

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PIAUÍ</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ</p> <p>Curso: ADS</p> <p>Disciplina: Tópicos Especiais em Sistemas de Informação</p> <p>Professor: Ely</p>
--	--

1. (1,5) Acesse o site <http://www.tce.pi.gov.br> e usando beautiful soap baixe e resolva as questões abaixo:

Nota: necessariamente use a biblioteca requests-cache para evitar requisições desnecessárias

- (2,0) Baixe o título das 5 notícias da página inicial;
- (0,5) Forneça uma opção um menu com escolha de 1 a 6 correspondendo às 5 matérias e uma sexta opção de sair;
- (2,0) Caso o usuário escolha uma das opções, o texto da mensagem deve ser baixado e exibido.
- (0,5) Após exibir a matéria, deve-se dar a opção para exibir o menu novamente.

Nota: necessariamente use a biblioteca requests-cache para evitar requisições desnecessárias



2. Usando o dataset sobre os sobreviventes do titanic, realize as seguintes operações:
 - Acesse a base de dados do dataset do Titanic (<https://www.kaggle.com/c/titanic>) e baixe seus dois datasets: train e test.
 - **Entendendo os atributos:**
 - PassengerId: id do passageiro;

- Survived: se sobreviveu ou não
- PClass: classe na qual o passageiro estava;
- SibSp: número de irmãos no barco;
- Parch: Número de pais ou filhos no barco;
- Ticket: número do ticket;
- Fare: valor da tarifa;
- Cabin: número da cabine;
- Embarked: porto onde embarcou;

Resolva as seguintes questões após ler o arquivo train.csv:

- (0,5) Imprima os sobreviventes;
- (0,5) Com base na questão anterior, imprima o percentual de sobreviventes. Use a função len()
- (0,5) Baseado no exemplo anterior, exiba o percentual de mulheres sobreviventes por classe. Pesquise como usar o operador & para juntar as duas condições.
- (0,5) Remova as linhas cujo valor da coluna Embarked for nula (obs: essa não é a melhor opção);
- (0,5) Altere os valores dos campos sex para 0 e 1
- (0,5) Sabendo que podemos criar uma nova coluna dessa forma:

```
train_df['FamilySize'] = train_df['SibSp'] + train_df['Parch'] + 1
```

Ou seja, criamos a coluna tamanho da família somando o número de filhos + número de pais ou filhos + a própria pessoa.

Crie uma coluna chamada IsAlone que recebe 1 se o 'FamilySize' for maior que 1 e 0 caso 'FamilySize' seja 1

- (0,5) Crie variáveis categóricas para a coluna Embarked
- (0,5) Crie o dataframe x com as colunas: Sex, IsAlone, Embarked, PClass e o dataframe y com a coluna survived
- (0,5) Treine o modelo com 80% de dados para treino e 20% para teste e execute o algoritmo. Imprima o score
- A coluna PClass possui valores distintos, pesquise se:
 - Retirá-la influi no score;
 - Se transformá-la em variáveis categóricas tipo 'A', 'B', 'C' e gerar "dummies influi no score .