

# CARD 4 - Relatório: Principais Bibliotecas e Ferramentas Python para Aprendizado de Máquina (I)

As aulas deste card são de um curso da Udemy ('Python para Data Science e Machine Learning'), no card são indicados 4 sessões do curso a serem vistas, o presente relatório visa apresentar todos os conhecimentos obtidos nestas seções e será dividido em tópicos referentes a cada seção.

## - Seção 3 (Jupyter Notebook)

Esta seção visa apresentar um ambiente de trabalho conhecido como Jupyter. Diferente de um código normal em Python, onde você tem apenas comentários e códigos seguidos, o Jupyter oferece a opção de dividir seus códigos em módulos e executá-los por partes. Além disso, permite adicionar Markdown (marcação), possibilitando fazer comentários mais elaborados, com títulos e tópicos, entre outros recursos. Isso é uma ótima opção para deixar o código mais agradável. Vale ressaltar que o arquivo gerado pelo Jupyter é do tipo 'ipynb'.

## - Seção 5 (Numpy)

Em resumo, esta parte do curso busca apresentar a biblioteca de Python conhecida como NumPy (Numerical Python), que é uma das bibliotecas mais importantes para computação científica em Python. Ela fornece suporte para arrays e matrizes de grandes dimensões, junto com uma coleção de funções matemáticas para operar nesses arrays de maneira eficiente. NumPy é amplamente utilizada para realizar cálculos numéricos rápidos, manipulação de dados e criação de algoritmos científicos, tornando-se uma ferramenta fundamental em áreas como ciência de dados, aprendizado de máquina, análise estatística e engenharia.

Abaixo, segue a lista dos principais comandos aprendidos referentes a essa biblioteca:

- **np.array**: usada para criar arrays a partir de listas ou outras estruturas de dados.
- **np.arange**: cria um array contendo uma sequência de números igualmente espaçados dentro de um intervalo especificado.
- **np.zeros** ou **np.ones**: cria uma matriz ou vetor de 0's ou de 1's.
- **np.eye**: cria uma matriz identidade.
- **np.random (rand, randn, randint)**: sorteia números com diferentes distribuições.
- **np.sqrt** e **np.exp**: raiz quadrada e exponenciação de um array, respectivamente.
- **np.mean**: calcula a média de um array.
- **np.std**: calcula o desvio padrão de um array.
- **np.sin**: calcula o seno dos elementos de um array.
- **np.max** e **np.min**: retornam o maior e o menor elemento de um array, respectivamente.

Nas aulas, são mostradas funções como a reshape, que transforma um vetor em matriz, e outras operações com vetores e matrizes, assim como explicações sobre indexação. No entanto, como o foco era citar as funções da biblioteca NumPy, mencionei apenas as funções principais.

## - Seção 6 (Pandas)

Esta seção tem por objetivo abordar a biblioteca Pandas do Python, que é uma ferramenta de código livre para manipulação e análise de dados. É amplamente utilizada por cientistas de dados e analistas devido às suas capacidades rápidas e eficientes de manipulação de dados. A biblioteca se destaca principalmente por permitir a criação de DataFrames, que são estruturas bidimensionais (tabelas), semelhantes a planilhas do Excel, capazes de conter diferentes tipos de dados e com rótulos para linhas e colunas.

Segue abaixo uma lista dos principais comandos aprendidos nesta aula utilizando a biblioteca Pandas:

- **pd.Series:** cria uma série, que é um array unidimensional rotulado;
- **pd.DataFrame:** cria um DataFrame, uma estrutura de dados bidimensional semelhante a uma tabela com linhas e colunas rotuladas, utilizado para armazenar e manipular conjuntos de dados complexos de forma organizada e eficiente;
- **dataframe.drop:** remove linhas ou colunas especificadas de um DataFrame (requer inplace=True para alterar diretamente a variável do DataFrame);
- **dataframe.loc:** acessa e modifica dados no DataFrame usando rótulos (nomes) das linhas e colunas;
- **dataframe.reset\_index:** redefine o índice do DataFrame para o padrão (inteiros sequenciais);
- **dataframe.set\_index:** define uma ou mais colunas do DataFrame como novo índice;
- **pd.MultiIndex:** cria um índice hierárquico para um DataFrame ou Series, permitindo índices com múltiplos níveis, útil para trabalhar com dados multidimensionais ou agrupados;
- **dataframe.xs:** extrai uma fatia de um DataFrame usando um valor específico de um nível do índice, útil para acessar dados em DataFrames MultiIndex;
- **dataframe.dropna:** exclui dados faltantes (NaN) de um DataFrame;
- **dataframe.fillna:** substitui dados faltantes (NaN) de um DataFrame por um valor escolhido, como a média dos outros;
- **dataframe.groupby:** permite agrupar dados de um DataFrame com base em valores de coluna específicos, facilitando operações agregadas como soma, média ou contagem em cada grupo separadamente;
- **pd.concat:** concatena (ou combina) objetos ao longo de um eixo específico de um DataFrame;
- **pd.merge:** combina (ou funde) DataFrames com base em colunas comuns;
- **dataframe.join:** combina DataFrames usando índices ou colunas comuns;
- **dataframe.unique:** seleciona dados únicos de uma coluna que não possuem cópias;
- **dataframe.nunique:** retorna o número de valores únicos em cada coluna do DataFrame;
- **dataframe.value\_counts:** retorna uma contagem de valores únicos na série, por padrão em ordem decrescente;
- **dataframe.apply:** aplica uma função ao longo de um eixo do DataFrame;
- **dataframe.sort\_values:** ordena os valores do DataFrame;
- **dataframe.isnull:** determina os valores nulos no DataFrame;
- **pd.read\_csv, pd.read\_excel, pd.read\_html:** utilizadas para ler diferentes tipos de dados em estruturas de DataFrame, convertendo os dados nos respectivos formatos para um DataFrame do Pandas, facilitando a análise e manipulação de dados.

Este é um resumo geral dos principais comandos da biblioteca Pandas aprendidos nas aulas. Vale destacar que também foram ensinados diversos outros conceitos sobre manipulação e indexação de DataFrames, sendo esses comandos os mais destacados e diferentes de outras estruturas aprendidas até o momento.

## - **Seção 7 (Exercícios Pandas)**

Nesta seção, são apresentados dois exemplos de aplicação do que foi aprendido na seção 6 (Pandas). Em ambos os vídeos, são demonstradas aplicações intuitivas e importantes para aprender como utilizar o Pandas com um banco de dados pré-existente. Os vídeos utilizam as principais funções e conhecimentos aprendidos anteriormente para reforçar o conteúdo.

## **Conclusão:**

Esse relatório conclui que as bibliotecas Pandas e NumPy desempenham papéis fundamentais no campo de Machine Learning e Data Science. O Pandas oferece estruturas de dados flexíveis e poderosas, como DataFrames, facilitando a manipulação e análise de dados complexos. Enquanto isso, o NumPy fornece suporte essencial para operações numéricas eficientes e arrays multidimensionais, essenciais para cálculos matemáticos e científicos.

Juntas, essas bibliotecas permitem explorar, limpar, transformar e modelar dados de maneira eficiente. Elas podem ser amplamente utilizadas para preparar conjuntos de dados, extrair características relevantes e alimentar algoritmos de Machine Learning, contribuindo significativamente para a precisão e eficácia dos modelos desenvolvidos. Em suma, Pandas e NumPy são pilares essenciais que sustentam a infraestrutura analítica e preditiva no mundo moderno de Data Science e Machine Learning.