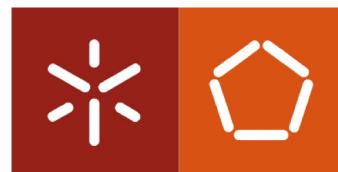


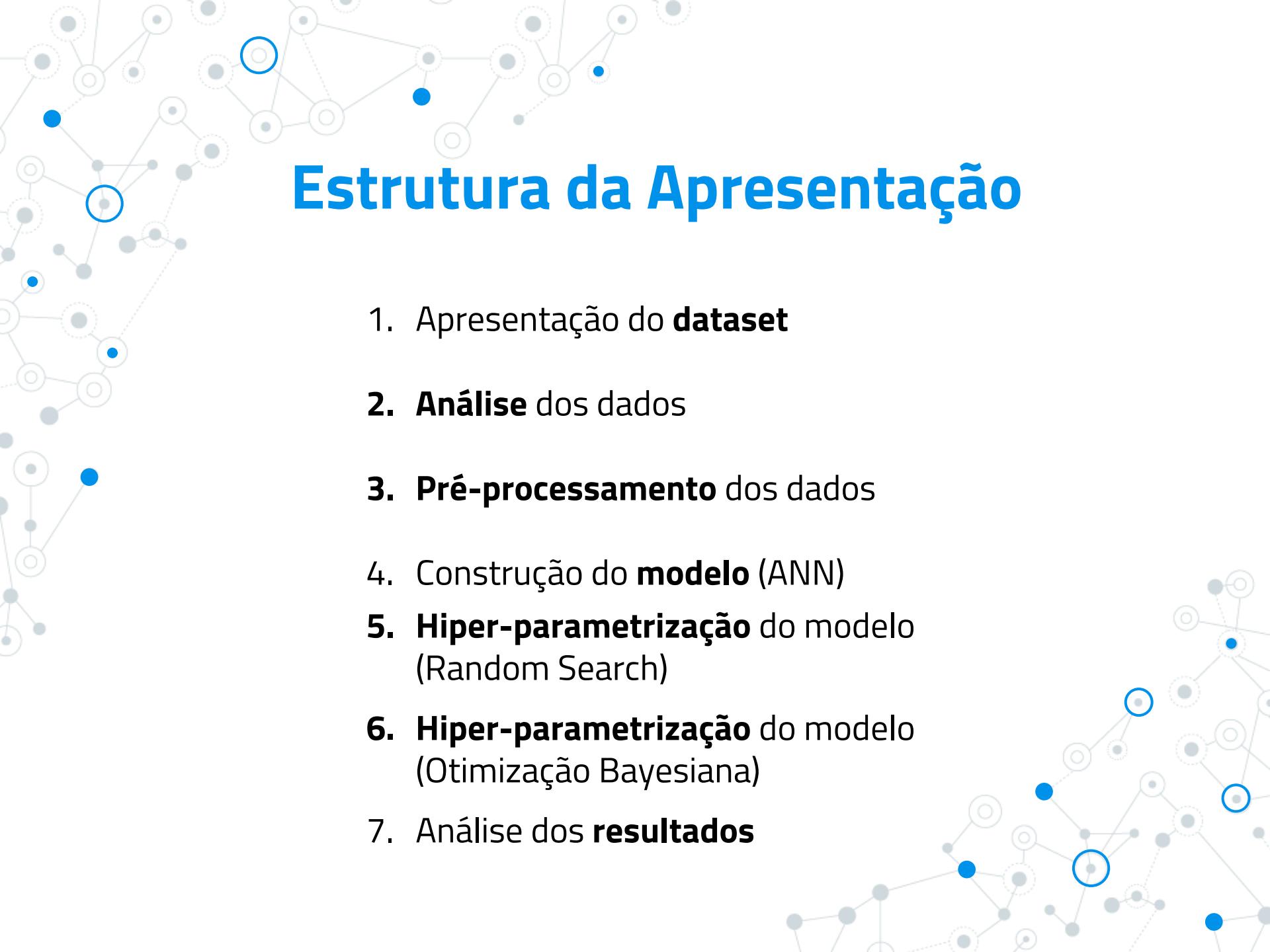
Estudo do dataset

Mammographic Masses



André Pereira, Carlos Lemos, João Barreira e Rafael Costa

Computação Natural (perfil Sistemas Inteligentes)
Mestrado em Engenharia Informática - 2018/19
Universidade do Minho



Estrutura da Apresentação

1. Apresentação do **dataset**
2. **Análise** dos dados
3. **Pré-processamento** dos dados
4. Construção do **modelo** (ANN)
5. **Hiper-parametrização** do modelo
(Random Search)
6. **Hiper-parametrização** do modelo
(Otimização Bayesiana)
7. Análise dos **resultados**

Mammographic Masses Dataset

- Dados clínicos correspondentes a **961 exames de mastografia**
- Métricas:
 - **BI-RADS**: nível da avaliação “Breast Imaging-Reporting and Data System” (incompleto=0, negativo=1, achado benigno=2, provavelmente benigno=3, suspeita de anormalidade=4, altamente sugestivo de malignidade=5, malignidade comprovada através de biópsia=6)
 - **Age**: idade do paciente (em anos)
 - **Shape**: forma da massa (redonda=1, oval=2, lobular=3, irregular=4)
 - **Margin**: margem da massa (circunscrito=1, micro-lobulado=2, obscurecido=3, mal-definido=4, espiculado=5)
 - **Density**: densidade da massa (alto=1, médio=2, baixo=3, contém gordura=4)
 - **Severity**: severidade da massa (benigno=0, maligno=1)
- Objetivo da análise: **deteção de massas malignas** no tecido mamário

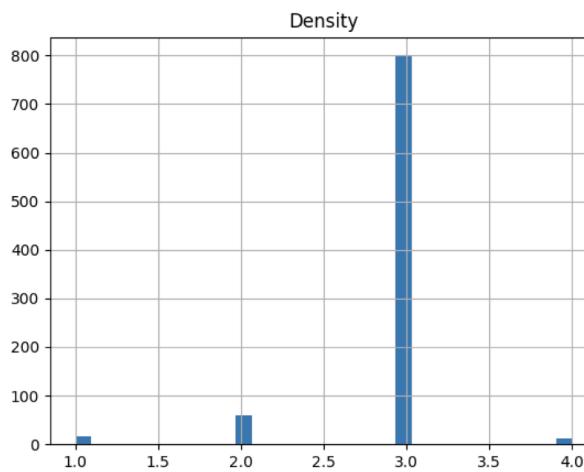
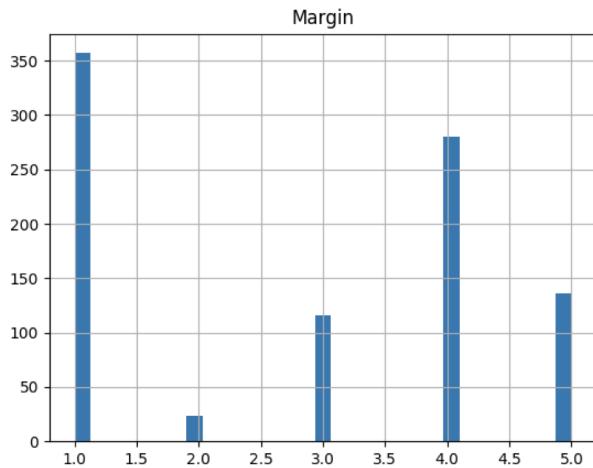
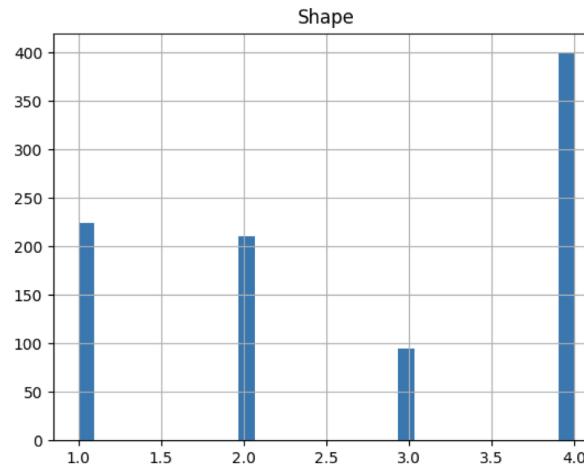
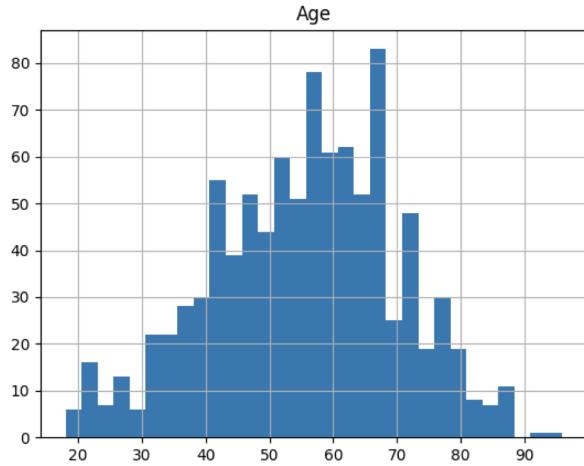
Análise dos Dados

Métrica	Média (σ)	Mediana	Min	Max	Percentil 75	#NaN
Age	55.49 (14.48)	57	18	96	66	5
Shape	2.72 (1.24)	3	1	4	4	31
Margin	2.80 (1.57)	3	1	5	4	48
Density	2.91 (0.38)	3	1	4	3	76
Severity	0.46 (0.50)	0	0	1	1	0

- **Distribuição por classes (severity):** 516 benignos (0), 445 malignos (1)

Análise dos Dados

Gráficos de Distribuição



Pré-processamento dos Dados

- Remoção dos registos com valores em falta (**Missing Data Filtering**)
- **Standardização** dos dados das *features*

```
# Discard lines with NaN values  
df = df.dropna()
```

```
# Standardize feature data  
scaler = StandardScaler()  
features[features.columns] = scaler.fit_transform(features[features.columns])
```



Construção do Modelo

- Criação do modelo através da biblioteca **Keras**
- Utilização de um *wrapper* (**KerasClassifier**) que possibilita a utilização de funcionalidades da biblioteca **scikit-learn**
- Facilita os processos de **Cross-Validation** e **Random Search**

```
# Create Keras classifier model

classifier = KerasClassifier(build_fn=create_model, verbose=0)

def create_model(hidden_layers=2, nodes_per_layer=3, activation_fn='relu', learning_rate=1e-2):
    model = Sequential()
    model.add(Dense(4, activation=activation_fn, input_shape=(4,))) # input layer

    for i in range(hidden_layers):
        model.add(Dense(nodes_per_layer, activation=activation_fn))

    model.add(Dense(1, activation=activation_fn)) # output layer

    adam = Adam(lr=learning_rate)
    model.compile(loss='mean_squared_error', optimizer=adam, metrics=['accuracy'])

    return model
```



Hiper-parametrização do Modelo

(Random Search)

- Definição da **distribuição** dos hiper-parâmetros:
 - **hidden_layers**: [2, 4, 8, 16, 32]
 - **nodes_per_layer**: [0, 1, ..., 19, 20]
 - **activation_fn**: ReLU ou Sigmoid
 - **learning_rate**: [1e-8, ..., 1e-2]
- Utilização das funcionalidades de **Random Search** do scikit-learn
- **Cross-validation** com 10 folds ($k=10$)
- Apresentação dos resultados para os **melhores modelos** encontrados

```
random_search = RandomizedSearchCV(classifier, param_distributions=hp_dist, n_iter=20, cv=10)
random_search.fit(features, target)
print_top_results(random_search.cv_results_)
```

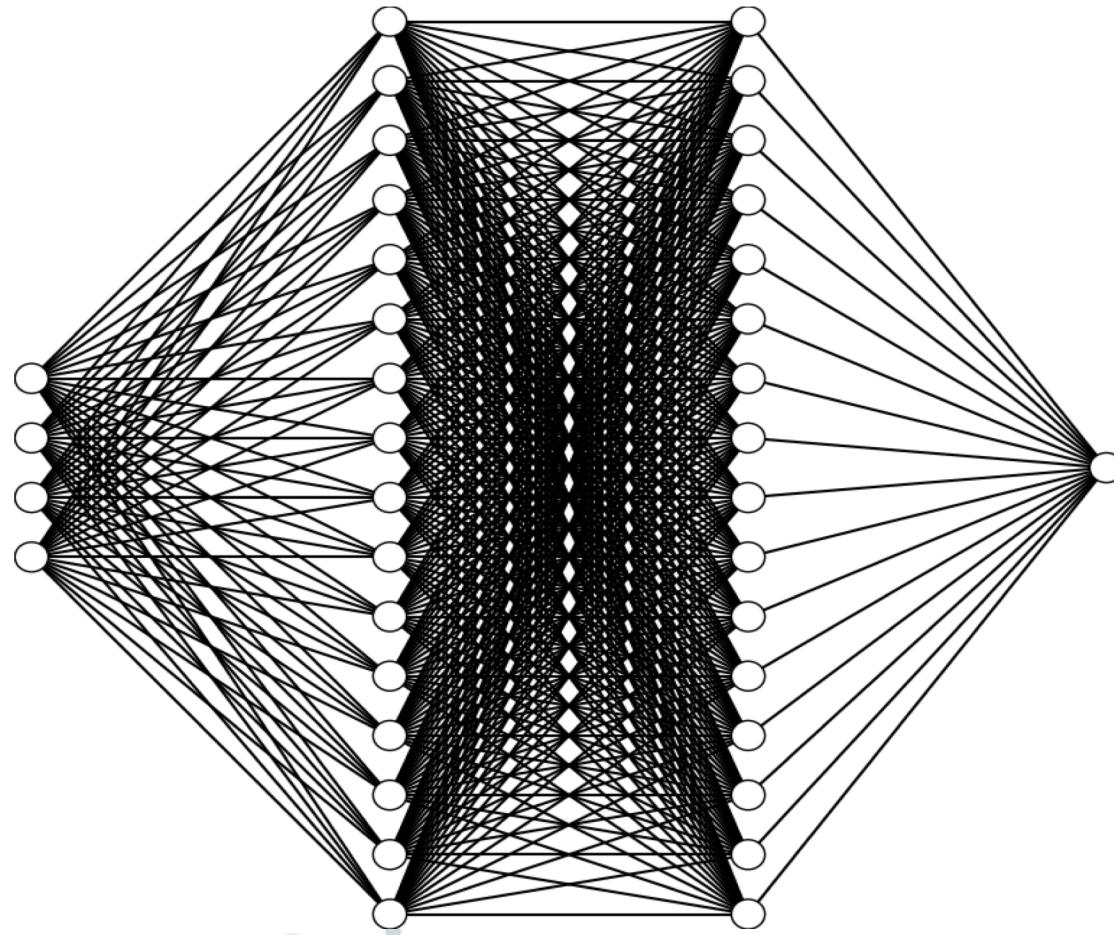


Model with rank: 1

Mean validation score: 0.733 (+/- 0.099)

Parameters: {'nodes_per_layer': 16, 'learning_rate': 0.01, 'hidden_layers': 2, 'activation_fn': 'relu'}

Model with rank: 2



Hiper-parametrização do Modelo

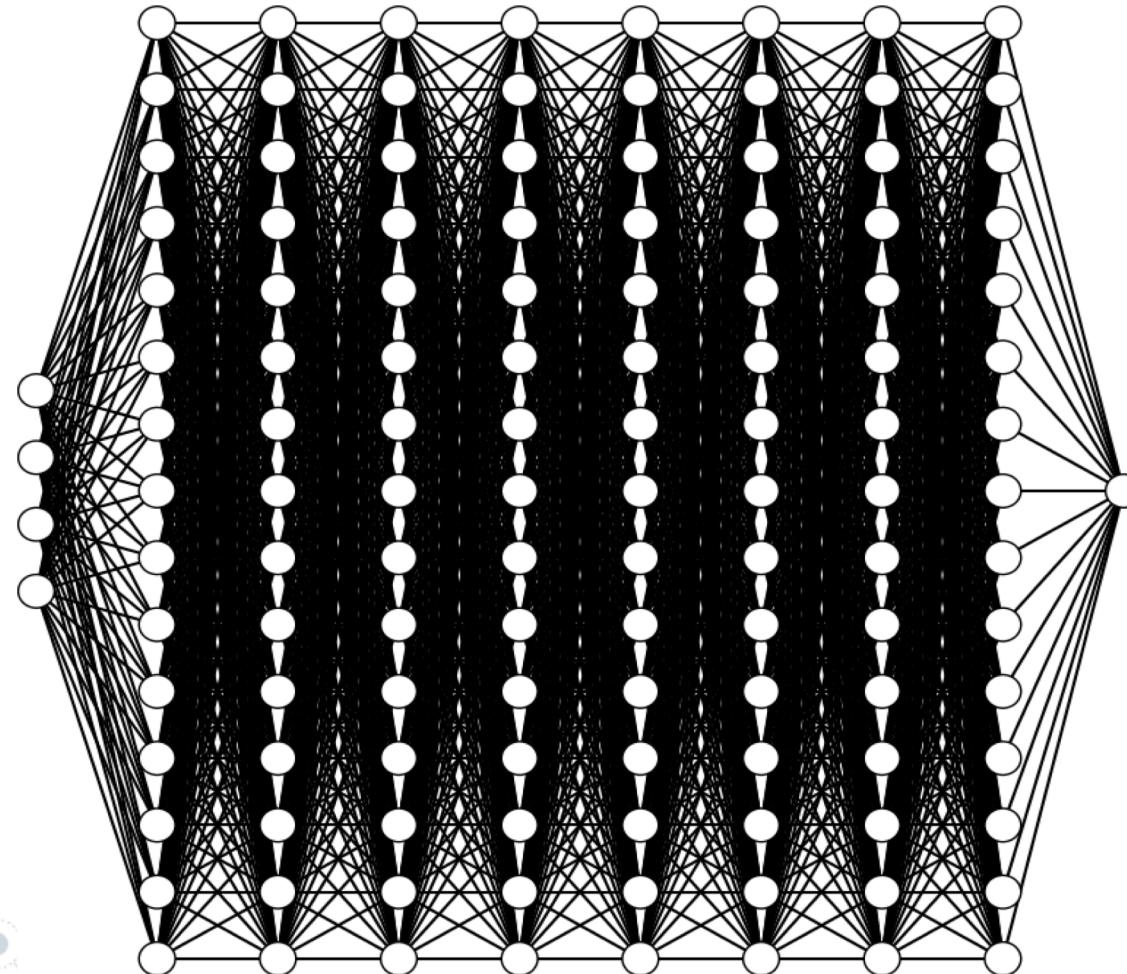
(Otimização Bayesiana)

- Utilização das bibliotecas **GPy** e **GPyOpt**
- Definição de uma **função de avaliação** do modelo para um subconjunto de hiper-parâmetros
- Passagem dessa função ao **método de Otimização Bayesiana**, juntamente com a **distribuição** dos valores dos hiper-parâmetros
- Execução da otimização (100 iterações) e **apresentação do melhor resultado** (i.e. valores dos hiper-parâmetros que resultaram num menor **loss** ou **accuracy**)



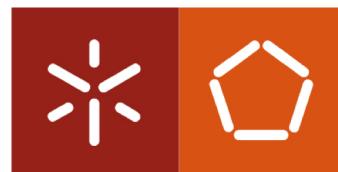
```
Optimized Parameters:  
hidden_layers: 8.0  
nodes_per_layer: 15.0  
activation_fn: 0.0  
learning_rate: 0.01
```

```
optimized accuracy: 0.7745454554124312
```



Estudo do dataset

Mammographic Masses



André Pereira, Carlos Lemos, João Barreira e Rafael Costa

Computação Natural (perfil Sistemas Inteligentes)
Mestrado em Engenharia Informática - 2018/19
Universidade do Minho