

## Questionário preparatório para avaliação – IA e Machine Learning

Respostas basear-se nos slides e nos programas desenvolvidos.

1. O que é aprendizado em máquinas?
2. Defina:
  - a) Classificação
  - b) Regressão
  - c) Acurácia
  - d) Erro ou perda
  - e) Previsão (“predição”)
  - f) Modelo
  - g) Características, ou propriedades, ou atributos (“features”)
3. O que representa a área de aprendizado em máquinas em relação a todas as áreas da I.A.?
4. Em aprendizado em máquinas, um modelo é o conjunto de regras que explica os dados para qual foi treinado e que serve para realizar previsões. Comparando o universo dos dados (conceito que define os dados reais) e o modelo, quais os casos que podem ocorrer comparando modelo e conceito?
5. Qual a utilidade dos contraexemplos?
6. Defina aprendizado supervisionado.
7. Defina aprendizado não supervisionado.
8. Nos modelos baseados em Python e suas bibliotecas, o que executam os métodos “*fit*” e “*predict*”?
9. O que são dados numéricos (quantitativos) e categóricos (qualitativos)? Como tratar cada um destes tipos ao utilizar um modelo classificador?
10. O que são e para que serve subdividir os dados em conjunto de teste e de treinamento?
11. Como é estruturada uma Redes multicamadas (“*multilayer perceptron*” – MLP) ou densa?
12. O que é uma função neural ou de ativação e quais as mais utilizadas?
13. Como um neurônio artificial processa os dados (quais suas entradas, qual o processamento interno dele e qual a saída)?
14. A tabela abaixo contém 10 informações sobre clientes de uma empresa de telecomunicações e se eles cancelaram ou não seus serviços.

Cliente	Ocupação	Gênero	Plano	Uso de Dados	Rotatividade
1	junior	Masculino	Pré-pago	Baixo	Não
2	pleno	Feminino	Pós-pago	Médio	Sim
3	pleno	Masculino	Pré-pago	Alto	Não
4	junior	Feminino	Pós-pago	Baixo	Não
5	pleno	Masculino	Pré-pago	Médio	Sim
6	pleno	Feminino	Pós-pago	Alto	Não
7	senior	Masculino	Pré-pago	Baixo	Sim
8	senior	Feminino	Pós-pago	Médio	Não

Cliente	Ocupação	Gênero	Plano	Uso de Dados	Rotatividade
9	senior	Masculino	Pré-pago	Alto	Sim
10	senior	Feminino	Pós-pago	Baixo	Não

Qual o resultado numérico do OneHot Encoder para o registro [pleno, feminino, pós-pago, alto]?

15. Se da biblioteca *sklearn.preprocessing* o OneHot Encoder cria uma quantidade de *bits* igual ao número de valores para cada atributo (variável), o que o LabelEncoder gera? Teste ambos na tabela da questão acima.

16. Defina matriz de confusão e dê um exemplo para três classes ou categorias.

17. Em que condições uma rede neural é considerada “treinada”?

18. Qual a importância da etapa “*backpropagation*” no treinamento de uma rede neural?

19. Quais tipos de redes neurais utilizam o “*backpropagation*” para reajuste dos pesos das conexões?

20. Defina a) “*overfitting*” (superajuste), b) “*underfitting*” (ajuste fraco) para os dados e c) generalização de um modelo.

21. Quando a acurácia para o conjunto de treinamento continua aumentando, mas passa a diminuir para o conjunto de testes, este problema para a capacidade de generalização do modelo é denominado \_\_\_\_\_.

22. Descreva época (ciclos ou “*epochs*”) e lote (“*batch*”).

23. Descreva: a) regressão linear, b) regressão não-linear, c) regressão linear múltipla

24. Qual a utilidade do coeficiente de correlação e quais casos podem ocorrer em relação aos dados?

25. O que executa esta parte do código do programa *play\_tennis*, para gerar o *dataframe* X?

```
X = df_jogar.loc[:, 'Aparencia':'Vento']
```

26. a) O que executa esta parte do código do programa *play\_tennis*?

```
X = encoder.fit_transform( df_jogar.loc[:, 'Aparencia':'Vento'] )
```

27. O que executa esta parte do código do programa *kerasWineQualUCI* ?

```
escala = StandardScaler()
X_train = escala.fit_transform(X_train)
X_test = escala.transform(X_test)
Qual a finalidade de realizar esta transformação?
```

28. No programa *wineQual\_UCI*, usando as funções da biblioteca “scikit-learn” relacionadas a seguir, qual a arquitetura (sequência e tamanho) das camadas da rede neural codificada?

```
print("Atributos de entrada = ", mlp.n_features_in_)
print("N de camadas = ", mlp.n_layers_)
print("Tamanhos das camadas ocultas: ", mlp.hidden_layer_sizes)
print("N de neurons saída = ", mlp.n_outputs_)
```

29. Execute este programa, explique qual o modelo de classificador utilizado e escreva qual o novo dado de teste e sua previsão.

```
### bibliotecas
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
import pandas as pd
import numpy as np

### Exemplo de dados
dados = pd.DataFrame({
    'Cor': ['Vermelho', 'Azul', 'Verde', 'Vermelho', 'Azul', 'Verde'],
    'Forma': ['Círculo', 'Quadrado', 'Triângulo', 'Círculo', 'Triângulo', 'Círculo'],
    'Classe': ['A', 'B', 'B', 'A', 'B', 'A']
})

# One-Hot Encoding
encoder = OneHotEncoder(sparse_output=False)
encoded_data = encoder.fit_transform(dados[['Cor', 'Forma']])

### Preparação dos dados
X = encoded_data
y = dados['Classe']
print(X, y)

# Treinamento do modelo
modelo = RandomForestClassifier(random_state=42)
modelo.fit(X, y)

### Dois novos dados de teste: círculo verde e quadrado verde
novo_dado = pd.DataFrame({
    'Cor': ['Verde', 'Verde'],
    'Forma': ['Círculo', 'Quadrado']
})

# Usando o mesmo encoder para o novo dado de teste
novo_dado_encoded = encoder.transform(novo_dado)

print("Novo dado de teste codificado:")
print(novo_dado_encoded)

### Novo_dado_encoded para fazer previsões com o modelo
previsao = modelo.predict(novo_dado_encoded)
print("Previsão Forma", novo_dado.iloc[0,1],
      "Cor", novo_dado.iloc[0,0], ":", previsao[0])
print("Previsão Forma", novo_dado.iloc[1,1],
      "Cor", novo_dado.iloc[1,0], ":", previsao[1])
```

30. O que executa esta parte do código do programa *WineQualUCI* e qual o dado fornecido, nos casos de: a) classificação e b) regressão?

```
# vinho desconhecido com as medições: 1 2 3 4 5 6 7 8 9 10 11
x_prev = np.array([[ 7.0,  0.5,  0.2,  2.0,  0.07, 10.0, 50.0, 0.991, 3.3, 0.55,
10.0 ]])
y_prev = mlp.predict(x_prev)
print('\nQualidade prevista: ', y_prev )
classe_prev = y[np.argmax(y_prev)]
print('\nClasse prevista: ', classe_prev, ' com probabilidade ',
y_prev[:,classe_prev])
```