FGV EMAp
ESCOLA DE MATEMÁTICA APLICADA

# Técnicas e Algoritmos em Ciência de Dados
# Tarefa avaliada 1

Esta tarefa deverá ser enviada até 16 de abril de 2020, às 8h00.
Envios atrasados NÃO serão aceitos.

## Resultados avaliado

Esta tarefa testará alguns conceitos básicos de programação Python para análise de dados (Pandas, Seaborn, Scikit-learn), incluindo carregamento e limpeza de dados, construção e avaliação de modelos simples de classificação e plotagem básica.

## Instruções

**Identificador**
Escolha um número aleatório de 6 dígitos e escreva-o na primeira célula do notebook. Assegure-se de manter uma cópia desse número, pois ele será usado para fornecer feedback (evite números triviais, como 000000 ou 123456 - obrigado).

**Envio**
Envie seus arquivos através do ECLASS. Os arquivos enviados não podem ser sobregravados por outra pessoa, e não podem ser lidos por nenhum outro aluno. No entanto, você pode sobregravar o envio com a frequência desejada, reenviando, embora somente a última versão enviada seja mantida. O envio após o prazo NÃO será aceito.

*Se tiver problemas, no último minuto, envie um e-mail para sua tarefa como anexo em alberto.paccanaro@fgv.br com o assunto "ENVIO URGENTE - TAREFA 1". No corpo da mensagem, explique o motivo de não enviar por ECLASS.*

**IMPORTANTE**

- Seu envio consistirá em um único notebook Python implementando suas soluções.
- O nome do arquivo será o número aleatório que o identifica (por exemplo, 568423.ipynb)
- Este curso consiste em 3 partes. Certifique-se de que as 3 partes estejam claramente separadas e identificáveis no seu notebook Python.
- NÃO ENVIE NENHUM CONJUNTO DE DADOS, apenas o código.
- Qualquer função de utilitário que você usará deve ser incluída no notebook - não envie scripts de utilitários.

**ACONSELHAMENTO SOBRE EXERCÍCIOS DE BÔNUS**

O valor de cada exercício em porcentagens é fornecido. *Observe, no entanto, que a soma total das notas dos exercícios é 110/100. Isso ocorre porque 10 pontos extras são dados para* **exercícios BÔNUS** – eles são claramente marcados no texto abaixo**.** *Observe que esses exercícios BÔNUS são difíceis e demorados. Aconselhamos que você tenha uma solução de trabalho final para todo o curso antes de tentar responder a esses exercícios opcionais.*

**Todo o trabalho que você enviar deve ser exclusivamente seu próprio trabalho. Os envios de trabalho de curso serão verificados quanto a isso.**

## Critérios de Marcação

Este curso é avaliado e obrigatório e vale 25% do seu total de notas finais para este curso. Para obter todos os pontos de cada pergunta, você deve respondeê-la corretamente e completamente. Pontos serão dados para códigos bem escritos e estruturados.

## PERGUNTAS

**Parte 1 - Carregamento de dados e pré-processamento - valor desta seção: 25 %**

Faça download do arquivo "Part1.tsv" da plataforma ECLASS.
Observe que este arquivo tem um formato customizado: cada ponto tem 5 características (os primeiros 5 valores) e um rótulo de classe associado (o último valor). Nesta parte, você carregará e visualizará o conjunto de dados.

Estas são as etapas que você precisará implementar:

a) Carregar os dados em um DataFrame de pandas
b) Limpar o conjunto de dados removendo os pontos com valores ausentes
c) Para cada característica do conjunto de dados, faça uma figura que contenha 2 sub-gráficos mostrando:
   - O histograma dos valores da característica (todas as classes combinadas).
   - O histograma dos valores da característica separadamente para cada classe. Use uma cor diferente para cada classe.

   Observe que todos os seus histogramas devem ter 50 barras (intervalos) e devem ser normalizados (a área sob o histograma deve totalizar até 1).
   Para cada característica, a figura deve se parecer com a Figura 1.

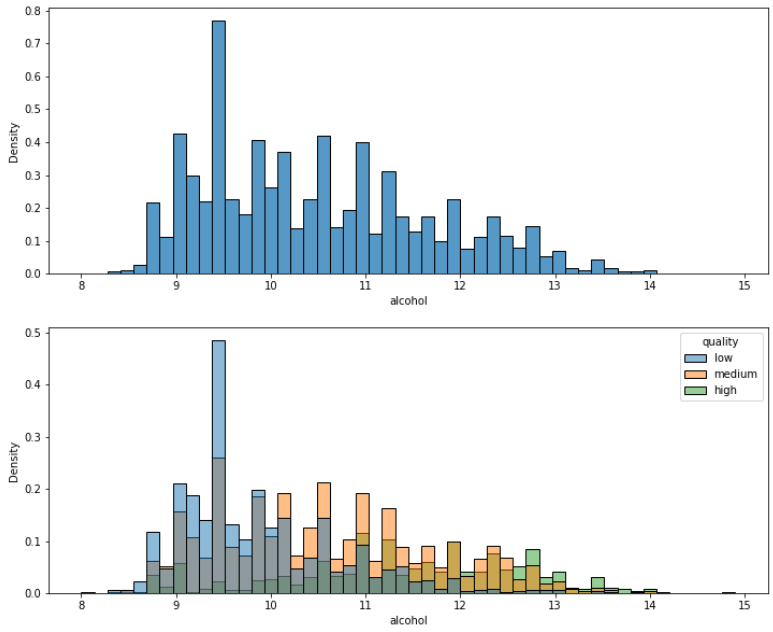


**Figura 1**

d) Faça uma figura contendo uma matriz de sub-gráficos. Esses sub-gráficos apresentarão projeções do conjunto de dados em cada par de características, histograma de suas distribuições e valores dos coeficientes de correlação de Pearson.
   A matriz de sub-gráficos será organizada como a figura 2.
   - A seção triangular superior da matriz conterá na entrada $(i, j)$ a correlação de Pearson entre feature i e feature j.

- A diagonal principal conterá na posição $(i, i)$ os histogramas dos valores da feature $i$. Esses histogramas são os mesmos que você já plotou no ponto c): você deve usar uma cor diferente para cada classe e os histogramas devem ter $50$ barras (intervalos) e devem ser normalizados (a área abaixo do histograma deve totalizar até 1).
- A seção triangular inferior da matriz conterá na entrada $(i, j)$ as projeções dos pontos nas características $i$ e *j*. Use uma cor diferente para cada classe e as cores devem corresponder às usadas nos histogramas.

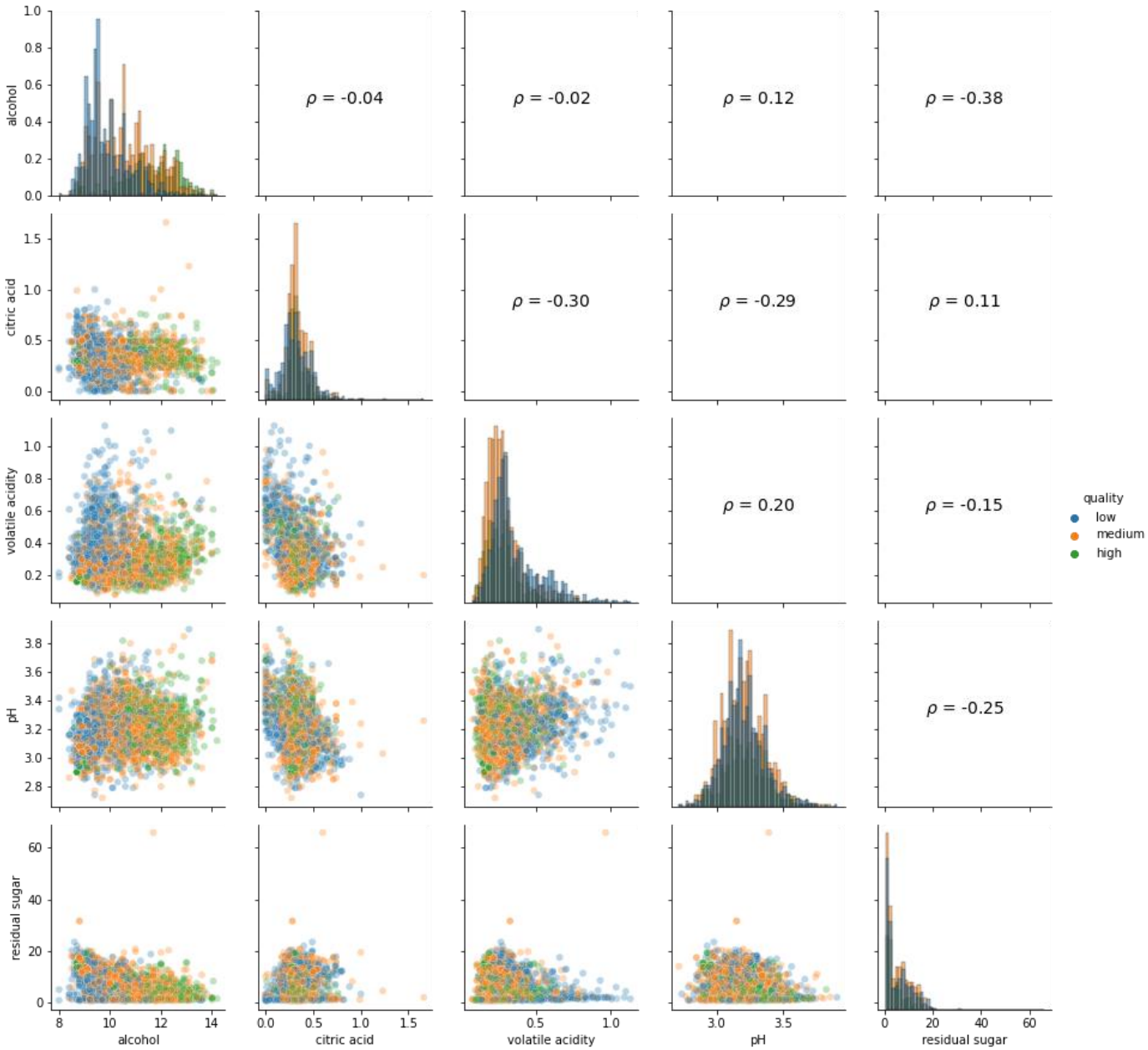


**Figura 2**

**Parte 2 - Mistura de Gaussianas como classificador - valor desta seção:45 %**

Faça download do arquivo "Part2.tsv" da plataforma ECLASS. Neste arquivo, cada ponto tem 2 características (os primeiros 2 valores) e um rótulo de classe associado (o último valor). Nesta parte, você construirá um classificador Bayesiano baseado no modelo de mistura de Gaussianas.

Estas são as etapas que você precisará implementar:

a) Carregar os dados em um DataFrame de pandas
b) Fazer um gráfico de dispersão dos dados, usando cores diferentes para indicar pontos de diferentes classes.
c) Utilizar a implementação do scikit-learn da Mixture of Gaussians para implementar um classificador bayesiano para os dados (use o teorema de Bayes).

*Dica 1: tome cuidado para que a implementação do scikit-learn da Mixture of Gaussians é orientada para o clustering (que é o aprendizado não supervisionado) e não para a estimativa de densidade.*

*Dica 2: tome cuidado para que a implementação da Mixture of Gaussians pelo scikit-learn fornece a log-verosimilhança (não a verosimilhança!) dos dados vindo do modelo. Você precisará levar isso em conta antes de aplicar o teorema de Bayes.*

**BÔNUS (6% de marcas extras):** Compatibilizar sua implementação com a API do estimador scikit-learn: https://scikit-learn.org/stable/developers/develop.html#rolling-your-own-estimator

**Parte 3 - Seleção do Modelo - valor desta seção: 30 %**

Esta peça usa os mesmos dados que você já usou na Parte 2 (ou seja, o arquivo "Part2.tsv"). Nesta parte, você comparará a performance de generalização de diferentes classificadores com relação a diversas medidas de performance.
As performances de generalização serão mensuradas em um conjunto de testes constituído por 25% dos dados.

Os classificadores são:

- implementação da Naive Bayes pela scikit-learn
- implementação do K-nearest-neighbour pela scikit-learn
- um modelo muito simples que classifica os pontos de dados usando apenas os priors de classe *(Dica: similar ao que você fez no laboratório da semana 5)*
- um classificador bayesiano que adequa uma única Gaussiana para cada classe. *(Dica: similar ao que você fez no laboratório da semana 6)*
- o classificador Bayesiano com base no modelo Mistura de Gaussianas que você implementou na Parte 2 *(Note: se você não conseguiu implementar o modelo Mixture of Gaussians na parte 2, então simplesmente omita esse modelo da sua análise).*

As métricas de performance são (todas implementadas no scikit-learn):

- Accuracy
- Precision
- Recall
- AUC-ROC
- AUC-PR

Seu código irá:

a) Imprima uma tabela com os valores da métrica de performance (nas colunas) para cada modelo (nas linhas) no conjunto de teste.
b) Faça uma única figura contendo 2 sub-gráficos. No primeiro sub-gráfico, você plotará as curvas de ROC de cada modelo. No segundo sub-gráfico, você plotará as curvas de PR para cada modelo.
c) Para cada modelo, faça uma figura com a fronteira de decisão do modelo.

**BÔNUS (4% de marcas extras):** Transforme suas implementações dos modelos gaussianos anteriores e multivariados em estimadores de scikit-learn.