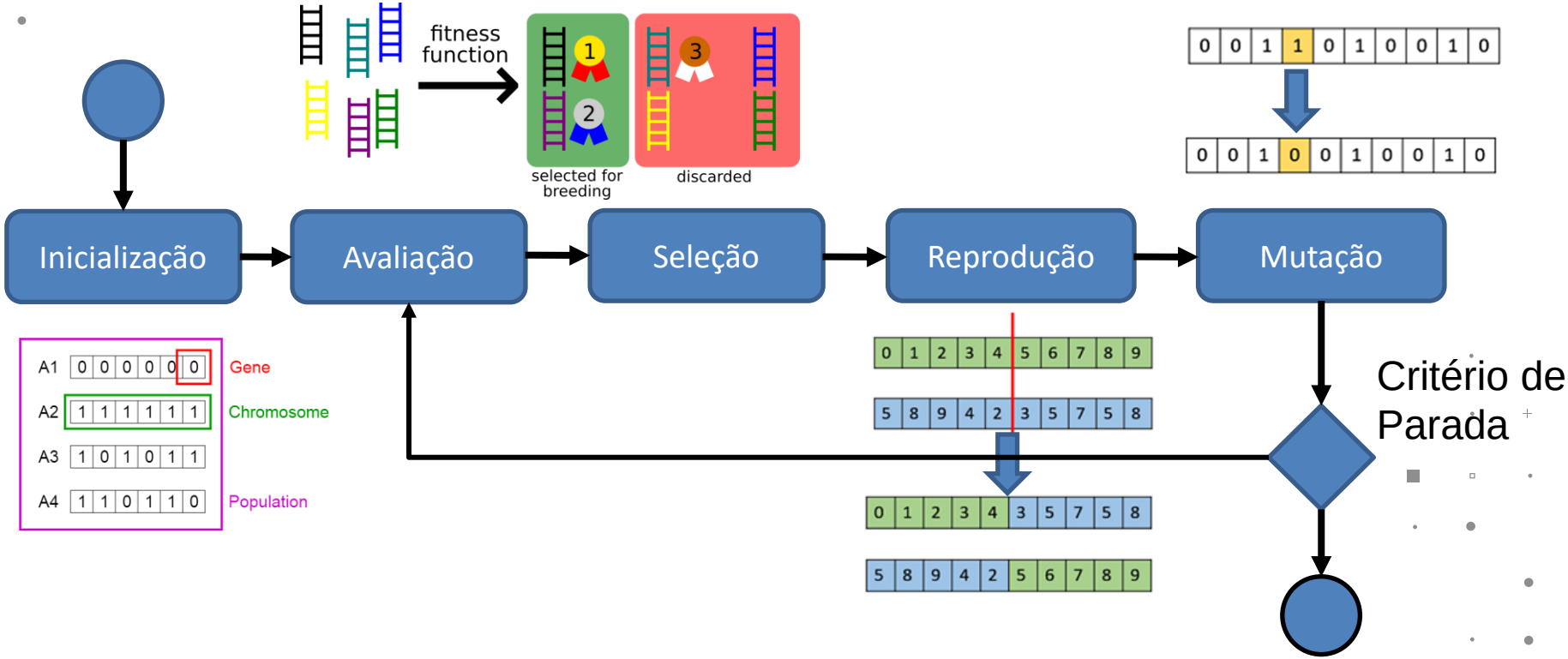
ALGORITMOS **GENÉTICOS**



ALGORITMOS **GENÉTICOS**

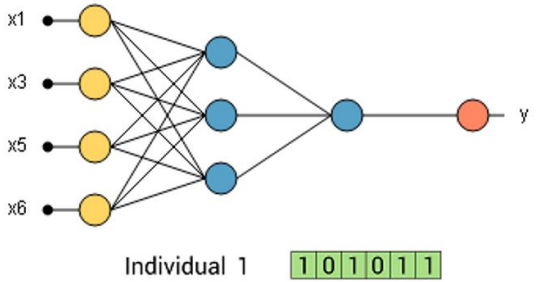


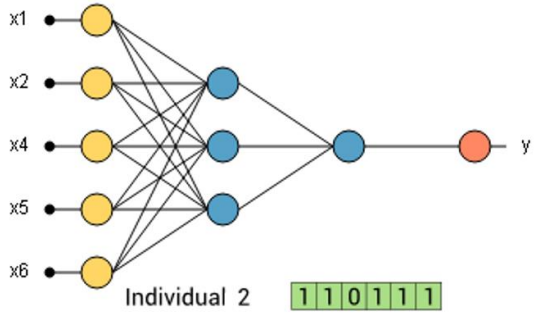
ALGORITMOS **GENÉTICOS**

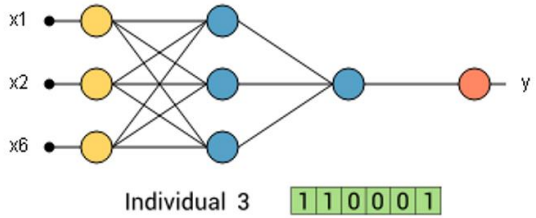


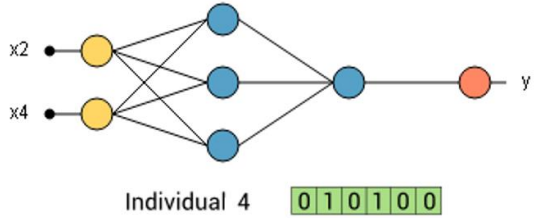
ALGORITMOS **GENÉTICOS**

- Codificação da Solução









Fonte: https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection

ALGORITMOS **GENÉTICOS**

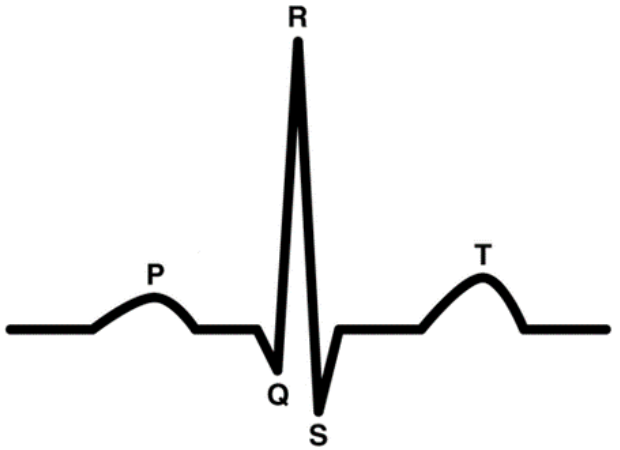
- Função de Avaliação:
  - Seleciona as características do cromossomo e faz validação cruzada (*cross validation*) e obtém a acurácia, usada como critério de avaliação do GA.

```
def calculate_fitness(individual):
        np_ind = np.asarray(individual)
        if np.sum(np_ind) == 0:
            return  (0.0,)
        else:
            feature_idx = np.where(np_ind==1)[0]
            x_temp = X[:,feature_idx]
        cv_set = np.repeat(-1.,x_temp.shape[0])
        skf = StratifiedKFold(n_splits = 5)
        for train_index,test_index in skf.split(x_temp,y):
            X_train,X_test = x_temp[train_index],x_temp[test_index]
            y_train,y_test = y[train_index],y[test_index]
            if X_train.shape[0] != y_train.shape[0]:
                raise Exception()
            classifier.fit(X_train,y_train)
            predicted_y = classifier.predict(X_test)
            cv_set[test_index] = predicted_y
        acc = accuracy_score(y, cv_set)
        return (acc,)
```

ALGORITMOS **GENÉTICOS – CASE I**

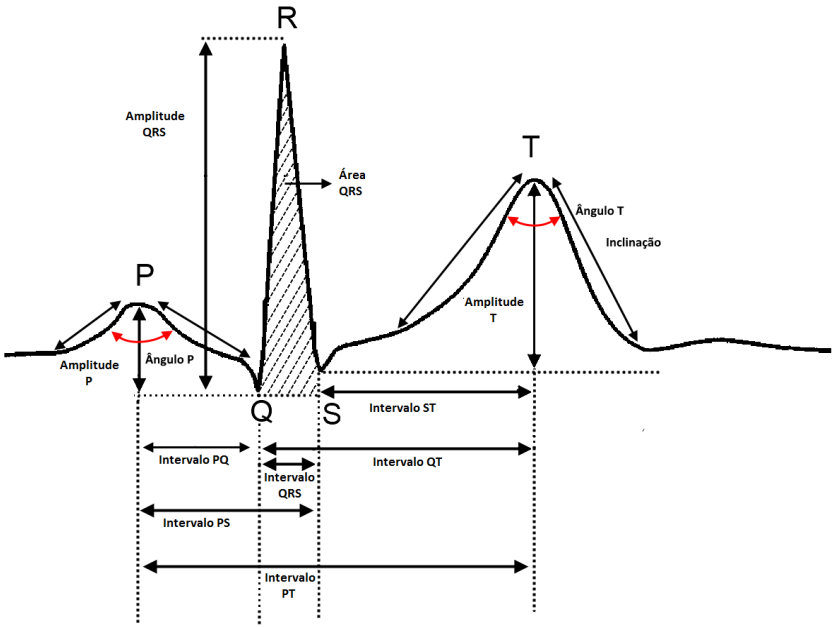
SELEÇÃO DE **FEATURES COM GA** – CASE I

- Problema: Biometria utilizando batimentos cardíacos:



SELEÇÃO DE **FEATURES COM GA** – CASE I

Extração de Características:



- Características do domino do tempo
- Transformada Discreta Cosseno
- Transformada de Fourier
- Função de Autocorrelação
- Modelo Autoregressivo
- Codificação Linear Preditiva
- Transformada Pulso Ativo
- Representação Linear por partes
- Polinômios de Hermite
- Coecientes Mel-cepstrais
- Transformada Wavelet
- Métodos de estimativa da Dimensão Fractal
- Decomposição do Modo Empírico

754 características extraídas

SELEÇÃO DE **FEATURES COM GA** – CASE I

- Conjunto de Dados:
  - PTB ECG Database;
  - 290 pessoas distintas (logo 290 classes);
  - 754 características extraídas;
  - A quantidade de gravações por pessoa varia de 2 a 20, aproximadamente 2000 amostras.

SELEÇÃO DE **FEATURES COM GA** – CASE I

- Resultados:
  - Acurácia de **97,93%** no conjunto de teste, redução de **754** características para **31** após o término da execução;
  - Classificador Optimum-Path Forest utilizado na função fitness;
  - Artigo publicado em: https://ieeexplore.ieee.org/document/7966216

OBRIGADO


FIAP