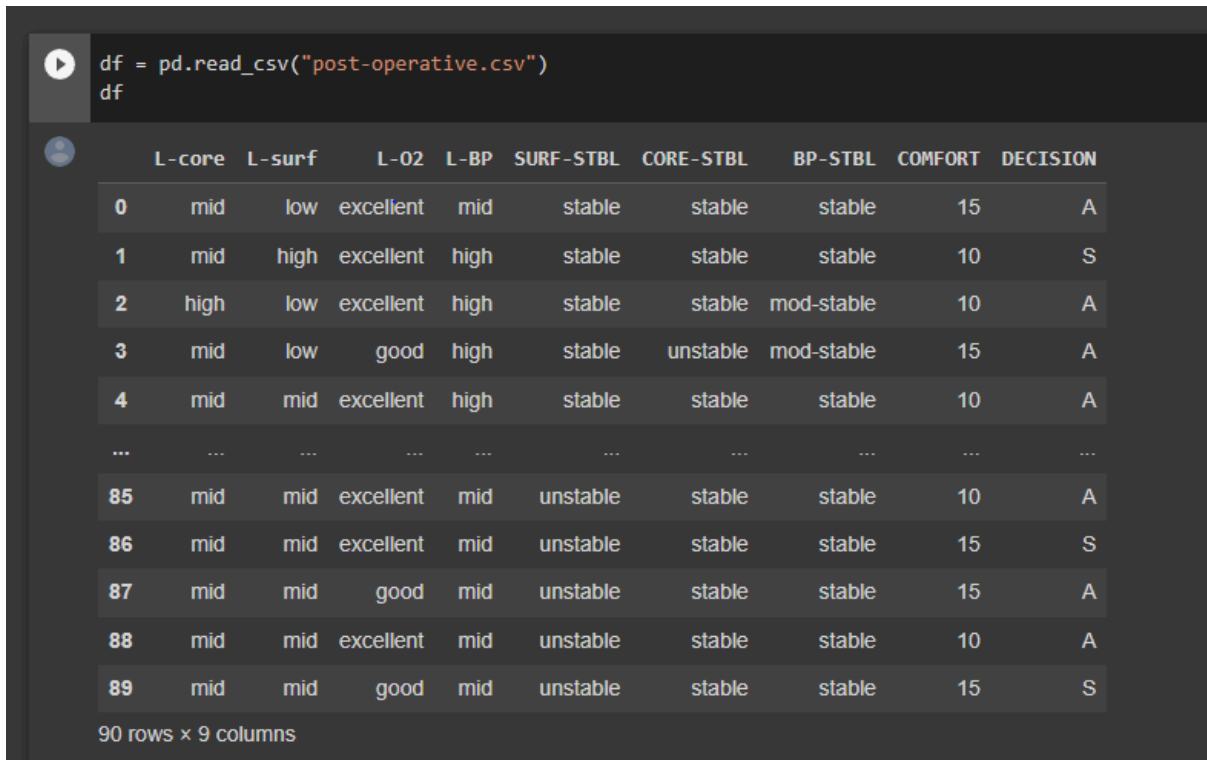


Estrutura de relatório da atividade N1 de Inteligência Artificial II

Descrição geral do dataset:

Os dados contidos neste relatório foram retirados de pacientes no seu período pós-operatório, dessa forma tenta-se estipular o progresso do estado de saúde, prevendo uma possível piora.

Neste dataset são nos apresentados 9 atributos sendo eles:



```
df = pd.read_csv("post-operative.csv")
df
```

	L-core	L-surf	L-O2	L-BP	SURF-STBL	CORE-STBL	BP-STBL	COMFORT	DECISION
0	mid	low	excellent	mid	stable	stable	stable	15	A
1	mid	high	excellent	high	stable	stable	stable	10	S
2	high	low	excellent	high	stable	stable	mod-stable	10	A
3	mid	low	good	high	stable	unstable	mod-stable	15	A
4	mid	mid	excellent	high	stable	stable	stable	10	A
...
85	mid	mid	excellent	mid	unstable	stable	stable	10	A
86	mid	mid	excellent	mid	unstable	stable	stable	15	S
87	mid	mid	good	mid	unstable	stable	stable	15	A
88	mid	mid	excellent	mid	unstable	stable	stable	10	A
89	mid	mid	good	mid	unstable	stable	stable	15	S

90 rows x 9 columns

Coluna 1 L-CORE: Temperatura interna do paciente em °C

High é temperatura Alta maior que 37°C

Mid é temperatura Média entre 36° e 37°C

low é temperatura Baixa menor que 36° C

Coluna 2 L-SURF: Temperatura superficial do paciente em °C

High é temperatura Alta maior que 36°C

Mid é temperatura Média entre 35 e 36.5°C

low é temperatura Baixa abaixo de 35°C

Coluna 3 L-O2: Saturação de Oxigênio no sangue em %.

Excellent é Excelente quando igual ou acima de 98

Good é Bom quando está entre 90 e 97

Fair é Razoável quando está entre 80 e 89

Poor é Ruim quando é menor que 80

Coluna 4 L-BP: Última medição da pressão arterial.

High é Alto quando está maior que 130/90

Mid é Médio quando está entre 130/90 e 90/70

Low é Baixo quando está abaixo de 90/70

Coluna 5 SURF-STBL: Temperatura estável superficial do paciente.

Stable é Estável (melhor estado)

Mod-stable é Moderado (estado médio)

Unstable é Instável (estado ruim)

Coluna 6 CORE-STBL: Temperatura estável interna do paciente.

Stable é Estável (melhor estado)

Mod-stable é Moderado (estado médio)

Unstable é Instável (estado ruim)

Coluna 7 BP-STBL: Estabilidade da pressão arterial do paciente

Stable é Estável (melhor estado)

Mod-stable é Moderado (estado médio)

Unstable é Instável (estado ruim)

Coluna 8 COMFORT: Nota de conforto do paciente após a alta

Um número inteiro entre 0 e 20.

Coluna 9 DECISION: Decisão de quitação:

I - Paciente encaminhado para unidade de terapia intensiva

S - Paciente preparado para receber alta

A - Paciente encaminhado para o andar geral do hospital

Fase de pré-processamento:

```
1 df.info()
```

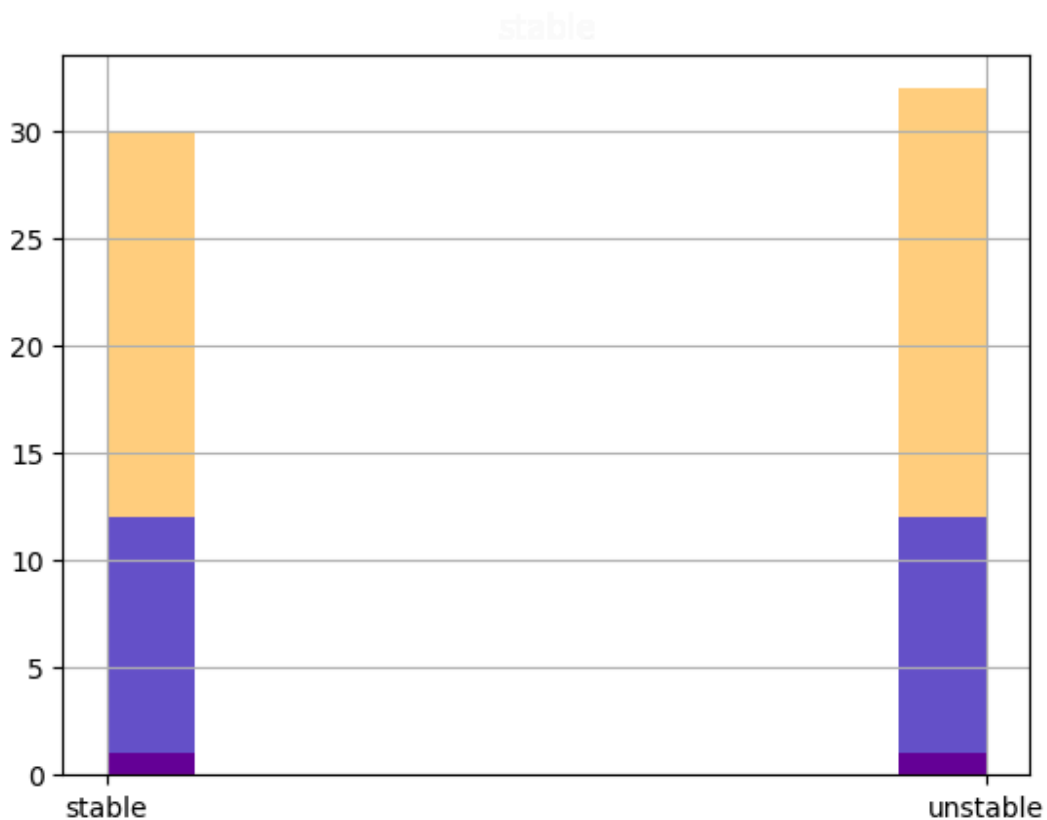
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 9 columns):
Column Non-Null Count Dtype
--- ---
0 L-core 90 non-null object
1 L-surf 90 non-null object
2 L-O2 90 non-null object
3 L-BP 90 non-null object
4 SURF-STBL 90 non-null object
5 CORE-STBL 90 non-null object
6 BP-STBL 90 non-null object
7 COMFORT 90 non-null object
8 DECISION 90 non-null object
dtypes: object(9)
memory usage: 6.5+ KB

Ao visualizarmos os dados presentes neste dataset percebemos que estamos visualizando informações brutas de todos os pacientes de pós-operatório com os sinais vitais. Dependendo do modo de filtramento que você quiser, todas as informações podem ser relevantes no momento.

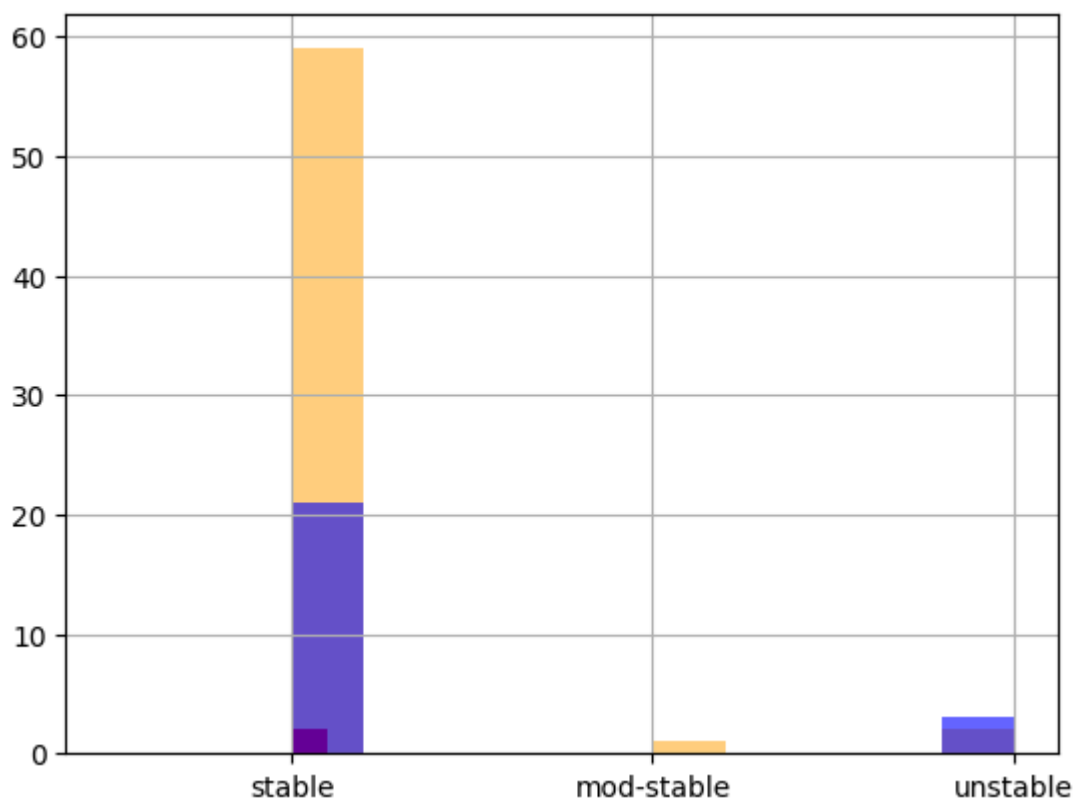
Analizando a situação dos pacientes pós operatório e a decisão final:

Foi separado os três tipos de decisão: se o paciente ficou no hospital, foi para a UTI ou saiu do hospital com relação a estabilidade da temperatura superficial, temperatura interna e pressão arterial após a cirurgia.

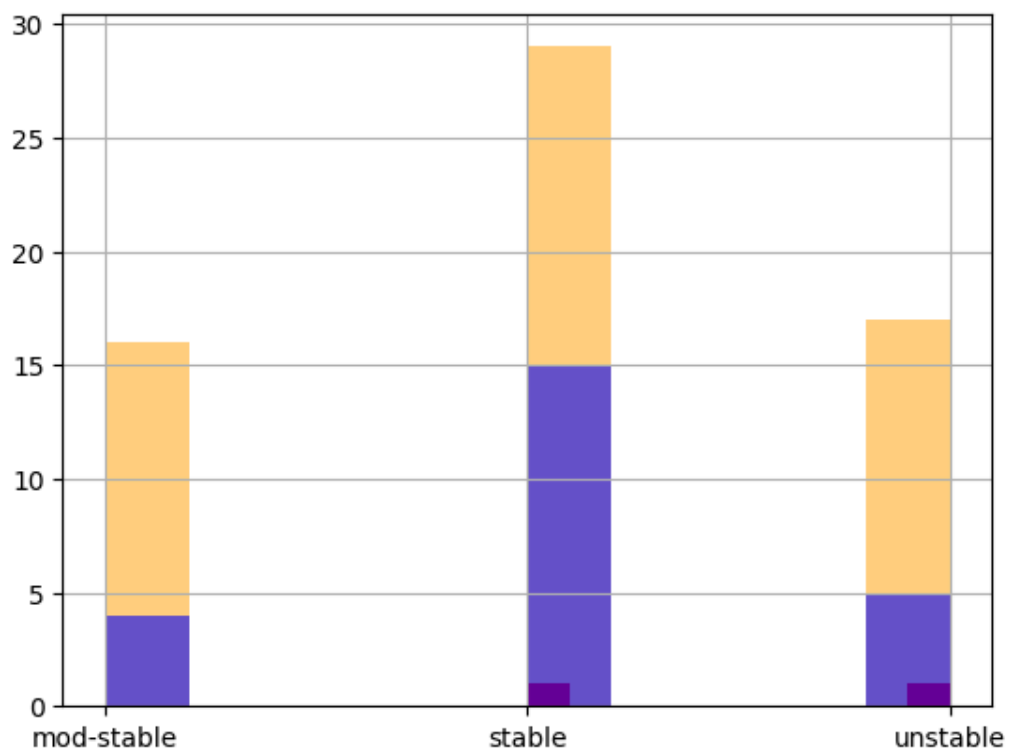
```
for column in columns:
    plt.title(column)
    dataset_A[column].hist(color = "orange", alpha=0.5)
    dataset_I[column].hist(color = "red")
    dataset_S[column].hist(color = "blue", alpha=0.6)
    plt.show()
```



Aqui vemos que não foram encontradas grandes diferenças de quem ficou no hospital, foi para a casa ou para a UTI. Ambos os pacientes cujo sua temperatura superficial ficou estável ou instável tiveram casos semelhantes de permanência ou não no hospital



Já aqui mostra que na maioria dos casos a temperatura interna do paciente está estável e os dados mostram alguns casos em que o paciente ficou no hospital com a temperatura interna moderada, saiu ou foi para UTI mesmo estando instável



Comparando a pressão arterial após a cirurgia, grande parte pertence ao grupo que ficou no andar geral do hospital independente da situação. Vale destacar que grande parte dos que tiveram alta estavam com a pressão estável.

Limpeza e normalização

As primeiras células excluídas, foram as que haviam ausência de dados.

Após a visualização dos dados brutos do Dataset restantes, foi notado que o conforto relatado pelo paciente após cirurgia, não traz importância na decisão final tomada, ou seja, se o mesmo foi liberado, internado ou movido de ala. Por esse motivo foi tratado como informação irrelevante para as futuras análises, sendo removido.

```
df_clean = df.drop([ 'COMFORT'], axis = 1)
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 90 entries, 0 to 89
Data columns (total 8 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   Lcore       90 non-null    object  
 1   Lsurf       90 non-null    object  
 2   LO2        90 non-null    object  
 3   LBP        90 non-null    object  
 4   SURFSTBL   90 non-null    object  
 5   CORESTBL   90 non-null    object  
 6   BPSTBL     90 non-null    object  
 7   DECISION   90 non-null    object  
dtypes: object(8)
memory usage: 5.8+ KB
```

Fase de Análise

Primeiramente utilizamos o label encoding para transformar as informações em texto para número. Para utilizar os algoritmos

- KNN**: tenta classificar cada amostra em um conjunto de dados avaliando sua distância até o vizinho mais próximo. Se os vizinhos mais próximos pertencem em sua maioria a uma classe, a amostra em questão será agrupada nela.

- DecisionTree**: a partir de um conjunto de regras é classificado uma classe dada as características.

•NAIVE BAYES seu funcionamento consiste em: para calcular a previsão, o algoritmo primeiro define uma tabela de probabilidades, que contém a frequência dos preditores em relação às variáveis de saída. O cálculo final então considera o mais provável para sugerir uma solução.

```

accuray = accuracy_score(y_teste, y_prev)
pres= precision_score(y_teste, y_prev, average=None)
res =recall_score(y_teste, y_prev, average=None)
print("Modelo KNN")
print("Acuracia:", accuray)
print("Precisão:", pres)
print("Revogação", res)

```

Modelo KNN

Acuracia: 0.5555555555555556

Precisão: [0.71428571 0. 0.]

Revogação [0.71428571 0. 0.]

```

accuray = accuracy_score(y_teste, y_pred)
pres= precision_score(y_teste, y_pred, average=None)
res =recall_score(y_teste, y_pred, average=None)
print("Modelo Arvore Decisão")
print("Acuracia:", accuray)
print("Precisão:", pres)
print("Revogação", res)

```

Modelo Arvore Decisão

Acuracia: 0.5185185185185185

Precisão: [0.75 0. 0.18181818]

Revogação [0.57142857 0. 0.4]

```

accuray = accuracy_score(y_teste, y_pred)
pres= precision_score(y_teste, y_pred, average=N
res =recall_score(y_teste, y_pred, average=None)
print("Modelo NAIVE BAYES")
print("Acuracia:", accuray)
print("Precisão:", pres)
print("Revogação", res)

```

Modelo NAIVE BAYES

Acuracia: 0.7407407407407407

Precisão: [0.79166667 0. 0.33333333]

Revogação [0.9047619 0. 0.2]

Os três algoritmos apresentaram acurácia relativamente semelhantes, com exceção do Naive Bayes.. A precisão e revocação apresentam resultados satisfatórios apenas na primeira coluna, nas demais ou está muito baixo ou o algoritmo não consegue classificar

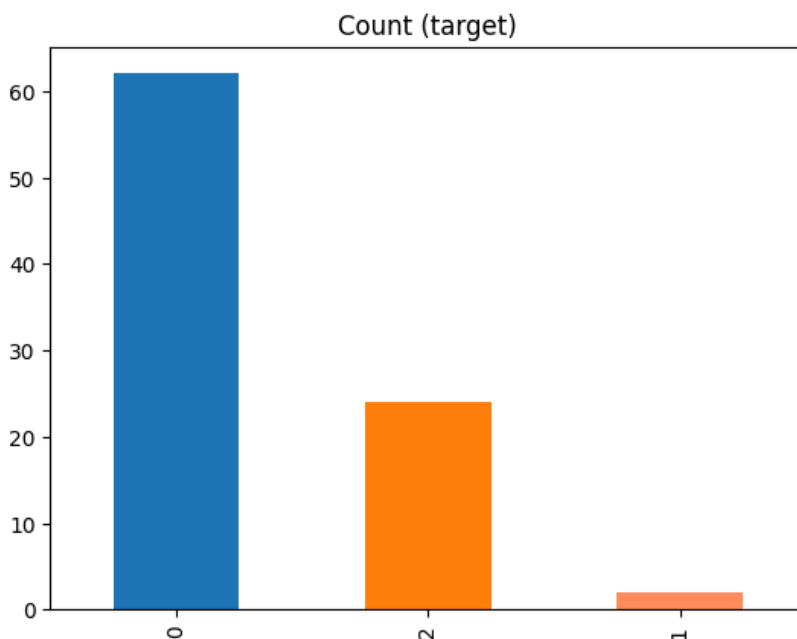
Balanceando o Dataset

```
dataset_clean.DECISION.value_counts()
```

```
0    62
2    24
1     2
Name: DECISION, dtype: int64
```

```
#rebalanceando o dataset
target_count = dataset_clean.DECISION.value_counts()
print('Class 0:', target_count[0])
print('Class 1:', target_count[1])
print('Class 2:', target_count[2])
target_count.plot(kind='bar', title='Count (target)', color = ['#1F77B4', '#FF7F0E', '#FF8F5E']);
```

```
Class 0: 62
Class 1: 2
Class 2: 24
```



Grande parte do dataset estão presentes os pacientes que ficaram em um andar geral do hospital.


```
#rebalanceando o dataset
from imblearn.over_sampling import RandomOverSampler, SMOTE
from collections import Counter
ros = RandomOverSampler()
X_train_ros, y_train_ros= ros.fit_resample(x_treino, y_treino)
print(sorted(Counter(y_train_ros).items()))
```

```
[(0, 41), (1, 41), (2, 41)]
```

```
X_train_ros.shape
```

```
(123, 7)
```

Utilizando Oversampling foi igualado os três tipos de decisões existentes, totalizando o dataset de treino de 61 instâncias para 123.

```
accuray = accuracy_score(y_teste, y_prev_knn)
pres= precision_score(y_teste, y_prev_knn, average=None)
res =recall_score(y_teste, y_prev_knn, average=None)
print("Modelo KNN")
print("Acuracia:", accuray)
print("Precisão:", pres)
print("Revogação", res)
```

```
Modelo KNN
Acuracia: 0.2962962962962963
Precisão: [0.5          0.          0.14285714]
Revogação [0.31578947 0.          0.28571429]
```

```
accuray = accuracy_score(y_teste, y_pred_tree)
pres= precision_score(y_teste, y_pred_tree, average=None)
res =recall_score(y_teste, y_pred_tree, average=None)
print("Modelo Arvore")
print("Acuracia:", accuray)
print("Precisão:", pres)
print("Revogação", res)
```

```
Modelo Arvore
Acuracia: 0.3333333333333333
Precisão: [0.53333333 0.          0.1         ]
Revogação [0.42105263 0.          0.14285714]
```

```
accuray = accuracy_score(y_teste, y_pred_gauss)
pres= precision_score(y_teste, y_pred_gauss, average=None)
res =recall_score(y_teste, y_pred_gauss, average=None)
print("Modelo NAIVE BAYES")
print("Acuracia:", accuray)
print("Precisão:", pres)
print("Revogação", res)
```

```
Modelo NAIVE BAYES
Acuracia: 0.4074074074074074
Precisão: [0.61538462 0.          0.21428571]
Revogação [0.42105263 0.          0.42857143]
```

Fase de conclusão

O dataset antes de balanceado teve o melhor resultado usando Naive Bayes, contudo era eficiente em apenas uma classificação. Os demais algoritmos eram imprecisos e ineficientes.

Após o oversampling no dataset de treino houve uma piora nos parâmetros, especialmente no modelo Naive Bayes, a acurácia por exemplo, caiu de 74% para 40%. Com taxas de acerto tão baixas, os modelos são ineficientes para conseguir prever a tomada de decisão ou mesmo uma atenção maior a pacientes com determinadas características que prejudicam seu estado.