Aula 2 – Meu primeiro projeto de MDE

Suponha que você foi contratado para ajudar a escola 'X' a melhorar a taxa de aprovação em uma disciplina específica, que tem apresentado resultados abaixo do esperado nos últimos anos. A escola 'X' disponibilizou uma base de dados aberta que contém informações dos alunos que se matricularam na disciplina nos últimos 5 anos, incluindo informações sobre suas notas nas disciplinas anteriores, gênero, idade, carga horária semanal de estudo, entre outras.

Você decide usar uma técnica de mineração de dados para construir um modelo que possa prever, com base nas informações disponíveis, se um aluno terá sucesso (aprovado) ou não na disciplina em questão. Para isso, você vai utilizar uma técnica de classificação binária.

Vamos utilizar a base de dados "*aula2.csv*" disponível no e-aula e adaptada da base de dados em https://archive.ics.uci.edu/ml/datasets/Student+Performance do repositório *UCI Machine Learning*. Essa base de dados contém informações sobre alunos de matemática de uma escola secundária em Portugal. Vamos criar um modelo que preveja se um aluno terá sucesso ou não na disciplina de matemática com base em algumas variáveis como gênero, idade, número de reprovações, carga horária semanal de estudo, entre outras.

Na página seguinte segue algumas orientações para o desenvolvimento do trabalho.

No entanto, eu indico que você tente **primeiramente resolver o problema por sua conta e apenas após leia as orientações**. Dessa forma, a sua leitura já vai ser mais engajada pelas dificuldades experimentadas e você vai se dar conta melhor do que você fez diferentemente e que novos comandos são úteis.

1) Entendendo o problema

A partir do enunciado você sabe que precisa prever se o aluno vai ter sucesso na disciplina de matemática ou português a partir de notas parciais e de outros dados.

- Você sabe que é um problema de classificação binária. Você precisa classificar o aluno como aprovado (representado por "1") ou reprovado (representado por "0 (zero)").
- A coluna 'approved' da base de dados (arquivo 'aula2.csv') contém essa informação.

2) Importando os dados:

2.1) Você pode fazer o download do arquivo 'aula2.csv', disponível no tópico aula2 do e-aula. Abra o arquivo `aula2.csv` no Google Colab. Esse arquivo contém a nota dos alunos em matemática, além de outros dados. Após, use o comando `df = pd.read_csv(filepath, delimiter=';')` (onde `df` pode ser qualquer nome que você escolheu para o dataframe que receberá os dados importados) do Pandas, `filepath` é o caminho do arquivo, e `delimiter=';'` é o delimitador csv dos dados (geralmente é , ou ;).

```
df = pd.read_csv(filepath, delimiter=';')
```

3) Entendendo os dados a partir de uma análise descritiva:

Vamos agora fazer algo que não vimos ainda nessa disciplina: uma análise descritiva dos dados. Essa análise vai nos ajudar a entender melhor os dados para decidir que tipo de ajustes seria necessário realizar.

3.1) Primeiramente, você pode usar o comando `head(numero_linhas)` para ter uma visualização rápida dos dados. Ele mostra as primeiras `numero_linhas` dos dados importados.

- Essa visualização vai lhe ajudar a ter uma ideia dos dados. Você também pode olhar o arquivo `student.txt` que veio junto no arquivo zipado para ter uma descrição dos dados.

3.2) Você pode também usar `print(df.dtypes)` para saber os tipos dos dados e `print(df.shape)` para ver o formato do dataframe (número de instâncias X número de variáveis). Você vai ver que o Pandas salvou as strings com o tipo `object` e os números como inteiros. Isso não vai ser um problema para nós, mas se desejar você pode usar o parâmetro `dtype` do comando `read_csv()` para configurar os tipos desejados para as colunas.

- Com esses comandos, você vai ter mais informações complementares sobre os dados que podem lhe ajudar a decidir que variáveis você quer utilizar.

3.3) Você também pode usar o comando `df.describe()` para visualizar uma descrição qualitativa dos dados. Esse comando vai retornar os valores máximo e mínimo, média, desvio padrão e percentis para cada uma das variáveis. Também retorna a quantidade de dados em cada coluna. Talvez você queira usar o comando `pandas.set_option('precision', 2)` para configurar a precisão dos números, o que vai mudar a visualização, como a seguir:

```
pd.set_option('precision', 2)
df.describe()
```

- Esses comandos ajudam a ver os valores dos dados e a sua distribuição. Podem lhe ajudar posteriormente no processamento. Por exemplo, você vai perceber que o valor no G1 é 19.
- Aqui também podemos ver se temos valores faltantes pela linha `count`. Todas as colunas têm 395 células, assim não temos dados faltantes.

4) Pré-processando os dados

Agora que você já visualizou os dados, vamos trabalhar neles, começando pelo pré-processamento.

4.1) Vamos primeiramente começar removendo as instâncias com dados faltantes com o comando `dropna` (videoaula 2.2). Como vimos que nossa base de dados não tem dados faltantes (item 3.3 acima), esse é um comando opcional para esse problema. Mas você pode querer inseri-lo mesmo assim caso no futuro deseje treinar o mesmo modelo com outra base de dados que você não tem certeza se vai ter ou não dados faltantes.

4.2) Você pode ainda inserir o código para remover as instâncias duplicadas, caso existam.

4.3) Não esqueça de verificar se você tem variáveis categóricas e realizar o processo de *one-hot-encoding*, como explicado nas videoaulas 2.2 e 2.3.

Obs: Ainda faltam alguns processamentos importantes para o algoritmo de regressão logística funcionar melhor. Mas como não quero sobrecarregar a aula de hoje, vamos deixar para a próxima aula e assim vocês podem estudar a diferença. ;-)

5) Treinando o modelo

Agora você está pronto para treinar o seu modelo. Você pode fazer isso usando o algoritmo de Regressão Logística visto na videoaula 2.3.

6) Avaliando o seu modelo

Agora você pode avaliar o seu modelo. Para avaliar você tem que fazer predições com o conjunto de teste e usar as métricas para comparar o valor predito com o valor real.

Não esqueça de treinar os modelos e avaliar para os diferentes cenários de pré-processamento.

- Insira os resultados encontrados com uma descrição em texto no seu código.
- Obs: O tipo de avaliação que estamos fazendo ainda não é a ideal. Na aula 9, aprenderemos a trabalhar com cross-validação e falaremos também de conjunto hold-outo e outras técnicas para melhorar nossas avaliações de modelos criados.

7) Submeta seu programa para avaliação pela professora

- Submeta seu código (arquivo ".ipynb") no e-aula na tarefa de submissão que se encontra na Aula 2 – Atividade Prática, até a data limite estabelecida.