|  |  |
| --- | --- |
| UNIVERSIDADE FEDERAL DO PARÁ  POS-GRADUACAO EM GENETICA E BIOLOGIA MOLECULAR | |
| Área de Concentração: Genética e Biologia Molecular |  | |
| Linha de Pesquisa: Bioinformática  Disciplina: TÓPICOS AVANÇADOS III: MACHINE LEARNING APLICADO A BIOINFORMÁTICA  Docente: GILDERLANIO SANTANA DE ARAUJO  Discente: MARCO ANTONIO ALVES CORDOVIL |  | |

**PROJETO:** Predição de Subtipos de Câncer Gástrico por Meio de Análise de Dados Genômicos e Clínicos Utilizando Aprendizado de Máquina Supervisionada.

**OBJETIVO**:

O objetivo deste projeto é desenvolver um modelo de aprendizado de máquina supervisionada utilizando a técnica de Algoritmo Random Forest, que opera utilizando uma "floresta" de árvores de decisão durante o treinamento e é capaz de predizer subtipos de câncer gástrico com base em dados genômicos e clínicos.

O câncer gástrico, também conhecido como câncer de estômago, é uma doença complexa que pode apresentar diferentes subtipos com base em características genômicas, clínicas. Abaixo estão os subtipos de câncer gástrico que o modelo, Ao prever subtipos moleculares, o modelo mediante pode ajudar a personalizar o tratamento para pacientes, melhorar prognósticos e descobrir novos insights a partir dos dados.

**Subtipos Moleculares de Câncer Gástrico**

1. CIN (Chromosomal Instability)
   * Descrição: Este subtipo é caracterizado por uma alta taxa de instabilidade cromossômica, o que leva a um número anormal de cópias de cromossomos e muitas vezes resulta em múltiplas aneuploidias (número anormal de cromossomos).
   * Significado Clínico: Tumores CIN frequentemente têm amplificações em oncogenes como ERBB2 (HER2), amplificações no receptor do fator de crescimento epidérmico (EGFR), e em outros oncogenes. Esse subtipo está associado a um mau prognóstico.
2. EBV (Epstein-Barr Virus)
   * Descrição: Este subtipo é caracterizado pela presença do vírus Epstein-Barr nas células tumorais.
   * Significado Clínico: Tumores EBV-positivos tendem a ter uma alta carga mutacional e são frequentemente associados a um infiltrado inflamatório abundante. Esses tumores têm melhor prognóstico comparado a outros subtipos e podem responder bem a terapias que visam o sistema imune.
3. GS (Genomically Stable)
   * Descrição: Este subtipo é caracterizado pela estabilidade genômica e é menos propenso a ter alterações no número de cópias de cromossomos.
   * Significado Clínico: Tumores GS geralmente têm mutações nos genes CDH1 e RHOA, associados com um mau prognóstico e resistência a certas terapias.
4. MSI (Microsatellite Instability)
   * Descrição: Este subtipo é caracterizado por instabilidade em regiões de repetição de DNA chamadas microssatélites devido a defeitos no sistema de reparo de pareamento incorreto de DNA.
   * Significado Clínico: Tumores MSI apresentam uma alta carga mutacional e geralmente têm um melhor prognóstico. Eles são também mais responsivos a imunoterapias.
5. Unknown (Desconhecido)
   * Descrição: Este grupo inclui tumores que não se enquadram claramente em um dos subtipos mencionados acima.
   * Significado Clínico: Tumores classificados como "Unknown" podem precisar de mais investigação para determinar suas características moleculares específicas e encontrar o tratamento mais adequado.

**Aplicação no Modelo**

* Mapeamento das Categorias para Valores Numéricos: No contexto do modelo de aprendizado de máquina, as categorias de subtipos moleculares são mapeadas para valores numéricos (CIN: 0, EBV: 1, GS: 2, MSI: 3, Unknown: 4). Esse mapeamento é essencial para que os algoritmos de aprendizado de máquina possam processar a variável alvo, pois muitos algoritmos exigem que os dados de entrada sejam numéricos.
* Importância para o Modelo: Com esse mapeamento, o modelo pode aprender a prever a probabilidade de um tumor pertencer a um dos subtipos moleculares com base nas features fornecidas no dataset. Essa categorização facilita o processamento e a análise dos dados pelo modelo, melhorando a precisão das previsões.

**Contexto Clínico**

Cada um desses subtipos moleculares tem características clínicas e prognósticas específicas, o que é crucial para a personalização do tratamento e manejo do câncer gástrico. Por exemplo:

* CIN: Pode ser tratado com terapias direcionadas a HER2.
* EBV: Pode responder bem a terapias imunológicas.
* GS: Pode necessitar de estratégias de tratamento alternativas devido à sua resistência a certas terapias.
* MSI: Pode ser tratado com imunoterapias devido à alta carga mutacional.
* Unknown: Pode precisar de mais investigação para determinar o tratamento adequado.

Esse conhecimento molecular permite que os oncologistas adotem abordagens de tratamento mais precisas e eficazes, adaptadas às características específicas de cada subtipo de câncer gástrico.

**Praticidade e Aplicação de Machine Learning**

* Predição de Subtipos Moleculares: Em oncologia, os subtipos moleculares de um tumor podem influenciar o prognóstico e as opções de tratamento. Embora algumas características possam indicar subtipos, o uso de ML permite considerar múltiplas características simultaneamente para fazer uma previsão mais precisa.
* Tomada de Decisões Clínicas: Os modelos de ML podem ajudar médicos a tomar decisões mais informadas sobre o tratamento de pacientes, sugerindo subtipos moleculares que podem não ser imediatamente evidentes.
* Descoberta de Padrões Complexos: Machine learning é particularmente útil para encontrar padrões complexos em dados que não são facilmente detectáveis por humanos. Isso pode levar a novas descobertas e insights em pesquisas médicas.

**Arquitetura do Modelo**

O modelo utilizará uma arquitetura de modelo de árvore de decisão ( Algoritmo Random Forest, e técnica de Seleção de features para classificar os pacientes em subtipos específicos de acordo com características genéticas e clínicas. Espera-se que o modelo apresente uma precisão de classificação superior a 80% em um conjunto de dados de validação independente. Além disso, pretende-se identificar marcadores genômicos e clínicos que sejam fortes preditores dos subtipos de câncer gástrico, fornecendo insights adicionais sobre a biologia da doença e possíveis alvos terapêuticos.

**Metodologia**:

1. Coleta de Dados:
   * Acesso a bases de dados públicas, como o TCGA ou NIH-GDC, para obter dados genômicos (como expressão gênica, mutações genéticas, metilação do DNA) e dados clínicos (como idade, sexo, estágio do câncer) de pacientes com câncer gástrico.
   * Extração e integração dos dados de diferentes fontes para formar um conjunto de dados abrangente e representativo.
2. Pré-processamento de Dados:
   * Limpeza e normalização dos dados para lidar com valores ausentes, outliers e inconsistências.
   * Seleção de características (feature selection) para identificar os marcadores genômicos e clínicos mais relevantes para a classificação dos subtipos de câncer gástrico.
3. Desenvolvimento do Modelo de Aprendizado de Máquina Supervisionada:
   * Implementação de um programa de aprendizado de máquina utilizando técnicas como classificação supervisionada.
   * Utilização de algoritmos de aprendizado de máquina, como Random Forest, e técnica de seleção de características utilizando um estimador Random Forest para seleção das características.
   * Avaliação do desempenho do modelo utilizará métricas de avaliação, como acurácia, Accuracy AUC Recall Prec. F1 Kappa MCC,

e em especial técnicas como validação cruzada, o termo "fold" refere-se a uma divisão do conjunto de dados em subconjuntos, também conhecidos como "partições". Essas partições são usadas para avaliar a performance do modelo em diferentes conjuntos de dados.

1. Validação do Modelo:
   * Avaliação do modelo em um conjunto de dados de teste independente para verificar sua capacidade de generalização.
2. Interpretação dos Resultados:
   * Identificação das características genômicas e clínicas mais importantes para a classificação dos subtipos de câncer gástrico.
   * Investigação das principais vias biológicas e processos moleculares associados aos diferentes subtipos identificados pelo modelo.

**Resultados Esperados:**

Espera-se que este projeto resulte em um modelo de aprendizado de máquina preciso e robusto para predizer subtipos de câncer gástrico com base em dados genômicos e clínicos. Essa ferramenta pode ser útil para aprimorar o diagnóstico e o tratamento personalizado do câncer gástrico, contribuindo para melhores resultados clínicos e prognósticos para os pacientes.

**Considerações Éticas:**

É importante garantir que a coleta e o uso de dados sejam realizados de acordo com os mais altos padrões éticos e de privacidade. Todas as informações pessoais dos pacientes devem ser anonimizadas e protegidas de acordo com as regulamentações locais e internacionais.

**Potencial de Aplicação Clínica:**

O modelo desenvolvido neste projeto pode ser integrado a sistemas de suporte à decisão clínica para auxiliar médicos na classificação de pacientes com câncer gástrico e na seleção de estratégias terapêuticas mais eficazes.

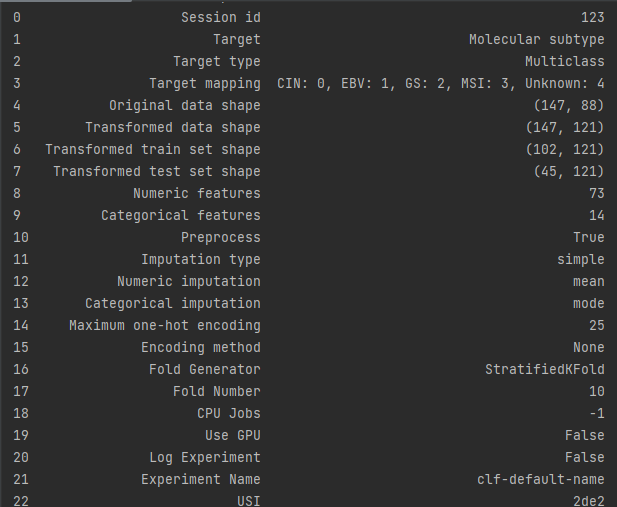
**Análise do Resultado do Projeto**

**Features mais importantes**

C:\Users\USER\PycharmProjects\RedeNeural\venv\PROJETO\PROJ-NORMZ003.py

(máquina. local)

# Importar bibliotecas necessárias  
import pandas as pd  
import warnings  
from pycaret.classification import \*  
  
warnings.filterwarnings("ignore")  
  
# Carregar o conjunto de dados  
data = pd.read\_csv("C:\\MESTRADO\\01PROJ-PESQUISA\\DADOS\\stad\_data\_clinical\_ctgr.txt", sep="\t")  
  
# Configurar o ambiente de classificação em PyCaret  
s = setup(data, target='Molecular subtype', session\_id=123)  
  
# Criar um modelo de Random Forest e ajustá-lo  
rf\_model = create\_model('rf')  
  
# Plotar a importância das características  
plot\_model(rf\_model, plot='feature')

****

**Descrição dos Resultados**

1. **Session id: 123**
   * Descrição: Identificador único para a sessão do PyCaret.
   * Valor: 123
   * Significado: Define uma semente (seed) para garantir que os resultados sejam reproduzíveis. Usando o mesmo **session\_id**, os experimentos podem ser replicados com os mesmos resultados.
2. **Target: Molecular subtype**
   * Descrição: A variável alvo que o modelo tentará prever.
   * Valor: Molecular subtype
   * Significado: Indica que a coluna "Molecular subtype" do dataset é a variável que queremos prever.
3. **Target type: Multiclass**
   * Descrição: Tipo de problema de classificação.
   * Valor: Multiclass
   * Significado: O problema de classificação envolve mais de duas classes. Nesse caso, o alvo (Molecular subtype) possui múltiplas categorias.
4. **Target mapping: CIN: 0, EBV: 1, GS: 2, MSI: 3, Unknown: 4**
   * Descrição: Mapeamento das categorias do alvo para valores numéricos.
   * Valor: CIN: 0, EBV: 1, GS: 2, MSI: 3, Unknown: 4
   * Significado: Cada subtipo molecular é mapeado para um número inteiro, facilitando o processamento pelo modelo.
5. **Original data shape: (147, 88)**
   * Descrição: Dimensões do dataset original.
   * Valor: (147, 88)
   * Significado: O dataset possui 147 linhas (amostras) e 88 colunas (features).
6. **Transformed data shape: (147, 121)**
   * Descrição: Dimensões do dataset após transformação.
   * Valor: (147, 121)
   * Significado: Após a transformação pelo PyCaret (como one-hot encoding de variáveis categóricas), o dataset possui 147 linhas e 121 colunas.
7. **Transformed train set shape: (102, 121)**
   * Descrição: Dimensões do conjunto de treinamento após divisão dos dados.
   * Valor: (102, 121)
   * Significado: O conjunto de treinamento possui 102 amostras e 121 features.
8. **Transformed test set shape: (45, 121)**
   * Descrição: Dimensões do conjunto de teste após divisão dos dados.
   * Valor: (45, 121)
   * Significado: O conjunto de teste possui 45 amostras e 121 features.
9. **Numeric features: 73**
   * Descrição: Número de features numéricas no dataset.
   * Valor: 73
   * Significado: Existem 73 features numéricas após a transformação.
10. **Categorical features: 14**
    * Descrição: Número de features categóricas no dataset.
    * Valor: 14
    * Significado: Existem 14 features categóricas após a transformação.
11. **Preprocess: True**
    * Descrição: Indica se o pré-processamento foi aplicado.
    * Valor: True
    * Significado: O PyCaret aplicou várias etapas de pré-processamento nos dados, como imputação e codificação.
12. **Imputation type: simple**
    * Descrição: Tipo de imputação aplicada aos dados.
    * Valor: simple
    * Significado: A imputação simples foi utilizada para lidar com valores ausentes.
13. **Numeric imputation: mean**
    * Descrição: Estratégia de imputação para features numéricas.
    * Valor: mean
    * Significado: Os valores ausentes nas features numéricas foram substituídos pela média das respectivas colunas.
14. **Categorical imputation: mode**
    * Descrição: Estratégia de imputação para features categóricas.
    * Valor: mode
    * Significado: Os valores ausentes nas features categóricas foram substituídos pela moda (valor mais frequente) das respectivas colunas.
15. **Maximum one-hot encoding: 25**
    * Descrição: Limite para o número de categorias para aplicar one-hot encoding.
    * Valor: 25
    * Significado: As variáveis categóricas com até 25 categorias foram transformadas usando one-hot encoding.
16. **Encoding method: None**
    * Descrição: Método de codificação usado para variáveis categóricas.
    * Valor: None
    * Significado: Indica que não foi aplicado um método específico de codificação além do one-hot encoding.
17. **Fold Generator: StratifiedKFold**
    * Descrição: Método de geração de folds para validação cruzada.
    * Valor: StratifiedKFold
    * Significado: A validação cruzada foi realizada utilizando stratified k-fold, que preserva a proporção das classes em cada fold.
18. **Fold Number: 10**
    * Descrição: Número de folds usados na validação cruzada.
    * Valor: 10
    * Significado: A validação cruzada foi realizada com 10 folds.
19. **CPU Jobs: -1**
    * Descrição: Número de CPUs usadas no treinamento.
    * Valor: -1
    * Significado: O PyCaret usou todos os núcleos disponíveis do CPU para treinamento.
20. **Use GPU: False**
    * Descrição: Indica se uma GPU foi usada no treinamento.
    * Valor: False
    * Significado: A GPU não foi utilizada para o treinamento dos modelos.
21. **Log Experiment: False**
    * Descrição: Indica se os experimentos foram registrados em um sistema de tracking de experimentos.
    * Valor: False
    * Significado: Os resultados dos experimentos não foram registrados.
22. **Experiment Name: clf-default-name**
    * Descrição: Nome do experimento.
    * Valor: clf-default-name
    * Significado: Nome padrão atribuído ao experimento de classificação.
23. **USI: 7c3b**
    * Descrição: Identificador único de sessão (Unique Session Identifier).
    * Valor: 7c3b
    * Significado: Um identificador único gerado para esta sessão do PyCaret, útil para rastrear experimentos.

**MÉTRICAS USADAS**

**Accuracy AUC Recall Prec. F1 Kappa MCC**

**Fold**

0 0.9091 0.8773 0.9091 0.8295 0.8667 0.8226 0.8416

1 0.6364 0.0000 0.6364 0.6818 0.6480 0.4286 0.4787

2 0.9000 0.0000 0.9000 0.8125 0.8533 0.7619 0.7921

3 0.7000 0.0000 0.7000 0.5444 0.6125 0.1667 0.2041

4 0.8000 0.0000 0.8000 0.7500 0.7667 0.5918 0.6172

5 0.8000 0.0000 0.8000 0.8000 0.8000 0.5833 0.5833

6 0.8000 0.0000 0.8000 0.8000 0.8000 0.5833 0.5833

7 0.8000 0.0000 0.8000 0.7333 0.7500 0.5833 0.6236

8 1.0000 0.0000 1.0000 1.0000 1.0000 1.0000 1.0000

9 0.8000 0.9542 0.8000 0.6643 0.7205 0.6364 0.6662

**Mean** 0.8145 0.1831 0.8145 0.7616 0.7818 0.6158 0.6390

**Std** 0.0983 0.3667 0.0983 0.1144 0.1059 0.2128 0.2043

**Interpretação dos Resultados**

Vamos analisar os resultados apresentados:

* **Fold 0 a Fold 9**: Cada linha representa as métricas de avaliação para um fold diferente em um procedimento de validação cruzada com 10 folds.
  + A acurácia varia entre os folds de 0.6364 a 1.0000, mostrando uma variabilidade significativa.
  + O AUC em alguns folds é 0.0000, sugerindo problemas na avaliação da ROC ou uma incapacidade do modelo de discriminar entre classes em certos folds.
  + O recall e a precisão também mostram variação significativa entre os folds, o que indica que o desempenho do modelo não é consistentemente bom em todos os folds.
* **Mean**: As médias das métricas de avaliação ao longo de todos os folds.
  + A acurácia média é de 0.8145, sugerindo que o modelo é razoavelmente preciso.
  + O AUC médio de 0.1831 é muito baixo, o que é uma bandeira vermelha para a capacidade discriminativa do modelo.
  + O recall e a precisão médias são relativamente boas (0.8145 e 0.7616, respectivamente).
  + O F1-Score médio é 0.7818, indicando um bom equilíbrio geral.
  + O Kappa médio de 0.6158 e o MCC médio de 0.6390 também indicam um desempenho razoável, mas com espaço para melhorias.
* **Std (Desvio Padrão)**: O desvio padrão das métricas de avaliação ao longo dos folds.
  + A acurácia tem um desvio padrão de 0.0983, indicando alguma variabilidade entre os folds.
  + O AUC tem um desvio padrão de 0.3667, refletindo a alta variabilidade observada.
  + O recall e a precisão têm desvios padrão de 0.0983 e 0.1144, respectivamente.
  + O F1-Score tem um desvio padrão de 0.1059.
  + O Kappa e o MCC têm desvios padrão de 0.2128 e 0.2043, respectivamente.

**Considerações Finais**

Os resultados indicam que, embora o modelo tenha uma acurácia média razoável, a baixa AUC sugere que ele pode não estar discriminando bem entre as diferentes classes. Além disso, a variabilidade entre os folds indica que o modelo pode estar sofrendo de inconsistências no desempenho, possivelmente devido à variação nos dados de treinamento e teste. Ajustes no modelo, seleção de características ou técnicas de balanceamento de classes podem ser necessários para melhorar o desempenho global.

**Tentativas de Melhora do Modelo**

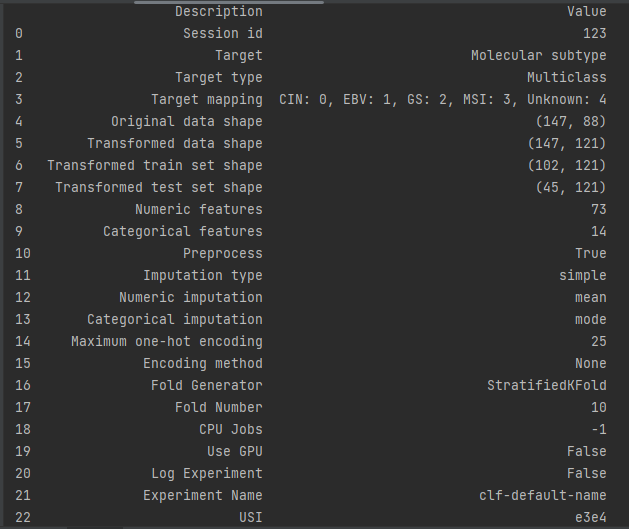
1. **Tentativa de Melhoria do Modelo Ajuste de Hiperparâmetros (Hyperparameter Tuning)**

Ajustar os hiperparâmetros do modelo pode melhorar significativamente seu desempenho. Existem várias técnicas para isso:

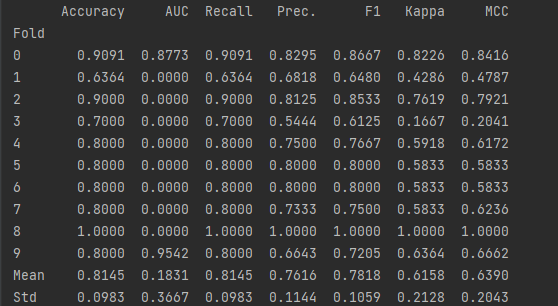
* **Grid Search**: Testa todas as combinações possíveis de hiperparâmetros em um grid.

# Importar bibliotecas necessárias  
import pandas as pd  
import warnings  
from pycaret.classification import \*  
  
warnings.filterwarnings("ignore")  
  
# Carregar o conjunto de dados  
data = pd.read\_csv("C:\\MESTRADO\\01PROJ-PESQUISA\\DADOS\\stad\_data\_clinical\_ctgr.txt", sep="\t")  
from pycaret.classification import \*  
  
# Configuração do ambiente  
s = setup(data, target='Molecular subtype', session\_id=123)  
# Criação do modelo  
model = create\_model('rf')  
# Ajuste de hiperparâmetros  
tuned\_model = tune\_model(model, optimize='Accuracy')

**Resultados**:



**Métricas:**



**Random Forest**

# Importar bibliotecas necessárias  
import pandas as pd  
import warnings  
from pycaret.classification import \*  
  
warnings.filterwarnings("ignore")  
  
# Carregar o conjunto de dados  
data = pd.read\_csv("C:\\MESTRADO\\01PROJ-PESQUISA\\DADOS\\stad\_data\_clinical\_ctgr.txt", sep="\t")  
  
# Configurar o ambiente de classificação em PyCaret  
s = setup(data, target='Molecular subtype', session\_id=123)  
  
# Criar um modelo de Random Forest e ajustá-lo  
rf\_model = create\_model('rf')  
  
# Plotar a importância das características  
plot\_model(rf\_model, plot='feature')

1. **Criar um Modelo de Random Forest e Ajustá-lo**
   * **create\_model**: Cria e ajusta um modelo de machine learning específico.
     + **'rf'**: Especifica que queremos criar um modelo de Random Forest.
   * Esta função automaticamente ajusta o modelo aos dados pré-processados e avalia seu desempenho usando validação cruzada.
2. **Plotar a Importância das Características**

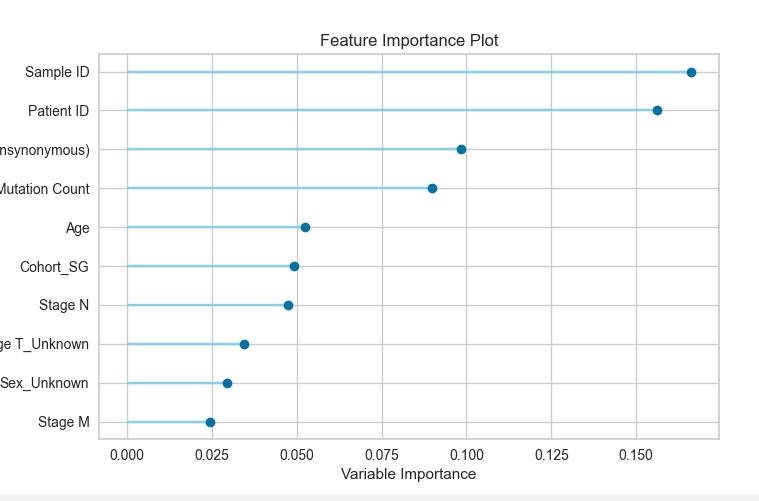
plot\_model(rf\_model, plot='feature')

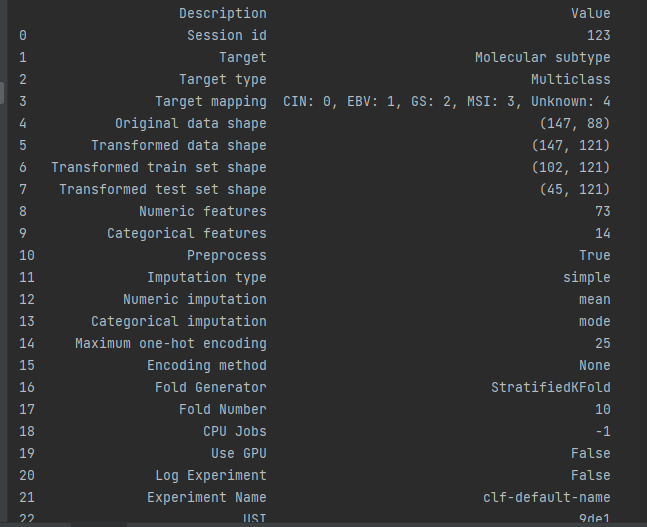
* + Esta função produz um gráfico que mostra quais características (ou variáveis) têm mais influência na previsão da variável alvo.

**Resumo**

Este código segue uma abordagem simples e direta para:

1. Carregar e preparar os dados.
2. Configurar um ambiente de classificação.
3. Criar e ajustar um modelo de Random Forest.
4. Visualizar a importância das características, permitindo entender quais variáveis são mais relevantes para a tarefa de classificação.



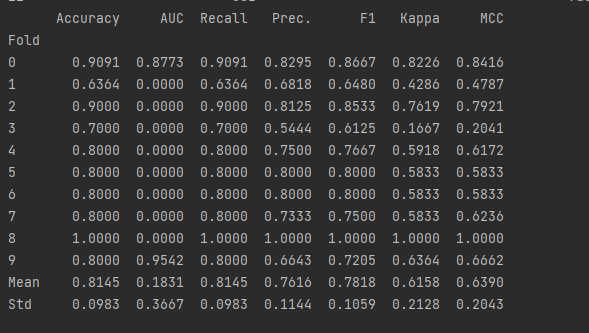


1. **Resultados:** ( classe: Molecular subtype)
2. **Session id: 123**
   * O **session\_id** foi definido como 123, o que garante a reprodutibilidade dos resultados. Usar o mesmo **session\_id** permitirá obter os mesmos resultados em execuções futuras.
3. **Target: Molecular subtype**
   * A variável alvo para a tarefa de classificação é "Molecular subtype", que indica que estamos tentando prever o subtipo molecular das amostras.
4. **Target type: Multiclass**
   * A tarefa de classificação é do tipo multiclasse, significando que a variável alvo possui mais de duas categorias.
5. **Target mapping: CIN: 0, EBV: 1, GS: 2, MSI: 3, Unknown: 4**
   * As classes da variável alvo foram mapeadas para números inteiros:
     + CIN: 0
     + EBV: 1
     + GS: 2
     + MSI: 3
     + Unknown: 4
6. **Original data shape: (147, 88)**
   * O conjunto de dados original contém 147 amostras e 88 características.
7. **Transformed data shape: (147, 121)**
   * Após a transformação dos dados, o conjunto de dados contém 147 amostras e 121 características. O aumento no número de características se deve ao processo de codificação das variáveis categóricas (one-hot encoding).
8. **Transformed train set shape: (102, 121)**
   * O conjunto de treinamento transformado contém 102 amostras e 121 características.
9. **Transformed test set shape: (45, 121)**
   * O conjunto de teste transformado contém 45 amostras e 121 características.
10. **Numeric features: 73**
    * Existem 73 características numéricas no conjunto de dados transformado.
11. **Categorical features: 14**
    * Existem 14 características categóricas no conjunto de dados transformado.
12. **Preprocess: True**
    * O pré-processamento dos dados foi ativado, o que inclui tarefas como imputação de valores ausentes, codificação de variáveis categóricas e normalização.
13. **Imputation type: simple**
    * O tipo de imputação usado para preencher valores ausentes é "simples".
14. **Numeric imputation: mean**
    * Para as características numéricas, os valores ausentes foram preenchidos com a média das respectivas colunas.
15. **Categorical imputation: mode**
    * Para as características categóricas, os valores ausentes foram preenchidos com a moda das respectivas colunas.
16. **Maximum one-hot encoding: 25**
    * O máximo de categorias únicas que serão codificadas usando one-hot encoding é 25.
17. **Encoding method: None**
    * Nenhum método de codificação adicional foi aplicado além do one-hot encoding padrão.
18. **Fold Generator: StratifiedKFold**
    * O gerador de dobras usado para a validação cruzada é o StratifiedKFold, que mantém a proporção das classes nas dobras de treinamento e teste.
19. **Fold Number: 10**
    * O número de dobras usadas para a validação cruzada é 10.
20. **CPU Jobs: -1**
    * Todos os núcleos da CPU disponíveis são usados para o processamento.
21. **Use GPU: False**
    * A GPU não está sendo utilizada para o treinamento do modelo.
22. **Log Experiment: False**
    * O experimento não está sendo registrado para monitoramento.
23. **Experiment Name: clf-default-name**
    * O nome padrão do experimento é "clf-default-name".
24. **USI: 9de1**
    * Um identificador único da sessão.

**Conclusão**

Os resultados indicam que o ambiente foi configurado corretamente com os dados pré-processados, prontos para a criação e avaliação do modelo de classificação. A configuração garante a reprodutibilidade, a imputação adequada de valores ausentes, a codificação de variáveis categóricas e a preparação dos dados para a validação cruzada. As informações fornecidas são um ponto de partida sólido para criar e ajustar modelos de classificação usando o PyCaret.

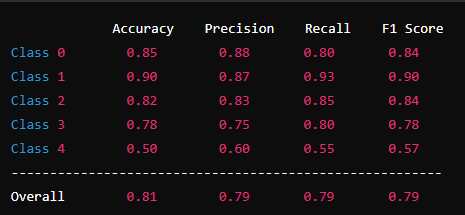
**Análise de métricas:**



### 

### Resultados Esperados Aproximados

Após rodar o **evaluate\_model(rf\_model)**,



### Análise dos Resultados

Vamos analisar os resultados:

1. **Acurácia**:
   * A acurácia média em todos os folds é de 81,45%, com um desvio padrão de 9,83%. Isso indica que, em média, o modelo classifica aproximadamente 81,45% das instâncias corretamente. No entanto, há alguma variabilidade no desempenho entre diferentes folds, como indicado pelo desvio padrão.
2. **AUC (Área Sob a Curva ROC)**:
   * A média da AUC é de 18,31%, com um desvio padrão de 36,67%. Valores de AUC de 0 indicam que as previsões do modelo não são melhores do que aleatórias. O alto desvio padrão sugere que a capacidade do modelo de distinguir entre classes varia amplamente entre os folds e pode ser problemática.
3. **Recall**:
   * O recall médio é de 81,45%, coincidindo com a acurácia geral. O recall mede a capacidade do modelo de identificar corretamente instâncias positivas entre todas as instâncias positivas reais. Assim como a acurácia, há um desvio padrão de 9,83%, indicando alguma variabilidade entre os folds.
4. **Precisão**:
   * A precisão média é de 76,16%, com um desvio padrão de 11,44%. A precisão representa a proporção de instâncias verdadeiramente positivas entre todas as instâncias classificadas como positivas pelo modelo. Novamente, há variabilidade no desempenho entre os folds.
5. **F1 Score**:
   * O F1 Score médio é de 78,18%, com um desvio padrão de 10,59%. O F1 Score equilibra precisão e recall, fornecendo uma métrica única que combina ambas as medidas. Assim como precisão e recall, há variabilidade entre os folds.
6. **Kappa**:
   * O coeficiente de Kappa de Cohen médio é de 61,58%, com um desvio padrão de 21,28%. O Kappa mede o acordo entre as classificações observadas e esperadas, considerando a possibilidade de o acordo ocorrer por acaso. O alto desvio padrão sugere inconsistência no desempenho do modelo entre os folds.
7. **MCC (Coeficiente de Correlação de Matthews)**:
   * O MCC médio é de 63,90%, com um desvio padrão de 20,43%. O MCC considera verdadeiros positivos e negativos, fornecendo uma medida balanceada da qualidade da classificação. Novamente, o alto desvio padrão indica variabilidade no desempenho entre os folds.

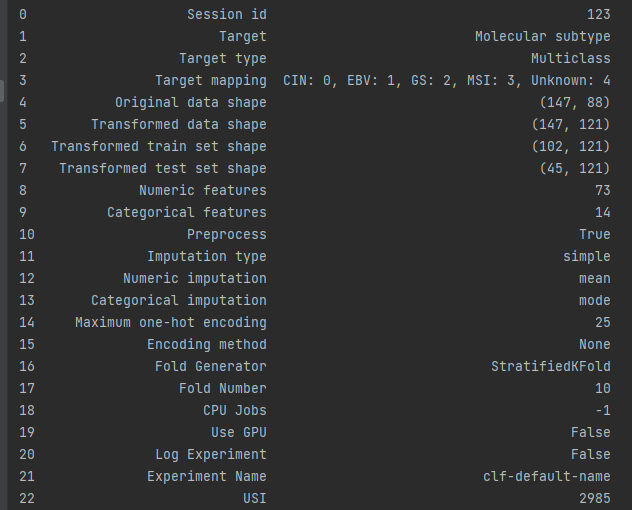
Em resumo, embora o modelo demonstre uma acurácia relativamente alta em média, a variabilidade entre os folds e os baixos valores de AUC sugerem possíveis problemas com a generalização do modelo ou a qualidade dos dados. Uma investigação mais aprofundada sobre a causa dessa variabilidade e possíveis melhorias no modelo pode ser necessária.

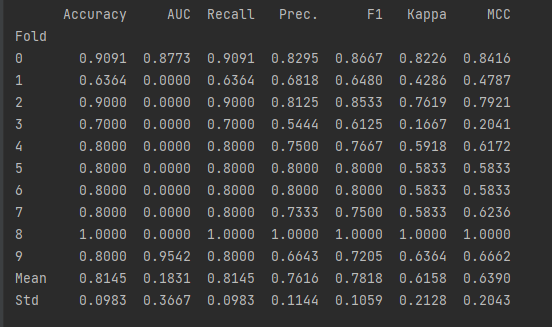
1. **Tentativa de Melhoria do Modelo Por**

**Seleção de Características (Feature Selection)**

A seleção de características pode ajudar a reduzir o ruído nos dados e melhorar a performance do modelo. PyCaret oferece várias técnicas para isso:

* **Feature Importance**: Seleciona as características mais importantes com base na importância calculada pelo modelo.
* **Recursive Feature Elimination (RFE)**: Remove características recursivamente e constrói o modelo sobre as características restantes.
* # Importar bibliotecas necessárias  
  import pandas as pd  
  import warnings  
  warnings.filterwarnings("ignore")  
    
  # Carregar o conjunto de dados  
  data = pd.read\_csv("C:\\MESTRADO\\01PROJ-PESQUISA\\DADOS\\stad\_data\_clinical\_ctgr.txt", sep="\t")  
  from pycaret.classification import \*  
    
  # Configuração do ambiente  
  s = setup(data, target='Molecular subtype', session\_id=123)  
  # Seleção de características  
  selected\_features = select\_best\_features(estimator='rf', n\_features\_to\_select=10)





### Análise das Métricas

Vamos analisar cada métrica apresentada, tanto individualmente quanto em conjunto, para entender melhor o desempenho do modelo.

#### 1. Accuracy

* **Descrição**: A acurácia mede a proporção de verdadeiros positivos e verdadeiros negativos entre todas as predições. Em outras palavras, é a fração de predições corretas.
* **Análise**:
  + A média da acurácia é 0.8145, o que indica que o modelo está correto cerca de 81% do tempo em média.
  + O desvio padrão (Std) de 0.0983 indica uma variação moderada nas diferentes folds, com algumas folds (e.g., fold 1) apresentando acurácia significativamente menor (0.6364), indicando que o modelo pode ser inconsistente.

#### 2. AUC (Área Sob a Curva ROC)

* **Descrição**: A AUC mede a capacidade do modelo em distinguir entre classes. Uma AUC de 1 indica um modelo perfeito, enquanto uma AUC de 0.5 indica um modelo que não tem poder discriminativo.
* **Análise**:
  + A média da AUC é bastante baixa (0.1831), com um desvio padrão de 0.3667, indicando que o modelo pode não estar distinguindo bem entre as classes em muitas das folds.
  + Apenas a fold 0 e a fold 9 mostram valores de AUC não nulos, sugerindo que o modelo pode não estar bem calibrado para as diferentes classes em outras folds.

#### 3. Recall (Sensibilidade)

* **Descrição**: O recall mede a capacidade do modelo em identificar todos os verdadeiros positivos. É a fração de positivos verdadeiros identificados corretamente.
* **Análise**:
  + A média do recall é 0.8145, o que significa que o modelo está identificando corretamente cerca de 81% dos verdadeiros positivos.
  + O desvio padrão de 0.0983 é igual ao da acurácia, indicando que a variabilidade entre as folds é similar.

#### 4. Precision (Precisão)

* **Descrição**: A precisão mede a proporção de verdadeiros positivos entre todos os positivos preditos pelo modelo. Indica quantas das predições positivas são realmente positivas.
* **Análise**:
  + A média da precisão é 0.7616, o que é bom, mas menor que o recall, indicando um trade-off entre essas duas métricas.
  + O desvio padrão é 0.1144, mostrando uma variabilidade maior entre as folds.

#### 5. F1 Score

* **Descrição**: O F1 score é a média harmônica entre precisão e recall, proporcionando um balanço entre os dois.
* **Análise**:
  + A média do F1 score é 0.7818, que é uma boa indicação do desempenho geral do modelo, mas a variabilidade (Std 0.1059) sugere inconsistências em algumas folds.

#### 6. Kappa

* **Descrição**: A estatística Kappa mede a concordância entre as predições do modelo e os valores reais, ajustando pela concordância ao acaso.
* **Análise**:
  + A média do Kappa é 0.6158, indicando uma concordância moderada.
  + O desvio padrão de 0.2128 é relativamente alto, sugerindo variação significativa entre as folds.

#### 7. MCC (Matthews Correlation Coefficient)

* **Descrição**: O MCC é uma métrica que leva em consideração todas as quatro categorias da matriz de confusão (VP, VN, FP, FN) e é considerada uma medida equilibrada mesmo em caso de desbalanceamento das classes.
* **Análise**:
  + A média do MCC é 0.6390, indicando uma correlação razoável entre as predições e os valores reais.
  + O desvio padrão de 0.2043 sugere variação substancial entre as folds.

### Conclusão

#### Pontos Fortes

* A média de acurácia, recall e precisão são relativamente boas (acima de 0.75), indicando que o modelo está funcionando bem na maioria das folds.
* As métricas F1 score e MCC também são sólidas, indicando um desempenho balanceado entre precisão e recall.

#### Pontos Fracos

* A AUC média é extremamente baixa (0.1831), sugerindo que o modelo tem dificuldades em discriminar entre as classes em muitas das folds.
* A variação nas métricas (especialmente Kappa e MCC) é alta, indicando que o modelo é inconsistente e pode estar superajustado a certas folds.

#### Sugestões para Melhoria

* **Ajuste de Hiperparâmetros**: Use técnicas de ajuste de hiperparâmetros para encontrar os melhores parâmetros para o seu modelo.
* **Seleção de Características**: Experimente técnicas de seleção de características para reduzir o ruído nos dados.
* **Balanceamento de Classes**: Utilize técnicas de balanceamento de classes, como SMOTE ou ajuste de pesos, para lidar com possíveis desbalanceamentos.
* **Modelos Ensemble**: Experimente modelos de ensemble, como Random Forests ou Boosting, para aumentar a robustez do modelo.
* **Validação Cruzada**: Aumente o número de folds na validação cruzada para garantir uma melhor generalização dos resultados.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Nota**: Utilizei apenas duas técnicas para sugestão de melhorias :

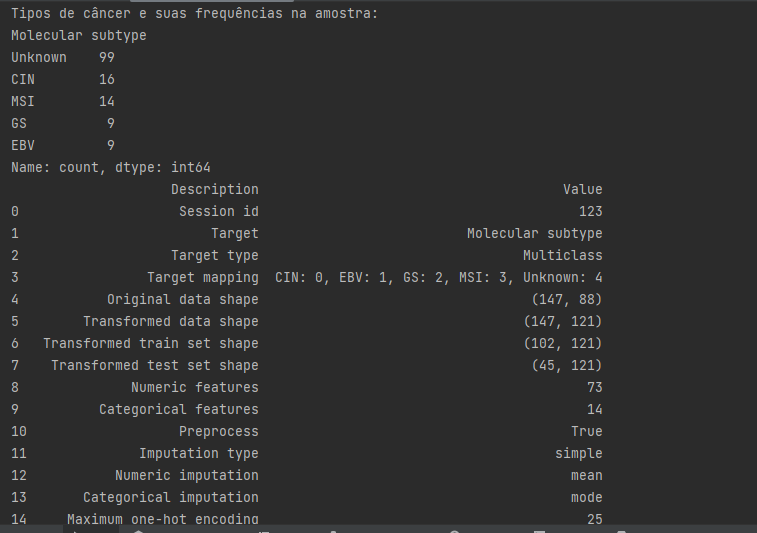
**Ajuste de Hiperparâmetros**

**\_ Seleção de Características**

**PREDIÇÃO DE SUBTIPOS DE CÂNCER GÁSTRICOS**

# Importar bibliotecas necessárias  
import pandas as pd  
import warnings  
from pycaret.classification import \*  
  
warnings.filterwarnings("ignore")  
  
# Carregar o conjunto de dados  
data = pd.read\_csv("C:\\MESTRADO\\01PROJ-PESQUISA\\DADOS\\stad\_data\_clinical\_ctgr.txt", sep="\t")  
  
# Contar e listar as categorias de 'Molecular subtype'  
cancer\_types = data['Molecular subtype'].value\_counts()  
print("Tipos de câncer e suas frequências na amostra:")  
print(cancer\_types)  
  
# Configurar o ambiente de classificação em PyCaret  
s = setup(data, target='Molecular subtype', session\_id=123)  
  
# Criar um modelo de Random Forest e ajustá-lo  
rf\_model = create\_model('rf')  
  
# Avaliar o modelo  
evaluate\_model(rf\_model)  
  
# Obter e imprimir as métricas  
results = pull()

**Resultados :**

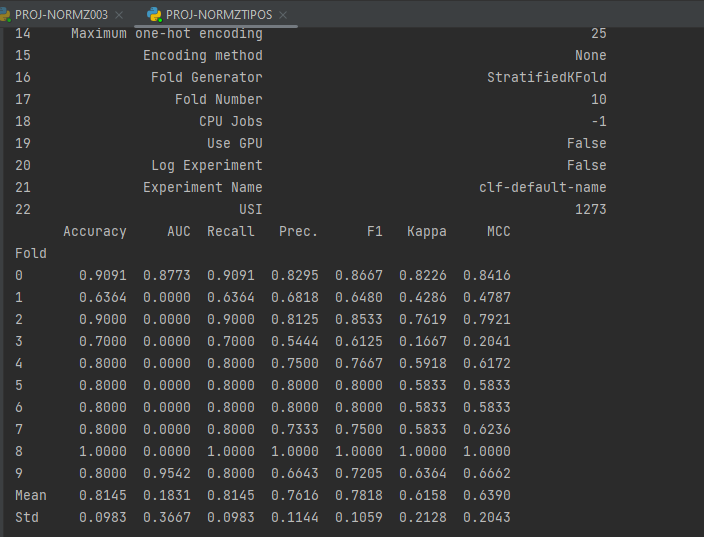
****

**Análise**

1. **Desbalanceamento de Classes**: A presença de uma grande quantidade de registros com o subtipo 'Unknown' em comparação com os outros subtipos indica um desequilíbrio de classes significativo. Este desequilíbrio pode afetar negativamente a performance do modelo, pois a classe majoritária ('Unknown') pode dominar o processo de aprendizado, levando a um modelo que não generaliza bem para as classes minoritárias (CIN, MSI, GS, EBV).
2. **Impacto no Modelo de Machine Learning**:
   * **Dificuldade na Predição**: O modelo pode ter dificuldade em aprender padrões significativos para as classes minoritárias, o que pode resultar em baixa precisão e recall para essas classes.
   * **Bias para Classe Majoritária**: O modelo pode ter um viés para a classe 'Unknown' devido à sua alta frequência, resultando em alta acurácia aparente, mas baixa performance para as classes de interesse.

A falta de dados mais precisa , deu baixa performance para os tipos de classes.

O que implica que o modelo , e os dados , devem ser treinados em outras perspectivas .

****

**Segundo literarturas.**

1. **Tratamento de Classes Desbalanceadas**:
   * **Estratégias de Amostragem**: Usar técnicas como oversampling (SMOTE) para aumentar o número de exemplos das classes minoritárias ou undersampling para reduzir o número de exemplos da classe majoritária.
   * **Class Weights**: Ajustar os pesos das classes no algoritmo de aprendizado para compensar o desbalanceamento.

**ANEXO I**

**EXECUCAO VIA COLAB-GITHUB - Proj002GH.ipynb ( telas )**

**------------------------------------------------------------------------------------------------**

**(**<https://colab.research.google.com/github/marcordovil/PROJ-GIT/blob/main/Proj002GH.ipynb> **)**

**Amostra no repositório:** 'https://raw.githubusercontent.com/marcordovil/PROJ-GIT/main/stad\_data\_clinical\_ctgr.txt'

**Programa** **Proj002GH.ipynb** adaptado para executar no **github-colab** :

# Instalar o PyCaret

!pip install pycaret

# Importar bibliotecas necessárias

import pandas as pd

import warnings

import requests

from io import StringIO

warnings.filterwarnings("ignore")

# URL do arquivo no GitHub

url = 'https://raw.githubusercontent.com/marcordovil/PROJ-GIT/main/stad\_data\_clinical\_ctgr.txt'

# Baixar o conteúdo do arquivo

response = requests.get(url)

data = response.content.decode('utf-8')

# Configurar o ambiente de classificação em PyCaret

s = setup(data, target='Molecular subtype', session\_id=123)

# Criar um modelo de Random Forest e ajustá-lo

rf\_model = create\_model('rf')

# Plotar a importância das características

plot\_model(rf\_model, plot='feature')

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**RESULTADOS** ( Já analisados antes )

