Problema

Objetivo do teste 1
Usar a aba dataset
1. Criar um modelo usando SVM pra prever
a. previsao A (coluna G do dataset) - resultado deve ser numerico
b. previsao B (coluna H do dataset) - resultado deve ser categorico
Importante:
dividir os dados em treino e validacao (80 e20)
1. Balancear as classes antes de criar o modelo
2. Usar as features para tentar prever a. e b.
3. Testar mais de um kernel
4. reportar a acuracia balanceada no conjunto de validacao
Objetivo do teste 2
Usar a aba sensor
1. Escrever uma funcao para encontrar os 4 platores marcados com a seta
2. A funcao deve encontrar o valor do eixo y no inicio e final de cada plato e apresentar o resultado na forma abaixo
plato 1 - inicio em y : 0,20 final em y:0,40
plato 2 - inicio em y : 0,50 final em y: 0,80
plato 2 - inicio em y : 0,50 final em y: 0,80
plato 3 - inicio em y : 0,50 final em y: 0,80
plato 4 - inicio em y : 0,50 final em y: 0,80
Calcular a media de todas as medicoes feitas pelo sensor entre o inicio e final apenas do plato 1 e plato 4
Reportar da seguinte forma
Media dos resultados do plato 1: 0,30 (numero de medicoes)
Media dos resultados do plato 5: 0,60 (e o numero de medicoes)

Observações sobre o teste 1

Se o objetivo é classificar ou predizer um modelo, ficar restrito a apenas um algoritmo não é uma boa estratégia. Sendo assim, eu sempre utilizo vários algoritmos para avaliar qual deles é mais adequado ao problema em questão.
Inicialmente, utilizei os seguintes algoritmos de classificação:

```
classifiers = {"RF": RandomForestClassifier(n_estimators=100),
               "KNN": KNeighborsClassifier(),
               "DTREE": DecisionTreeClassifier(),
               "GNB": GaussianNB(),
               "LRG": LogisticRegression(),
               "ABC": AdaBoostClassifier(),
               "MLP": MLPClassifier(max_iter=500),
               "QDA": QuadraticDiscriminantAnalysis(store_covariance=True),
               "SVM": SVC(probability=True),
               "SGD": SGDClassifier(loss="hinge", penalty="l2", max_iter=5)
               }
```

O que apresentou melhor resultado inicial para o problema foi o "KNN": KNeighborsClassifier().

Dividir os dados em 80/20 é uma boa estratégia para dados balanceados mas quando há restrição do número de exemplos da classe minoritária pode haver problemas. Utilizamos a técnica de Cross Validation para validar o modelo e separação dos dados de forma estratificada, ou seja, garantimos que haverá, nos subconjuntos dos dados, exemplos das duas classes (minoritária ou positiva e majoritária ou negativa).
O balanceamento, neste caso, é obrigatório pois há apenas 4 exemplos da classe minoritária, o que no meu ponto de vista, é irrisório para criarmos um modelo satisfatório.
Mas, analisando o problema, minha proposta de solução é:

1. codificar as features categóricas com técnicas mais sofisticadas para garantir um melhor aprendizado. Utilizei LabelEncoder para as classes e TargetEncoder para as demais features categóricas.
2. Eu testei a seguinte estratégia: como não há muitos exemplos disponíveis da classe positiva, peguei uma pequena amostra contendo todos os dados da classe positiva e alguns dados da classe negativa e apliquei o RandonOversampling para gerar mais dados da classe positiva. Em seguida, juntei estes dados novos com os dados originais (retirando os exemplos repetidos) e apliquei outras técnicas de oversampling como SMOTE, SVMSMOTE,DTOSMOTE, Geometric-SMOTE.
3. Até que tive bons resultados mas não consegui validar o modelo de forma aceitável, uma vez que não há exemplos extras disponíveis para validação.
4. O que percebi das classes é que há um dataset que diz “Bom” ou “Ruim” e outra numérica. Pude identificar que a Previsão A também é categórica pois 0.955 ocorre na maior parte das vezes (classe dominante) e os outros números são da classe minoritária. Portanto, pode parecer um resultado numérico (Regressão) mas não é. Trata-se de uma variável categórica, uma string e foi codificada também.
5. Percebi que na Feature 4 todos os exemplos são únicos. Data e hora pode ter relação com o clima, temperatura e até mesmo quem está operando as máquinas da produção. Foram codificados também. Resumindo, a única feature numérica é a Feature 3.
6. Não fiz a parte 2 dos sensores.