P.PORTO

ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO

Análise e previsão de ataque cardíaco

Inteligência Artificial

João António Pinto Vieira 8200620

Índice

Índice de Imagens

1. Introdução

Um ataque cardíaco, também chamado de infarto do miocárdio, é uma emergência médica grave que ocorre quando o fluxo de sangue para uma parte do músculo cardíaco é bloqueado, geralmente por um acúmulo de placas nas artérias. Este bloqueio corta o fornecimento de oxigênio ao músculo cardíaco, o que pode danificar ou destruir o tecido.

O culpado mais comum é a doença arterial coronariana (CAD), onde placas (feitas de colesterol e outras substâncias) se acumulam dentro das artérias coronárias que fornecem sangue ao coração. Isso estreita as artérias, reduzindo o fluxo sanguíneo.

1.1. Contextualização

No âmbito da unidade curricular de Inteligência artificial, desenvolvi um sistema de inteligência artificial para estimar a chance de alguem ter um ataque cardíaco baseado em diversos dados introduzidos como input. O foco principal deste trabalho reside na área de Machine Learning, utilizando técnicas para analisar dados e identificar padrões que indiquem um risco elevado que seja propício a desencadear um ataque cardíaco.

Este trabalho visa prever a probabilidade de ocorrer um ataque cardíaco num paciente baseado em diversas características que ele mesmo apresentra. Essas características são usadas para classificar um indivíduo numa das duas classes possíveis (sofrer ou não sofrer um ataque cardíaco). Logo, podemos afirmar que estamos perante um problema de classificação.

1.2. Apresentação do Caso de Estudo

Os seguintes dados descrevem o conjunto de dados (dataset) utilizado para o desenvolvimento dos modelos para estimar a probabilidade de alguem ter um ataque cardíaco. Contém informações sobre pacientes e a presença ou ausência de um algum problema cardíaco.

- **Idade (Age):** Representa a idade do paciente.
- **Sexo (Sex):** Indica o sexo do paciente.
- **Angina induzida por esforço (exang):** Informa se o paciente apresenta angina induzida por esforço (1 = sim; 0 = não).

- **Número de vasos principais (ca):** Indica o número de vasos coronários principais afetados (0-3).
- **Tipo de dor torácica (cp):** Classifica o tipo de dor torácica relatada pelo paciente:
    i. Valor 1: Angina típica
    ii. Valor 2: Angina atípica
    iii. Valor 3: Dor não anginosa
    iv. Valor 4: Assintomático

- **Pressão arterial sistólica em repouso (trtbps):** Representa a pressão arterial sistólica do paciente em repouso (em mm Hg).
- **Colesterol (chol):** Indica o nível de colesterol do paciente em mg/dl, obtido através do sensor de IMC.
- **Glicemia em jejum (fbs):** Informa se a glicemia em jejum do paciente é superior a 120 mg/dl (1 = verdadeiro; 0 = falso).
- **Resultado do eletrocardiograma de repouso (_restecg):** Classifica o resultado do eletrocardiograma de repouso:

    i. Valor 0: Normal
    ii. Valor 1: Apresenta anormalidade da onda ST-T (inversões da onda T e/ou elevação ou depressão do segmento ST > 0,05 mV)
    iii. Valor 2: Mostra provável ou definitiva hipertrofia ventricular esquerda pelos critérios de Estes

- **Frequência cardíaca máxima (thalach):** Representa a frequência cardíaca máxima alcançada pelo paciente durante o exame.
- **Alvo (target):** Indica a probabilidade de ataque cardíaco:

    i. 0 = Menor chance de ataque cardíaco
    ii. 1 = Maior chance de ataque cardíaco

2. Tratamento e análise de dados

2.1. Carregar o dataset

```
data = pd.read_csv("heart.csv")
data.head()
```

Out[627]:

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	thall	output
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Figura 1 - Carregar o dataset

2.2. Análise inicial do dataset

In [18]:
```
print("Shape of Dataset:", data.shape, '\n')
data.info()
```

```
Shape of Dataset: (302, 14)

<class 'pandas.core.frame.DataFrame'>
Int64Index: 302 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       302 non-null    int64
 1   sex       302 non-null    int64
 2   cp        302 non-null    int64
 3   trtbps    302 non-null    int64
 4   chol      302 non-null    int64
 5   fbs       302 non-null    int64
 6   restecg   302 non-null    int64
 7   thalachh  302 non-null    int64
 8   exng      302 non-null    int64
 9   oldpeak   302 non-null    float64
 10  slp       302 non-null    int64
 11  caa       302 non-null    int64
 12  thall     302 non-null    int64
 13  output    302 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 45.4 KB
```

Figura 2 - Análise inicial do dataset

2.2. Análise dos outputs

- O conjunto de dados consiste em 303 linhas e 14 colunas.
- Todas as variáveis estão em formato numérico (Int ou Float).
- À primeira vista, não há campos vazios no conjunto de dados.

3. Análise exploratória dos dados

3.1. Verificar a ausência de campos

Após a verificação dos campos do dataset, podemos confirmar que realmente todos os campos estão preenchidos e nenhum está a null.

```
In [20]: data.isnull().sum()

Out[20]: age         0
         sex         0
         cp          0
         trtbps      0
         chol        0
         fbs         0
         restecg     0
         thalachh    0
         exng        0
         oldpeak     0
         slp         0
         caa         0
         thall       0
         output      0
         dtype: int64
```

Figura 3 - Verificação do dataset

3.2. Verificar existência de dados duplicados

Dados duplicados representam informações repetidas que não agregam valor ao modelo. Portanto neste passo removi esses dados duplicados para garantir que o modelo seja treinado com um conjunto de dados mais conciso e preciso, de maneira a evitar distorções e aumentar a eficiência.

Podemos confirmer que neste dataset existia uma linha de dados repetida, a qual eu removi.

```
In [4]: data.duplicated().sum()
Out[4]: 1

In [5]: data.drop_duplicates(inplace=True)
        data.shape
Out[5]: (302, 14)
```

Figura 4 - Verificação de dados duplicados

```
In [27]: unique_number = []
         for i in data.columns:
             x = data[i].value_counts().count()
             unique_number.append(x)
         pd.DataFrame(unique_number, index=data.columns, columns=["Total de Unique Values"])
```

Out[27]:

	Total de Unique Values
age	41
sex	2
cp	4
trtbps	49
chol	152
fbs	2
restecg	3
thalachh	91
exng	2
oldpeak	40
slp	3
caa	5
thall	4
output	2

Figura 5 - Verificar dados unicos

3.3. Analisar valores únicos

Analisando a quantidade de valores únicos para cada variável no dataset conseguimos classificar as variáveis em duas categorias:

- **Variáveis Numéricas:** Possuem uma quantidade alta de valores únicos. Nesse grupo, encontramos "age", "trtbps", "chol", "thalachh" e "oldpeak".
- **Variáveis Categóricas:** Possuem um número baixo de valores únicos. Esse grupo inclui "sex", "cp", "fbs", "restecg", "exng", "slp", "caa", “thall” e "target" (risco de ataque cardíaco).

Na próxima etapa, separarei essas variáveis em duas listas distintas.

3.4. Separação e análise das variáveis (Numéricas e Categóricos)

```
In [31]: numeric_var = ["age", "trtbps", "chol", "thalachh", "oldpeak"]
         categorical_var = ["sex", "cp", "fbs", "rest_ecg", "exang", "slope", "ca", "thal", "target"]

         data[numeric_var].describe()
```

Out[31]:

	age	trtbps	chol	thalachh	oldpeak
count	302.00000	302.000000	302.000000	302.000000	302.000000
mean	54.42053	131.602649	246.500000	149.569536	1.043046
std	9.04797	17.563394	51.753489	22.903527	1.161452
min	29.00000	94.000000	126.000000	71.000000	0.000000
25%	48.00000	120.000000	211.000000	133.250000	0.000000
50%	55.50000	130.000000	240.500000	152.500000	0.800000
75%	61.00000	140.000000	274.750000	166.000000	1.600000
max	77.00000	200.000000	564.000000	202.000000	6.200000

Figura 6 - Separação das variáveis numéricas e categóricas

Aqui separei as variáveis numéricas das variáveis categorias de maneira a conseguir ter mais informações sobre as variáveis numéricas para entender melhor a população em análise, as médias dos valores por pessoa.

As variáveis categóricas não são relevantes analisar neste tipo de tabela, uma vez que costumam ser valores já definidos que categorizam algo. (Por exemplo: Target é categórica porque só pode ser 0 ou 1, dependendo se o paciente tem problemas cardíacos ou não).

3.5. Análise das variáveis numéricas

- Podemos verificar através deste gráfico que a maior parte da minha população são pessoas com 58, 57 e 54 anos.

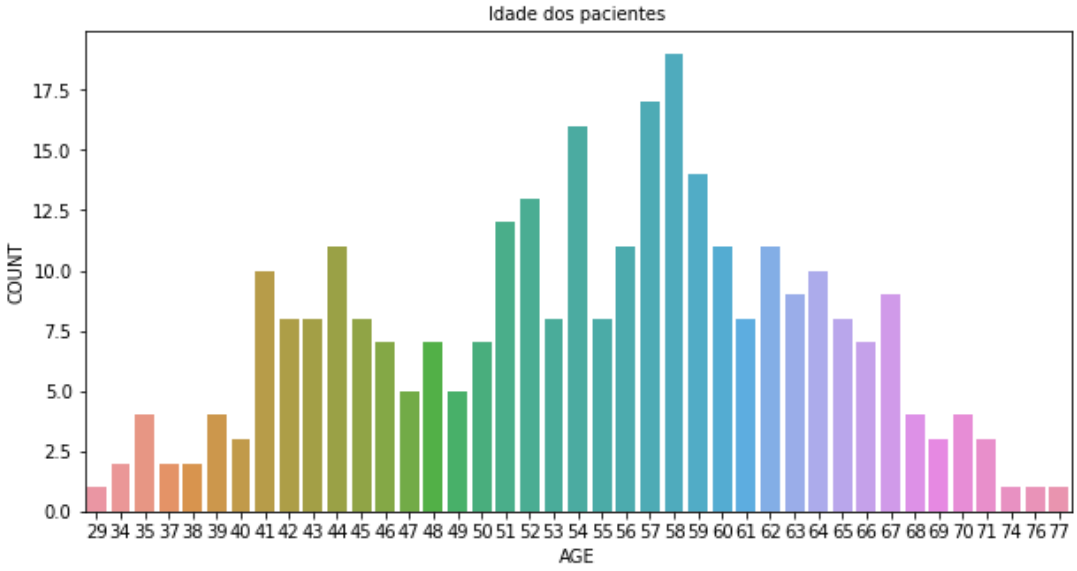

Figura 7 - Distribuição dos pacientes por idade

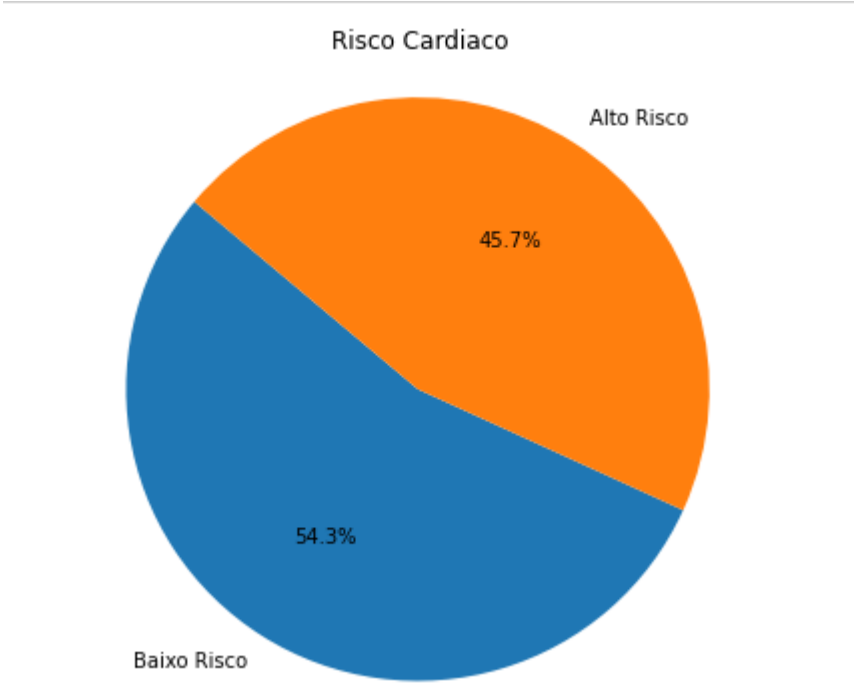

Figura 8 - Distribuição dos pacientes por grupos de risco

- Baseado no campo target fiz este gráfico para verificar a minha população de alto risco e comparar com a população de baixo risco (pessoas que têm problemas cardíacos vs pessoas que não têm).

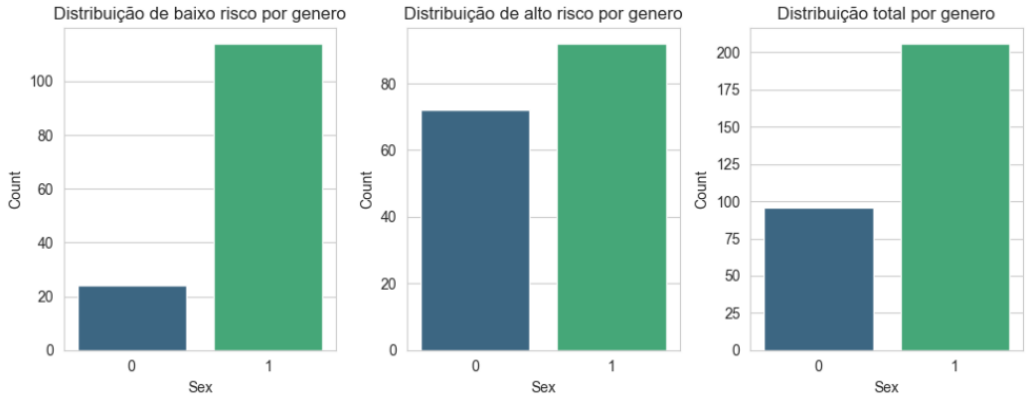

Figura 9 - Distribuição dos pacientes por genero e por risco

- Aqui está a distribuição das pessoas por gênero dentro cada grupo de risco e o total de pessoas de cada género também. Podemos verificar que o nosso dataset tem menos de 100 pessoas do género 0 (feminino) enquanto tem mais de 200 pessoas do género 1 (masculino).
- Para obter uma proporção e estimativa, decidi dividir o número de pessoas do género 1 (masculino) por 2, para obter um valor mais próximo do número de pessoas do género 0 (feminino) e conseguir ter uma ideia de qual género está mais propício a pertencer ao grupo de risco.
- Distribuição de baixo risco: Um pouco mais de 20 pessoas do género 0 pertencem ao grupo de baixo risco, enquanto (mais ou menos 110/2) 55 pessoas do género 1 pertencem ao grupo de baixo risco.
- Distribuição de alto risco: Um pouco mais de 70 pessoas do género 0 pertencem ao grupo de alto risco enquanto (mais ou menos 110/2) 55 pessoas do género 1 pertencem ao grupo de alto risco.
- Podemos então afirmar que pessoas do género 1 (masculino) no geral costumam ter menos probabilidade de ter doenças cardíacas e pessoas do género 0 (feminino) costumam ter mais probabilidade de ter doenças cardíacas, ou seja, o género é um dos fatores que contribui para aumentar o risco de ataque cardíaco.

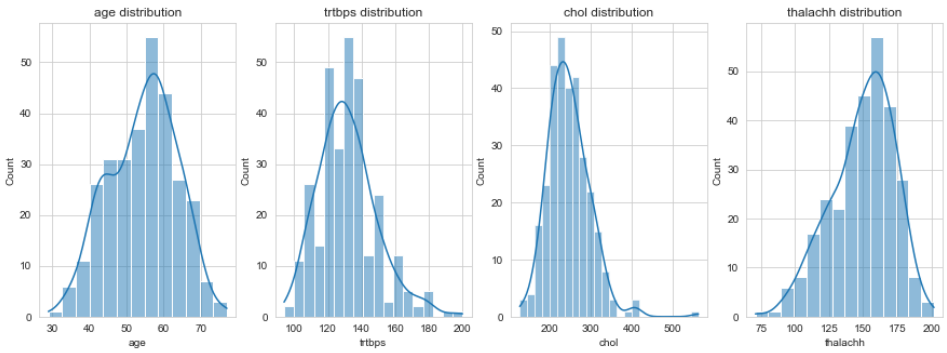

Figura 10 - Distribuição da população por idade, pressão, diabetes e batimento cardíaco máximo

- **Distribuição da idade (age):** A maioria das pessoas deste caso de estudo estão entre 40 e 70 anos
- **Distribuição de colestrol (chol):** A maioria dos valores de colesterol está entre 200 e 350 mg/dL. A população estudada tem níveis de colesterol que variam principalmente entre a faixa limite e alta, com alguns casos de hipercolesterolemia grave.
- **Distribuição da Frequência Cardíaca Máxima (thalachh):** A maioria das leituras de frequência cardíaca máxima está entre 100 e 180 bpm. A frequência cardíaca máxima da população estudada está dentro do esperado para adultos. A frequência cardíaca máxima mais alta pode indicar boa capacidade cardiovascular, enquanto valores mais baixos podem ser um indicador de piores condições de saúde ou maior idade.

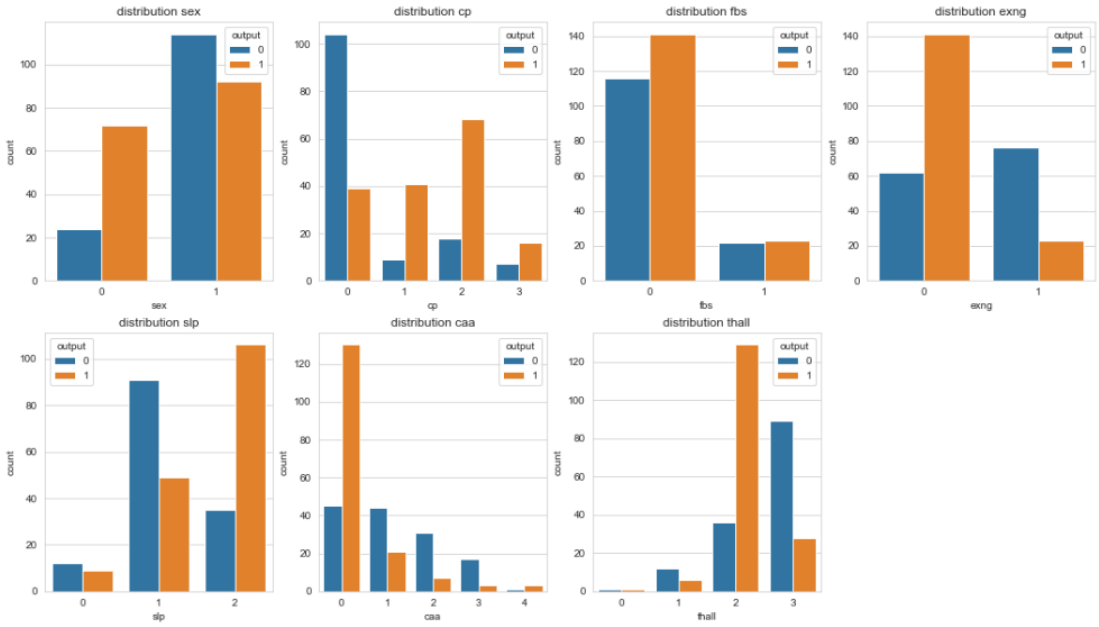

Figura 11 - Divisão de todos os dados por grupo de risco

- Neste caso de estudo a maioria dos pacientes do genero 1 (masculino) tem mais probabilidade de ter ataques cardíacos em comparação com o genero 0 (feminino). No entanto, há bem mais pessoas do genero 1 na amostra geral.

- Pacientes com cp = 1 (angina típica) e cp = 2 (angina atípica) têm maior probabilidade de ataque cardíaco (output = 1), enquanto cp = 0 (assintomático) não está tão associado a ataques cardíacos.

- A maioria dos pacientes, tanto com maior quanto com menor chance de terem ataques cardíacos, não tem elevados niveis de glicemia em jejum. No entanto, a presença de elevados niveis de glicemia não parecem ser um fator muito significativo para a chance de ter ataques cardíacos.

- A maioria dos pacientes que não têm angina induzida por exercício (exng = 0) têm maior chance de terem ataques cardíacos. A presença de exng parece estar associada a uma maior incidência de ataques cardíacos.

- Pacientes com slp = 2 (ascendente) têm maior probabilidade de terem ataques cardíacos, enquanto aqueles com slp = 0 (descendente) têm menor probabilidade.

- Pacientes com caa = 0 têm uma menor chance de terem ataques cardíacos, enquanto um aumento no número de vasos coloridos (caa = 1 a 3) está associado a uma maior probabilidade de ataque cardíaco.

- Pacientes com thall = 2 (defeito reversível) estão associados a uma maior chance de terem ataques cardíacos. Os resultados thall = 1 e 3 também estão associados a uma maior probabilidade incidência de ataques cardíacos em comparação com thall = 0.

3.6. Relação entre variáveis

Correlação das variáveis com a variável Target (chance de ataque cardíaco):

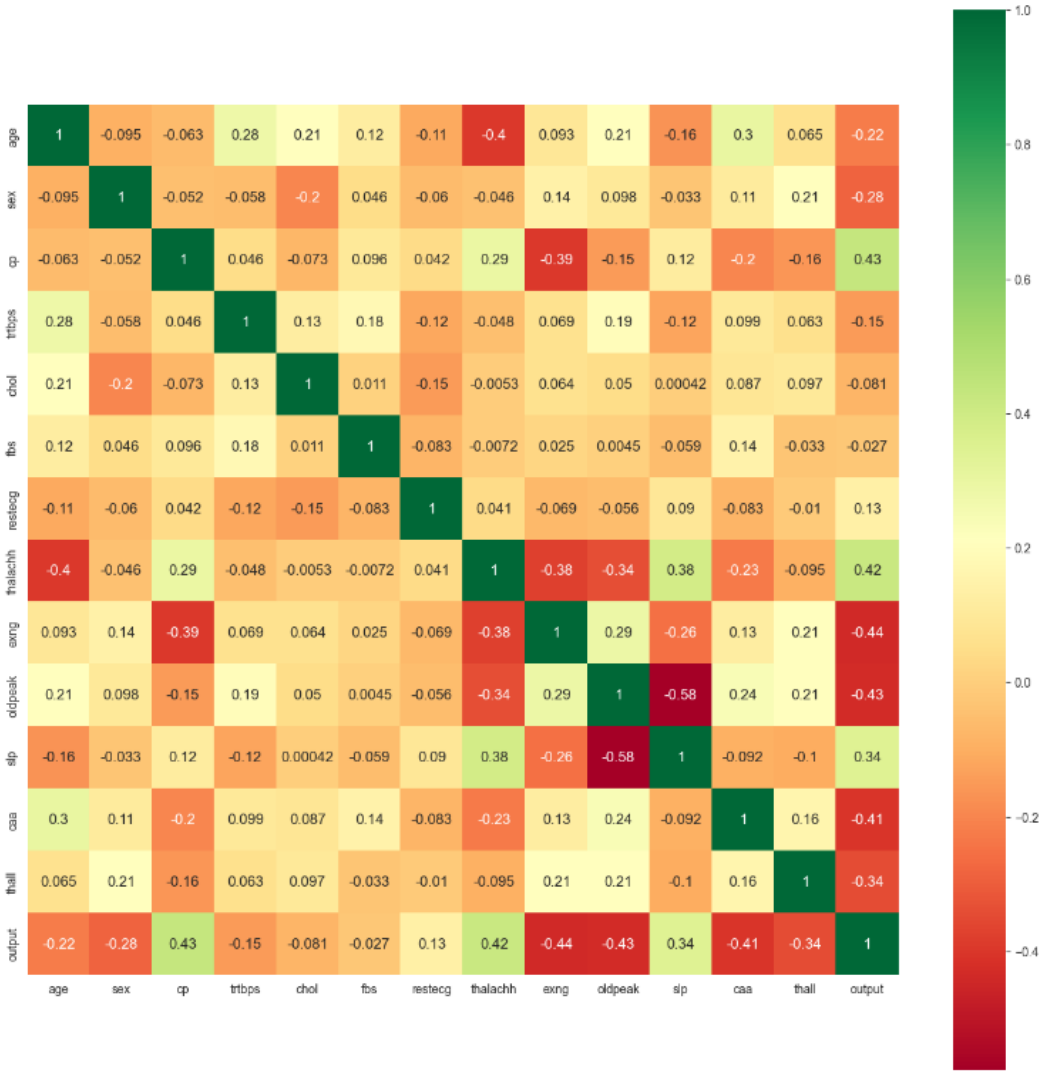

Figura 12 - Matriz de correlação

- Variáveis mais correlacionadas positivamente: cp (0.43) e ca (0.35).
- Variáveis mais correlacionadas negativamente: thalach (-0.43), oldpeak (-0.43), exng (-0.44), thal (-0.36), slp (-0.36), age (-0.23), e sex (-0.28).

Podemos através disto concluir o seguinte:

- Pessoas mais jovens tendem a ter frequências cardíacas máximas mais altas e menor probabilidade de ter ataque cardíaco.
- O tipo de dor no peito (cp) tem uma forte correlação positiva com o resultado (target), o que nos indica que diferentes tipos de dor no peito podem ser um fator significativo para prever a chanced de ataque cardíaco.

- A angina induzida por exercício (exng) e a depressão do ST induzida pelo exercício (oldpeak) tem uma correlação negativa com a chance de ter ataque cardíaco, indicando que esses fatores não são muito relevantes para o aumento do risco.

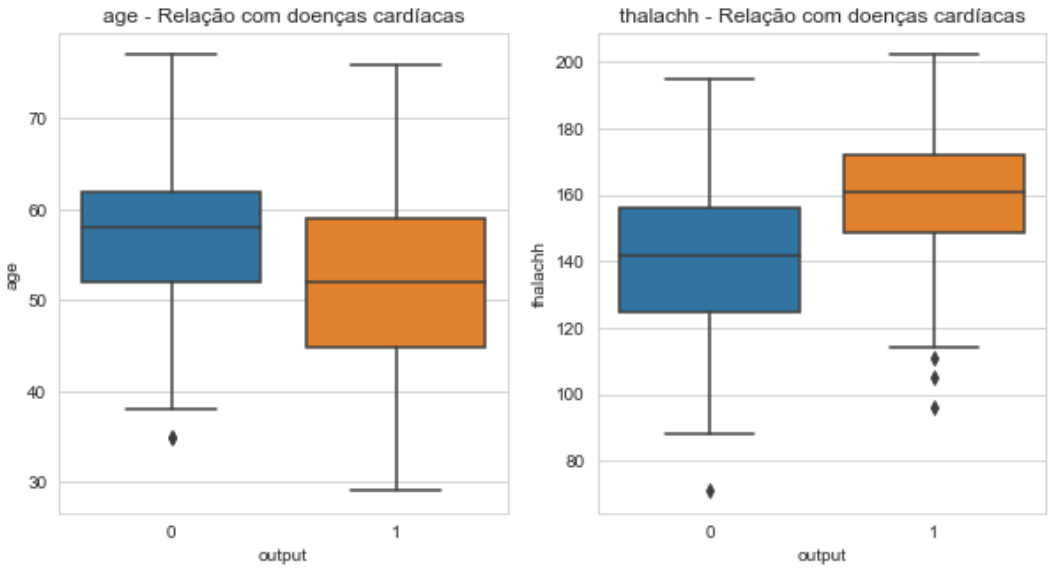

Figura 13 - Relação da idade e batimento cardíaco máximo com doenças cardíacas

- A mediana da idade para pacientes com doenças cardíacas (~52 anos) é um pouco menor que a mediana da idade para pacientes sem doenças cardíacas (~57 anos).
- A faixa de idade é bastante similar para ambos os grupos, mas há uma tendência de pacientes sem doenças cardíacas serem ligeiramente mais velhos.

- A mediana da frequência cardíaca máxima para pacientes com doenças cardíacas (160 bpm) é significativamente maior do que para pacientes sem doenças cardíacas (140 bpm).
- Pacientes com doenças cardíacas tendem a ter uma frequência cardíaca máxima mais alta, indicando uma possível relação entre a alta frequência cardíaca e a presença de doenças cardíacas.

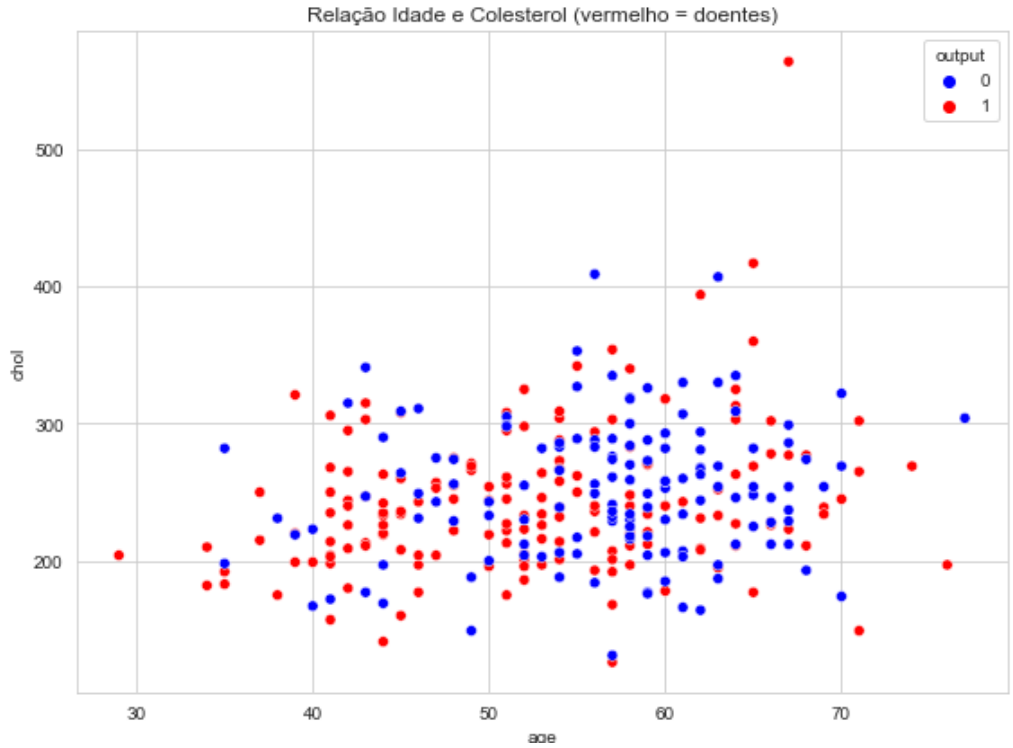

Figura 14 - Relação da idade com o colestrol

O gráfico de dispersão mostra a relação entre a idade (age) e os níveis de colesterol (chol) dos pacientes. A cord dos pontos indica se o paciente tem uma doença cardíaca (vermelho) ou não (azul).

Distribuição Geral:

- A maioria dos pontos estão concentrados entre os 40 e 70 anos de idade, com níveis de colesterol variando por volta dos 150 a 350 mg/dL.
- Há uma dispersão considerável nos níveis de colesterol, com alguns valores extremos (outliers) acima de 400 mg/dL.

Pacientes com e sem Doenças Cardíacas

- Estes pacientes estão distribuídos por toda a faixa etária, desde 30 a 70 anos, sem haver concentração numa faixa específica de idade.
- Os níveis de colesterol variam bastante tanto para pacientes com doenças como para pacientes sem doenças cardíacas. Não há uma clara distinção de que níveis mais altos ou mais baixos de colesterol sejam estejam relacionados a qualquer um dos grupos.
- A sobreposição significativa entre os níveis de colesterol dos dois grupos sugere que, embora o colesterol seja um fator importante na saúde cardíaca, ele sozinho pode não ser um indicador suficiente para prever a presença de doenças cardíacas.

4. Modelos implementados e resultados

a) Definição das Features e do Target

- Features: Refere-se ás colunas que serão usadas como características (features) do modelo.
- X: Refere-se ao DataFrame apenas com as colunas definidas em features.
- y: Refere-se á coluna output que é a variável alvo (target).

```
features = ['age', 'sex', 'cp', 'trtbps', 'chol', 'fbs', 'restecg', 'thalachh', 'exng', 'caa', 'thall', 'slp']
X = data[features]
y = data['output']
```

Figura 15 - Definição das Features e do Target

b) Transformação de variáveis categóricas

Converti algumas variáveis categóricas binárias em variáveis dummy (0 ou 1).

```
X = pd.get_dummies(X, columns=['cp', 'restecg', 'thall'], drop_first=True)
```

c) Divisão dos dados para treino e teste

A divisão dos dados de treino e teste é sempre a mesma para cada um dos modelos que eu escolhi. Dividi 20% dos dados para serem utilizados pelo conjunto de teste, enquanto os 80% restantes são usados para o conjunto de treino.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y)
```

d) Padronização dos dados

- **StandardScaler:** Padroniza os dados para que tenham média 0 e desvio padrão 1.
- **fit_transform e transform:** Ajusta o scaler aos dados de treino e, em seguida, transforma os dados. Isso significa que o scaler faz o calculo da média e do desvio padrão dos dados de treino e usa esses valores para padronizar os dados.

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figura 16 - Padronização dos dados

e) Treinamento do modelo

Cria e treina um modelo escolhido com os dados de treino escalados (neste caso podemos ver um exemplo utilizando Logistic Regression).

```
model = LogisticRegression(random_state=123)
model.fit(X_train_scaled, y_train)
```

Figura 17 - Treinamento do modelo

f) Predicts

- **predict:** Gera previsões de classe para os dados de teste.
- **predict_proba:** Gera previsões de probabilidade para a classe positiva.

```
y_pred = model.predict(X_test_scaled)
y_pred_proba = model.predict_proba(X_test_scaled)[:, 1]
```

Figura 18 - Gerar predicts

g) Avaliação do Modelo

Antes de perceber as métricas é importante entender o que são verdadeiros e falsos positivos e verdadeiros e falsos negativos.

- **Verdadeiros Positivos:** São os casos em que o modelo previu corretamente que um paciente está em risco de sofrer um ataque cardíaco, e de fato o paciente está em risco, ou seja, modelo acertou ao identificar este caso como caso positivo.

- **Falsos Positivos:** São os casos em que o modelo previu incorretamente que um paciente está em risco de sofrer um ataque cardíaco, mas na verdade o paciente não está em risco, ou seja, o modelo identificou este caso como positivo quando na verdade era negativo.

- **Verdadeiros Negativos:** São os casos em que o modelo previu corretamente que um paciente não está em risco de sofrer um ataque cardíaco, e de fato o paciente não está em risco, ou seja, o modelo conseguiu identificar corretamente um caso negativo.

- **Falsos Negativos:** São os casos em que o modelo previu incorretamente que um paciente não está em risco de sofrer um ataque cardíaco, mas na verdade o paciente está em risco, ou seja, o modelo indicou um resultado negativo quando na verdade era positivo.

  Para ter algumas métricas e entender a eficiência do meu modelo imprimi a accuracy e o classification report (precision, recall, f1-score).

  - A accuracy é a quantidade de previsões corretas (tanto verdadeiros positivos quanto verdadeiros negativos) em relação ao total de previsões realizadas.

- A precision é a quantidade de verdadeiros positivos em relação ao total de previsões positivas (soma dos verdadeiros positivos e falsos positivos).

- O recall é a quantidade de verdadeiros positivos em relação ao total de exemplos positivos reais (soma dos verdadeiros positivos e falsos negativos).

- O F1-score é a média da precision e do recall. Esta métrica equilibra as duas, sendo útil quando precisamos de um balanço entre a precisão e o recall.

```
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Figura 19 - Métricas do modelo

h) Função de previsão

A função de predict recebe um conjunto de dados como input (input_data), os quais são utilizados pelo modelo para fazer a previsão de uma pessoa com as características introduzidas ter um ataque cardíaco.

```
def predict_heart_attack(input_data):
    input_df = pd.DataFrame([input_data], columns=features)
    input_df = pd.get_dummies(input_df, columns=['cp', 'restecg', 'thall'], drop_first=True)
    input_df = input_df.reindex(columns=X_train.columns, fill_value=0)
    input_scaled = scaler.transform(input_df)
    probability = model.predict_proba(input_scaled)[:, 1]
    return probability[0]
```

Figura 20 - Função de previsão

i) Resultados

Através da função predict é calculada a probabilidade uma pessoa com as características introduzidas ter um ataque cardíaco e essa probabilidade é apresentada no final. Após isso o modelo guarda os seus dados para o estudo do problema num ficheiro pkl que é posteriormente utilizado para o front-end trazer resultados para esse modelo.

```
probability = predict_heart_attack(input_data)
print(f'Probabilidade de ataque cardíaco: {probability:.2f}')

joblib.dump(model, 'linear_regression_model.pkl')
```

Figura 21 - Apresentação dos resultados

4.1 Logistic Regression

A regressão logística é um algoritmo de machine learning utilizado para problemas de classificação, apesar de ter a palavra regressão no nome. Seguidamene podemos ver algumas métricas do modelo de regressão logistica implementado e podemos tirar algumas conclusões:

- **Accuracy (0.8):**

A accuracy indica a quantidade de previsões corretas que o modelo fez em relação ao número total que era previsto. Neste caso, o modelo possui uma accuracy de 80%, o que significa que 80% das previsões feitas pelo modelo estão corretas.

- **Precision:**

Classe 0 (baixo risco de ataque cardíaco): 0.90

- A precision para a classe 0 indica que 90% das previsões que o modelo fez para esta classe estão corretas. Quando o modelo prevê que um paciente não está em risco de ataque cardíaco, ele está correto 90% das vezes.

Classe 1 (alto risco de ataque cardíaco): 0.76

- Para a classe 1, a precision indica que 76% das previsões de alto risco de ataque cardíaco feitas pelo modelo estão corretas.

- Recall:

Classe 0 (baixo risco de ataque cardíaco): 0.64

- O recall para a classe 0 mostra que o modelo capturou corretamente 64% dos casos onde existia um baixo risco de ataque cardíaco, em relação ao total de casos dos verdadeiros negativos.

Classe 1 (alto risco de ataque cardíaco): 0.94

- O recall é alto para a classe 1 (94%), indicando que o modelo identificou corretamente a grande maioria dos casos de alto risco de ataque cardíaco, em relação ao total de verdadeiros positivos.

- F1-score:

- O F1-score é de cerca de 0.80, o que nos indica que o modelo está a conseguir um bom equilíbrio entre a precisão (capacidade de prever corretamente os casos positivos) e o recall (capacidade de identificar todos os casos positivos).

```
Accuracy: 0.8032786885245902
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.64      0.75        28
           1       0.76      0.94      0.84        33

    accuracy                           0.80        61
   macro avg       0.83      0.79      0.79        61
weighted avg       0.82      0.80      0.80        61
```

Figura 22 - Métricas do modelo Linear Regression

- **Análise de curva ROC e matriz de confusão**

Com base na matriz de confusão e no gráfico da curva ROC, parece que o meu modelo de previsão de ataque cardíaco está a funcionar bem. Os valores de precisão, recall e F1-score sugerem que o modelo pode classificar corretamente uma boa parte dos pacientes. A curva ROC confirma isso e indica-nos uma boa capacidade do modelo distinguir as classes positivas e negativas. Podemos ver um AUC de 0.79 o que sugere um ótimo desempenho uma vez que acima de 0.7 é considerado como um desempenho sólido.

Em relação á classe 0 (baixo risco de ter ataque cardíaco), o modelo classificou 18 pessoas com baixo risco de ter ataque cardíaco e realmente pertentem a esta classe, enquanto que outras 10 pessoas foram classificadas também com baixo risco de ter ataque cardíaco mas pertencem á classe 1 (alto risco de ter ataque cardíaco).

Em relação á classe 1 (alto risco de ter ataque cardíaco), o modelo classificou 2 pessoas com baixo risco de ter ataque cardíaco e errou, porque essas pessoas pertencem á classe 1. Por outro lado o modelo acertou em 31 casos, classificando-os como pertencentes á classe de alto risco.

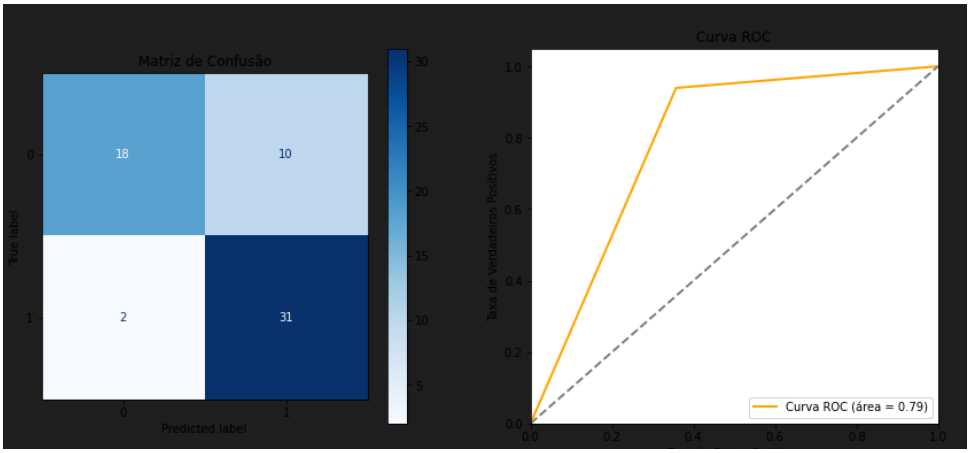

Figura 23 - Métricas do modelo Linear Regression

4.1. Random Forest

O Random Forest é um algoritmo de machine learning que usa vários subconjuntos de dados de treinamento para construir uma série de árvores de decisão. A accuracy é ligeiramente menor neste modelo, o que á partida nos indica que ele é menos eficiente do que o Logistic Regression.

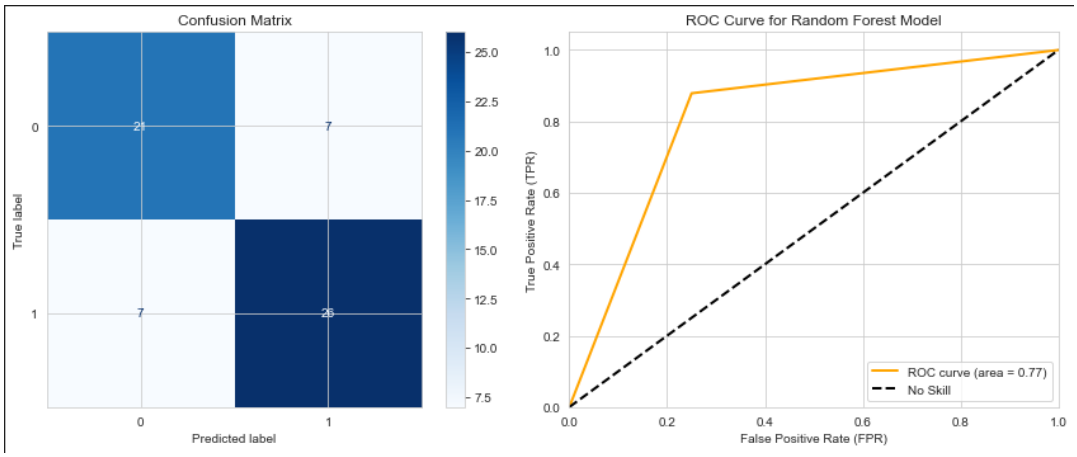

Figura 25 - Métricas do modelo Random Forest

```
Accuracy: 0.7704918032786885
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.75      0.75        28
           1       0.79      0.79      0.79        33

    accuracy                           0.77        61
   macro avg       0.77      0.77      0.77        61
weighted avg       0.77      0.77      0.77        61
```

Figura 24 - Métricas do modelo Random Forest

Através de um gráfico de feature importance descobri o possível motivo para a accuracy ser pior neste modelo, uma vez que ele dá bastante importância a features como o “caa”, “exng” e “thall” que supostamente não deveriam ter uma forte relação com problemas cardíacos e dessa forma influenciar muito na probabilidade de ter um ataque cardíaco.

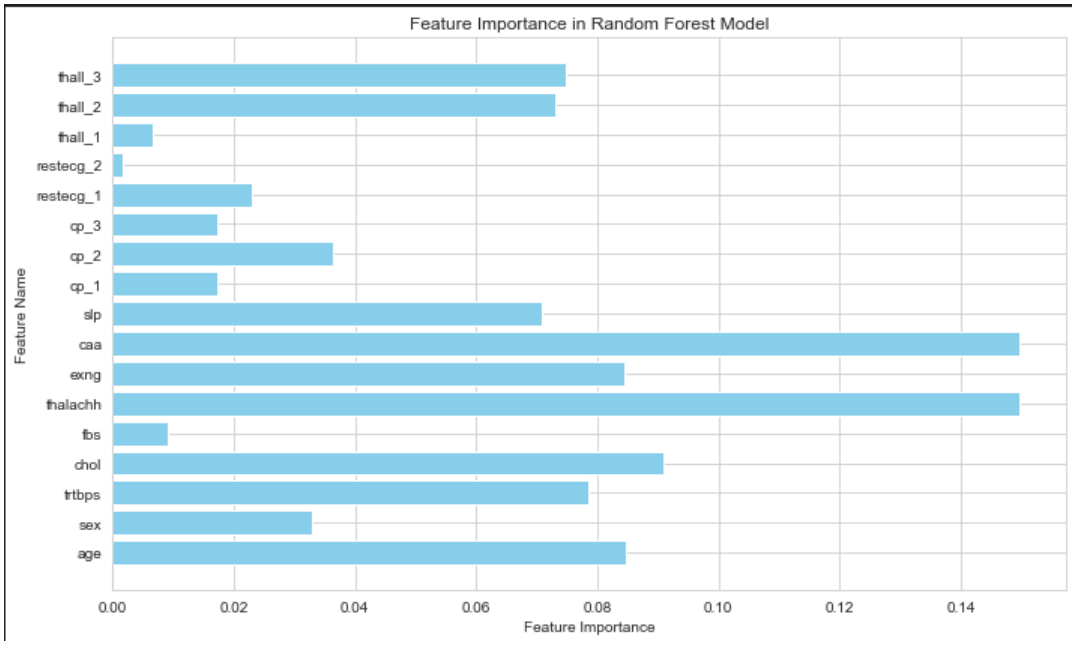

Figura 26 - Métricas do modelo Random Forest

4.2. Support Vector Machine

O SVM é baseado em métodos de aprendizagem que analisam os dados e reconhecem padrões, sendo este usado para classificação e análise de regressão.

Podemos ver que, dos 3 modelos escolhidos, este foi o modelo com maior accuracy, o que nos indica que á partida este pode ser o modelo mais eficiente para o nosso problema.

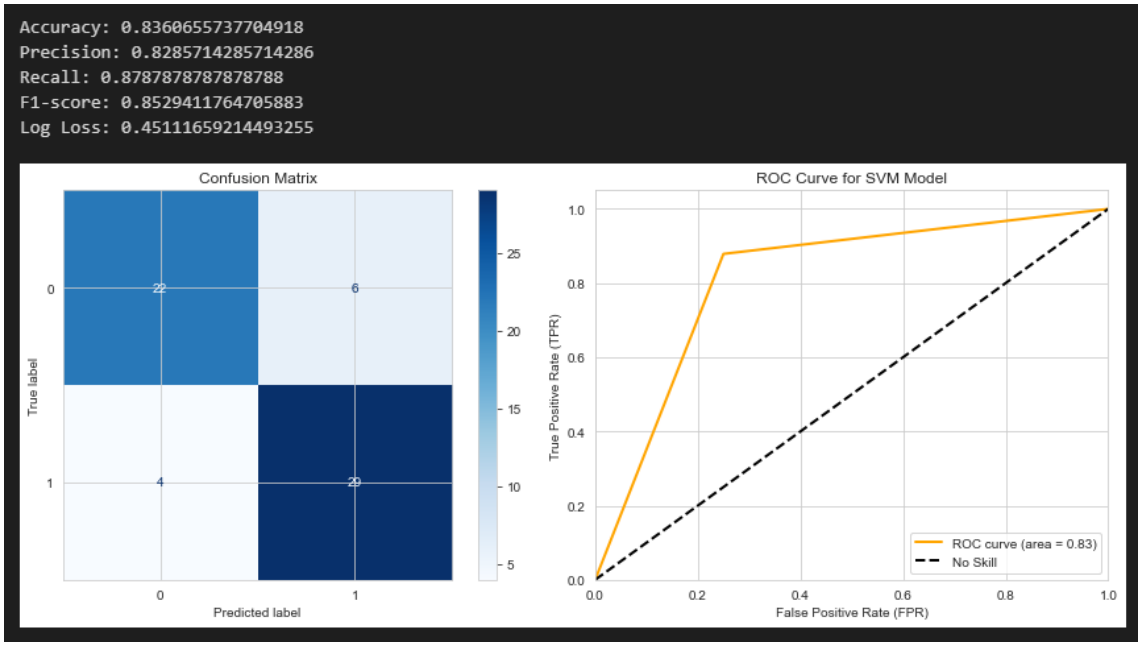

Figura 27 - Métricas do modelo SVM

5. Interface Gráfica

Para tornar capaz a interação entre o modelo treinado e os dados de input, criei uma interface gráfica com os dados mais relevantes segundo a matriz de correlação e após fazer o cálculo da probabilidade coloquei as métricas a serem apresentadas. Nesta interface é possível selecionar o modelo que queremos utilizar de entre os 3 modelos desenvolvidos.

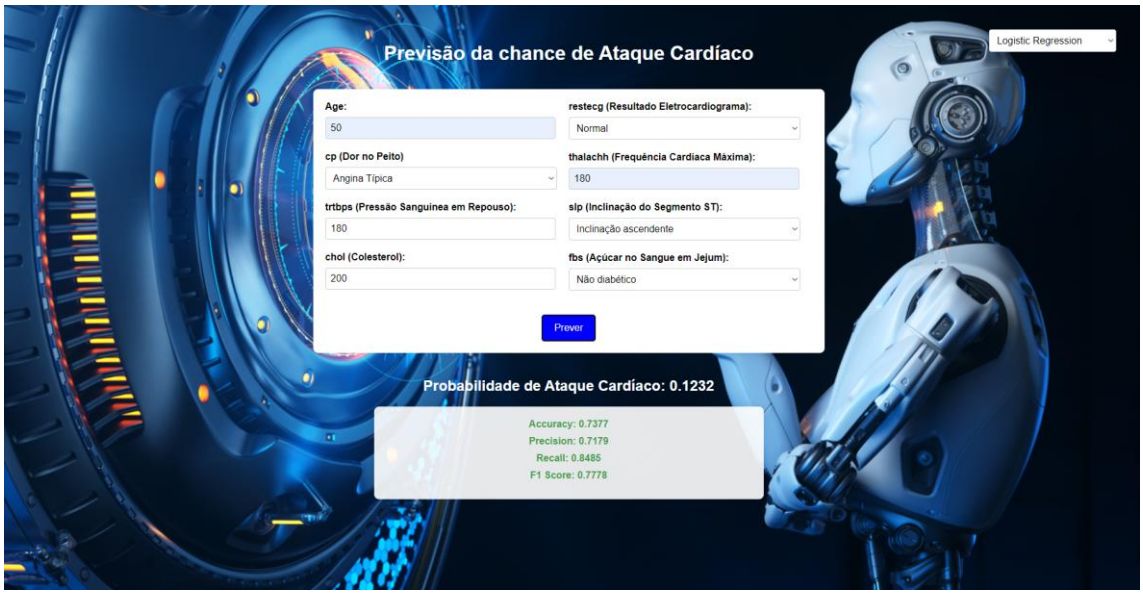

Figura 28 - Interface gráfica

6. Conclusão

Este projeto demonstrou a aplicação prática de técnicas de machine learning para resolver um problema real na área da saúde. Através de um processo iterativo de modelagem e validação, consegui desenvolver um modelo capaz de prever a probabilidade de alguem ter um ataque cardíacos. Os objetivos do projeto foram alcançados, integrando conhecimento teórico e prático em Inteligência Artificial, e proporcionando experiência na modelagem, análise crítica e melhoria de soluções baseadas em machine learning.