Preprocessament i anàlisi del dataset "Heart Attack Analysis & Dataset"

Alumnes: Marina Arias Queralt i Jordi Pomada Foix

Tipologia i Cicle de Vida de les Dades

13 de juny de 2023

En aquest document fem les fases de selecció, integració i neteja de les dades. L'anàlisi i la presentació de resultats es farà en altres documents.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

1 i 2. Descripció, integració i selecció de les dades

El joc de dades està composat per dos fitxers csv:

```
- heart.csv
- o2Saturation.csv
```

Es poden descarregar de Kaggle.

Aquests dos fitxers csv contindran variables diferents que es correspondran a mesures preses a pacients que estan en ingres intrahospitalari. L'objectiu d'aquesta pràctica es donar resposta a la pregunta de quins paràmetres fan que un pacient tingui major probabilitat de patir un atac de cor. Gràcies a aquest anàlisis es podrà compendre quins factors influeixen augmentan el risc d'atac de cor i en quins rang de valor cal que estiguin situats per prevenir-los.

```
data_heart = pd.read_csv('input/heart.csv')
data_heart.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trtbps    303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
```

```
 7   thalachh  303 non-null    int64
 8   exng      303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slp       303 non-null    int64
 11  caa       303 non-null    int64
 12  thall     303 non-null    int64
 13  output    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

El joc de dades està compost per 303 observacions de 14 variables, on totes les variables són de tipus enter excepte la variable oldpeak, que és de tipus numèric. El significat de cada variable és el següent:

1. age: Edat del pacient.
2. sex: Sexe del pacient.
3. cp: Tipus de dolor de pit, codificat:
   - 1: angina típica.
   - 2: angina atípica.
   - 3: dolor no anginal.
   - 4: asimptomàtic.
4. trtbps: pressió arterial en repos, mesurada en milímetres de mercuri.
5. chol: colesterol en mg/dl.
6. fbs: sucre en sang a dejú major que 120 mg/dl.
7. restecg: resultats de prova electrocardiogràfica en repòs.
   - 0: normal.
   - 1: anormalitat en l'ona ST-T.
   - 2: hipertròfia ventricular esquerra, probable o segura.
8. thalachh: màxima freqüència cardíaca assolit.
9. exng: angina induïda per l'exercici, cert o fals.
10. oldpeak: depresió del ST induÏda per el exercici físic en relació al repos (mm).
11. slp: pendent del pic del segment ST d'exercici.
12. caa: número de vassos majors.
13. thall: presencia de talassemia.
    - 0: normal.
    - 1: defecte arreglat.
    - 2: defecte reversible.
14. output: menys probabilitat d'atac de cor (0) o més probabilitat d'atac de cor (1).

```
data_o2 = pd.read_csv('input/o2Saturation.csv')
data_o2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3585 entries, 0 to 3584
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
```

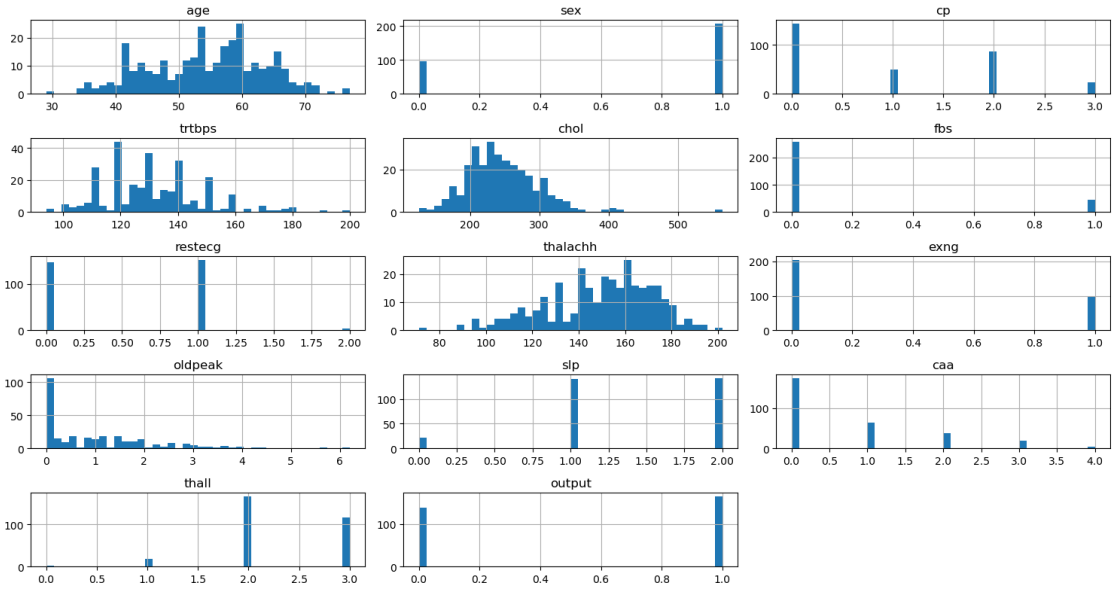
```
---   ------   --------------   -----
 0    98.6     3585 non-null    float64
dtypes: float64(1)
memory usage: 28.1 KB
```

El fitxer 02Saturation.csv està compost per 3585 observacions d'una sola variable. En la descripció del dataset a la web no s'explica quina relació té aquest fitxer amb l'altre, ni tan sols el seu significat. Per tant decidim no analitzar aquest fitxer, ja que no en podrem extreure conclusions sòlides.

3. Neteja de les dades

Com hem vist no hi ha valors nuls com a tals al joc de dades, però aquests podrien estar codificats en algun valor específic no nul. No s'especifica a la descripció del dataset, així que intentem veure-ho amb histogrames.

```
data_heart.hist(bins=40, layout=(5, 3), figsize=(15, 8))
plt.tight_layout()
plt.show()
```



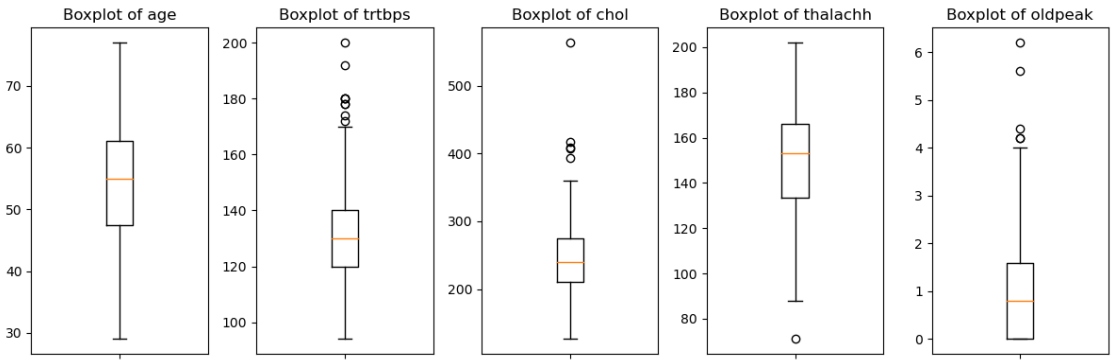
No s'observen irregularitats en les dades que indiquin manca de qualitat en aquestes. Sí que podem observar, però, que tot i que la majoria de les variables són de tipus enter, de fet podem dividir-les entre categòriques i numèriques segons el codi de la cel·la següent.

```
num_vars = ['age', 'trtbps', 'chol', 'thalachh', 'oldpeak']
cat_vars = [v for v in data_heart.columns if v not in num_vars]
```

Podem veure també com la majoria de les variables categòriques estan desequilibrades. Finalment, veiem la classe oldpeak presenta una concentració d'observacions al zero.

Procedim a analitzar la presència de valors extrems en les diferents variables numèriques. Per fer aquest estudi, s'implementaràn diagrames de caixes el qual ens permetran detectar aquelles observacions que es troben molt allunyades de la resta.

```
fig, axes = plt.subplots(nrows=1, ncols=5, figsize=(12, 4))

for position, column_name in enumerate(num_vars):
    axes[position].boxplot(data_heart[column_name])
    axes[position].set_title('Boxplot of {}'.format(column_name))
    axes[position].set_xticklabels([])

plt.tight_layout()
plt.show()
```



Es pot observar que es detecten valors atípics a les variables "trtbps", "chol", "thalachh" i oldpeack.

A continuació determinaré si es tracta d'un valor extrem:

- "**Chol**":

  Els cuatre valors atípics detectats de colesterol no seràn extrems. Ja que segons aquest article article un valor de colesterol superior als 400 mg/dl és considera hipercolesterolèmia severa. Per tant, els valors atípics que es troben al voltant de 400 encara que es considerin elevats s'han de tenir en compte. Tot i així, el valor al voltant de 500 és considera molt elevat i molt poc habitual, com que es tracta d'una observació aïllada, es pot eliminar del estudi.

- "**Oldpeak**":

  Els cuatre valors atípics detectats de oldpeack no seràn extrems. Sabem que oldpeak referencia a la depresió de la ona ST induïda per el exercici fisic que es troba en mm i valors elevats d'aquest són indicadors de cardiopaties isquemicas. Segons el següent article, la depresió d'una ona ST sol situar-se al voltant d'entre 0.5 i 1mm però aquesta pot augmentar quan el pacient realitza esforç físic. Per tant, podem considerar que són valors que entren dins del rang.

- "**trtbps**":

Els sis valors atípics detectats de "trtbps" no seràn extrems. Sabem que trtbps referencia a la presió artesial en repós. Segons el següent article, les presions arterial superiors a és consideren crisis hipertensives, per tant, són valors atípics els que tenim però estan dins del rang de valors possibles.

- "**thalachh**":

  El valor atípic detectats de "thalachh" no serà extrem. Sabem que "thalachh" fa referència a la màxima freqüència cardíaca del pacient. Segons el següent article, la freqüencia cardíaca màxima depen directament de la edad i aquesta pot presentar valors menors en pacients que són esportistes d'elit, per tant el valor atípic entra dins del rang possible.

```
index_chol_over_500 = data_heart.loc[data_heart['chol'] > 500].index
data_heart_clean = data_heart.drop(index_chol_over_500, axis=0)
```

Finalment, afegim la variable log_oldpeak per estudiar la variable oldpeak de manera més efectiva, i eliminem la original.

```
data_heart_clean['log_oldpeak'] = data_heart['oldpeak'].apply(lambda
x: np.log(x+1))
data_heart_clean.drop('oldpeak', axis=1, inplace=True)
data_heart_clean.info()
```

```
<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   slp          302 non-null    int64
 10  caa          302 non-null    int64
 11  thall        302 non-null    int64
 12  output       302 non-null    int64
 13  log_oldpeak  302 non-null    float64
dtypes: float64(1), int64(13)
memory usage: 35.4 KB
```

Dataset resultant

Emmagatzemem les dades de data_heart_clean al fitxer data_heart_clean.csv.

```
data_heart_clean.to_csv('input/heart_clean.csv', index=False)
```

4. Anàlisi del dataset "Heart Attack Analysis & Dataset"

En aquest notebook analitzarem les dades del joc de dades un cop aquestes han estat netejades, és a dir, partim del fitxer data_heart_clean.csv.

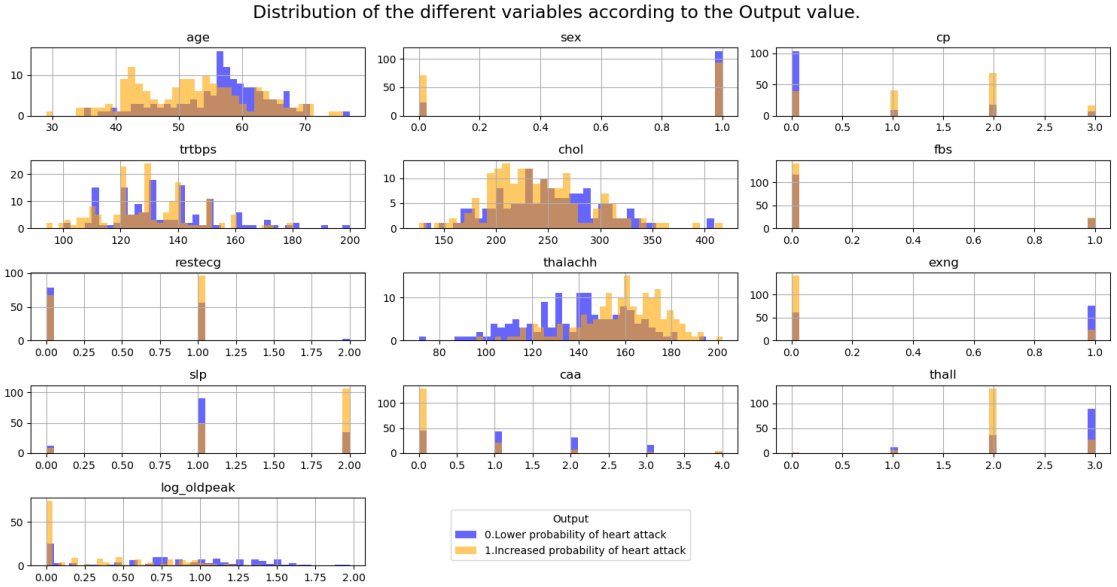
```
#import libraries
from scipy.stats import shapiro, bartlett, ttest_ind, chi2_contingency
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from statsmodels.stats.outliers_influence import
variance_inflation_factor

# load the cleaned dataset
data_heart = pd.read_csv('input/heart_clean.csv')

# make column names list for each type of variable
numerical_variable_names = ['age', 'trtbps', 'chol', 'thalachh',
'log_oldpeak']
categorical_variable_names = [v for v in data_heart.columns if v not
in numerical_variable_names]
```

4.1 Selecció dels grups de dades que es volen analitzar/comparar.

La primera part de l'anàlisi del joc de dades serà seleccionar quines variables volem utilitzar per estudiar quins paràmetres són indicadors d'una major probabilitat d'atac de cor. Per tal de fer aquesta selecció, primer analitzarem visualment com es distribueixen les variables explicatives depenent de si el pacient presenta major o menor probabilitat d'atac de cor. Aquesta probabilitat és la que bé proporcionada per variable "output", la qual és la variable a predir.

```
def plot_conditional_histograms(alpha=0.6):
    fig, axes = plt.subplots(5, 3, figsize=(15, 8))
    descriptive = [v for v in data_heart.columns if v != 'output']
    legend_ax = fig.add_subplot(111, frame_on=False)

    for i, v in enumerate(descriptive):
        ax = axes[i//3, i%3]
        data_heart.loc[data_heart['output']==0, v].hist(bins=40,
ax=ax, alpha=alpha, color='blue', label="0.Lower probability of heart
attack")
        data_heart.loc[data_heart['output']==1, v].hist(bins=40,
ax=ax, alpha=alpha, color='orange', label="1.Increased probability of
heart attack")
        ax.set_title(v)

    axes[4, 1].axis('off')
    axes[4, 2].axis('off')

    legend_ax.axis('off')
```

```
    legend_ax.legend(*ax.get_legend_handles_labels(), loc='lower
center', title='Output')
    fig.add_subplot(legend_ax)

    fig.suptitle('Distribution of the different variables according to
the Output value.', fontsize=17)
    fig.tight_layout()
    plt.show()

plot_conditional_histograms()
```



De primeres, podem afirmar que tots els histogrames obtinguts són prometedors, ja que les distribucions de les variables explicatives varien considerablement segons la categoria de la variable **Output**. A més, podem veure que algunes variables tenen una distribució més diferenciada que d'altres, fet que ens permetrà fer una idea inicial de quines variables explicatives seran les més rellevants per predir la variable **Output**.

A escala visual podem observar com les variables categòriques **thall**,**caa**,**slp**,**exng** i **cp** mostren una distinció més clara entre les dues categories de la variable **Output**. Això suggereix que aquestes variables explicatives seràn les que descriuran millor la variable a predir. Tot i així, en apartats posteriors realitzarem un estudi amb més profunditat per confirmar aquesta hipòtesi inicial. Pel que fa a les variables numèriques, podem veure que en totes elles hi ha una superposició els dos tipus d'histogrames corresponent a cada categoria de la variable **Output** . Per tant, en aquest cas no és possible portar a cap una hipòtesi inicial respecte a quines variables seran les més rellevants i, per tant, serà necessari fer un estudi en més profunditat per identificar-les.

El estudi que durem a terme per establir quines variables explicaràn millor la variable **Output** serà el següent:

- Aplicar test d'hipòtesis per a cadascuna les variables numèriques per tal de determinar si existeixen diferències significatives entre els valors d'aquestes variables segons a quina categoria de la variable **Output** pertanyen. Si hi ha diferències significatives, significa que són variables rellevant per la predicció de **Output**.

- Elaborarem una anàlisi de correlacions entre les diferents variables numèriques i la variable **Output** per determinar com es relacionen entre si, si existeix col·linealitat entre variables numèriques i el grau de correlació que tenen amb la variable **Output**.

- Farem una anàlisi de les variables categòriques conjuntament amb la variable de sortida, calculant els seus corresponents coeficients de V-Cramer, per tal de determinar com es relacionen entre si i quines variables són més rellevants per la predicció de la variable **Output**.

- Finalment, generarem un model de regresió logística utilitzant les variables numèriques i categòriques seleccionades anteriorment com a rellevants per a predir la variable **Output**. Gràcies a la generació d'aquest model, podrem determinar si les variables seleccionades expliquen de forma òptima la variable **Output** mitjançant el càlcul de l'exactitut del model generat.

4.2.a Anàlisi de la normalitat del joc de dades.

Procedim a elaborar una anàlisi de la normalitat de les variables numèriques que formen el joc de dades. Per fer aquesta anàlisi utilitzarem el test de Shapiro i Wilk.

El test de Shapiro i Wilk és un test d'hipòtesis que establirà com hipòtesi nul·la que la distribució de les observacions de la variable analitzada segueix una distribució normal. Per tant, mentre el p-value associat a la variable sigui superior al nivell de significança (en aquest cas establirem d'un 0,01), es podrà establir que hi ha evidència suficient per rebutjar la hipòtesis nul·la i afirmar que les dades segueixen una distribució no normal.

```
normality_results = {var: list(shapiro(data_heart[var])) for var in
numerical_variable_names}

pd.DataFrame(normality_results, index=['statistic', 'p-value'])
```

```
                age    trtbps      chol  thalachh   log_oldpeak
statistic  0.986622  0.966081  0.982803  0.976608  8.896660e-01
p-value    0.006696  0.000002  0.001096  0.000077  5.537369e-14
```

En aquest cas, es pot afirmar amb una confiança superior al 99% que cap de les variables numèriques es distribueix normalment.

Així i tot, cal tenir en compte que depenent del nombre de mostres que tinguem per les dues categories possibles de la variable **Output**, pot ser possible aplicar el teorema del límit central (TLC), el qual estableix que el contrast d'hipòtesis sobre la mitjana d'una

mostra s'aproxima a una distribució normal encara que la població original no segueixi aquesta distribució si la mostra té un nombre d'observacions elevat (aprox. major a 30).

```
# Check that there are more than 30 values for each value.
data_heart['output'].value_counts()
```

```
1    164
0    138
Name: output, dtype: int64
```

Si mirem la taula anterior, veiem que el nombre d'observacions per cadascuna de les categories de la variable **Outpot** és superior a 30, per tant, d'acord amb el TLC podem assumir que la mitjana mostral de les variables numèriques es distribueix normalment.

4.2.b Anàlisi de la homoscedasticitat del joc de dades.

Procedim a elaborar una anàlisi de la igualtat de variància entre les observacions de les variables numèriques segons el tipus de categoria de la variable **Output**. Per fer aquesta anàlisi utilitzarem el test de Bartlett.

El test de Bartlett és un test d'hipòtesis que s'aplica en variable distribuïda normalment, que establirà com hipòtesi nul·la que la variància de les observacions de la variable per cada categoria serà igual. Per tant, mentre el p-value associat a la variable sigui major al nivell de significança (en aquest cas establirem d'un 0,05), es podrà establir que les dades de la variable segons el tipus de **Output** compleixen el criteri d'homoscedasticitat.

```
homoscedasticity_results = {
    var: list(bartlett(
        data_heart[data_heart['output'] == 0][var],
        data_heart[data_heart['output'] == 1][var]
    )) for var in numerical_variable_names
}


homoscedasticty_results_df = pd.DataFrame(homoscedasticity_results,
index=['statistic', 'p-value'])
homoscedasticty_results_df
```

	age	trtbps	chol	thalachh	log_oldpeak
statistic	4.621701	3.192989	0.268626	3.87912	8.720618
p-value	0.031570	0.073955	0.604255	0.04889	0.003146

En aquest cas, podem confirmar amb un 95% de confiança que les variables numèriques **trtbps** i **chol** segons la categoria de la variable **Output** compleixen amb el criteri d'homoscedasticitat. Mentre que les variables **age**, **thalachh** i **log_oldpeak** compliran amb el criteri d'heteroscedasticitat.

4.3 Estudi estadístic de les variables seleccionades respecte la variable Output.

Test d'hipotesis

A continuació es portarà a terme un conjunt de test d'hipòtesis per determinar si existeixen diferències significatives entre els grups corresponents a cada categoria de la variable **Output** per les diferents variables numèriques del joc de dades. Aquelles variables que no presentin diferències significatives amb un 95% de confiança seran descartades del model de regressió logístic final.

Cal tenir en compte que en aquest cas les variables a analitzar poden presentar criteri d'homoscedasticitat o no, per tant, com a test d'hipòtesis aplicarem el t-test de Welch el qual és apte pels dos casos.

```
def check_equality_in_mean():
    n_num_vars = len(numerical_variable_names)
    results_table = pd.DataFrame(0, columns=numerical_variable_names,
index=['statistic', 'p-value'])
    for var in numerical_variable_names:
        data_0 = data_heart.loc[data_heart['output']==0, var]
        data_1 = data_heart.loc[data_heart['output']==1, var]
        _, p_mean = ttest_ind(data_0, data_1, equal_var=False)
        results_table.loc['statistic', var] = _
        results_table.loc['p-value', var] = p_mean
    return results_table

print("Results from the hipotesis test of means by restricting to each
class:")
equality_in_mean_results = check_equality_in_mean()
equality_in_mean_results
```

```
Results from the hipotesis test of means by restricting to each class:
```

	age	trtbps	chol	thalachh	log_oldpeak
statistic	4.169903	2.461914	1.930037	-7.930994e+00	8.205121e+00
p-value	0.000040	0.014439	0.054589	5.763131e-14	1.105115e-14

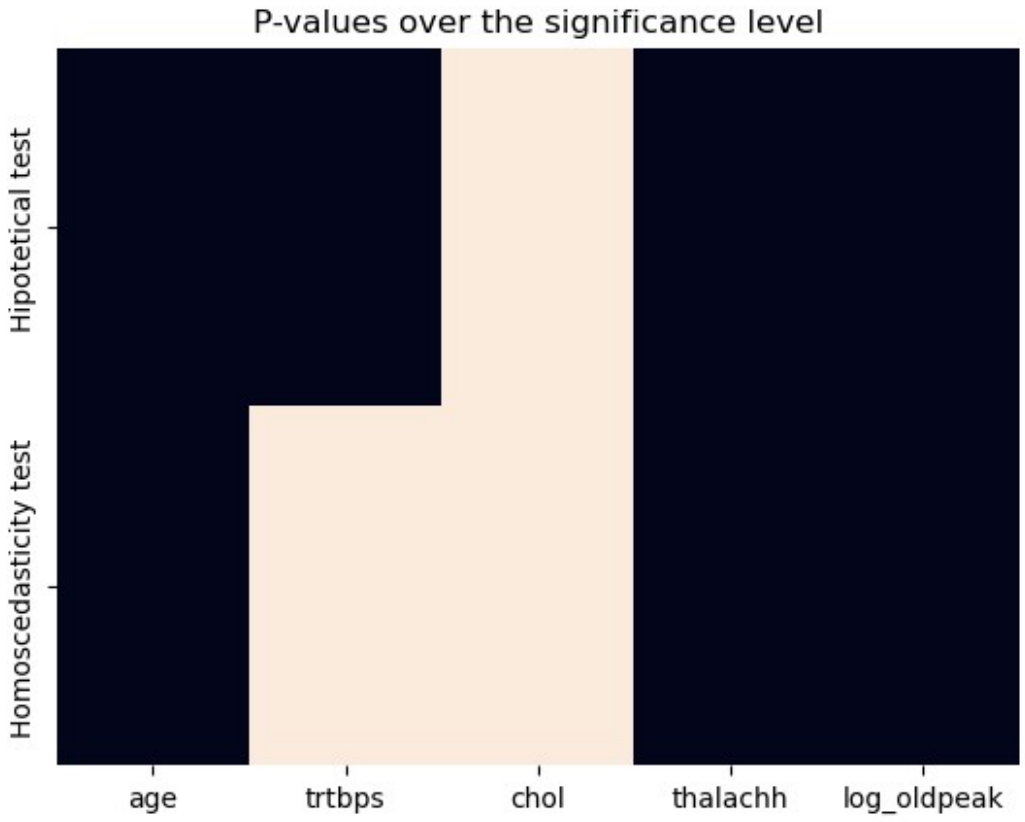
En aquest cas, podem confirmar amb un 95% de confiança que totes les variables numèriques tenen diferències significatives respecte a la categoria de la variable **Output** amb l'excepció de la variable **chol**.

Amb els resultats obtinguts tant en el test d'hipòtesis com en l'anàlisi d'homoscedasticitat realitzats anteriorment, podem conclure que tant la variable numèrica **chol** com la variable numèrica **trtbps** aporten menys informació respecte a la variable **output** que la resta de variables numèriques. En conseqüència, no seran considerades com a rellevants per explicar la variable a predir.

Podem veure la representació gràfica dels p-values de cada variable en el següent gràfic (p -values majors (en ocre) o menors (en negre) al nivell de significança de 0,05).

```
p_mean = pd.DataFrame(equality_in_mean_results.loc['p-
value']).rename(columns={'p-value': 'Hipotetical test'}).transpose()
p_variance = pd.DataFrame(homoscedasticty_results_df.loc['p-
value']).rename(columns={'p-value':

'Homoscedasticity test'}).transpose()

p_values_df = pd.concat([p_mean, p_variance])

p_values_heatmap = sns.heatmap(p_values_df > 0.05, cbar=False)
p_values_heatmap.set_title("P-values over the significance level")
```

```
Text(0.5, 1.0, 'P-values over the significance level')
```

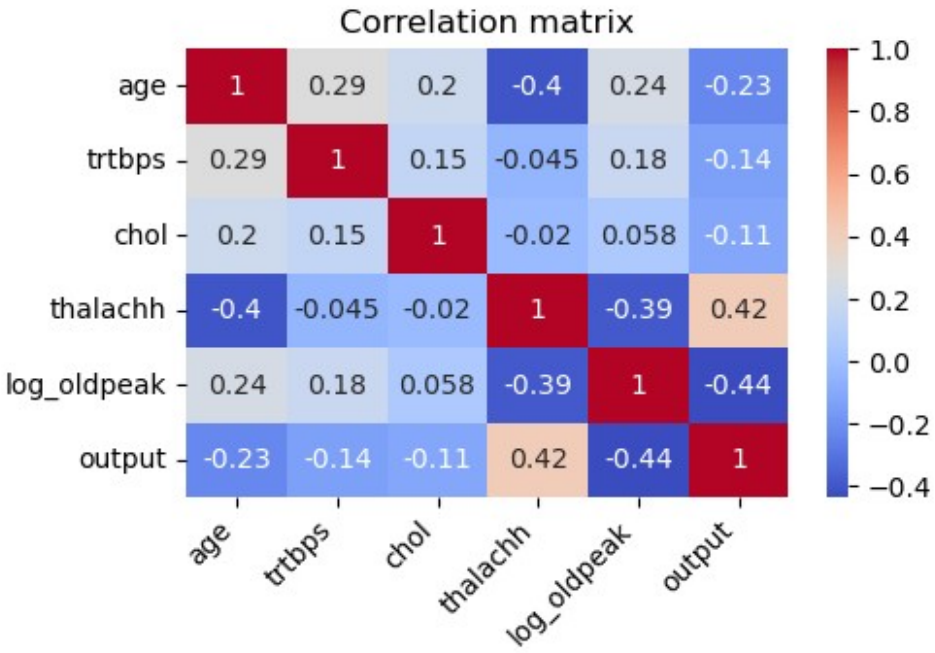


Matriu de correlació de les variables numèriques.

A continuació realitzarem una matriu de correlacions que ens permetrà determinar com es relacionen les diferents variables numèriques entre si i amb la variable **Output**.

```
numerical_variable_names_with_output = ['age', 'trtbps', 'chol',
'thalachh', 'log_oldpeak', 'output']
numerical_data_heart =
data_heart[numerical_variable_names_with_output]
correlation_matrix = numerical_data_heart.corr()
```

```
fig, ax = plt.subplots(figsize=(5, 3))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', ax=ax)
ax.set_xticklabels(correlation_matrix.columns, rotation=45,
horizontalalignment='right')
ax.set_yticklabels(correlation_matrix.columns, rotation=0)

ax.set_title('Correlation matrix')
plt.show()
```



Sabem que una elevada correlació entre variables numèriques independentment del signe és indicativa d'una possible col·linealitat, fet que cal tenir en compte en realitzar el model de regressió logística final. Veiem que hi ha una possible col·linealitat entre les variables **age**-**thalachh** i les variables **thalachh**-**log_oldpeak**.

Així i tot, si observem la correlació obtinguda entre les variables numèriques i la variable de sortida veiem que **log_oldpeak** i **thalachh** són les que estan més correlacionades. En conseqüència, a l'hora de generar el model de regressió logística sols seleccionarem aquestes dues variables numèriques, ja que són les úniques que es poden considerar com a rellevants per explicar la variable a predir. Veiem que s'elimina el possible problema de col·linealitat entre **age**-**thalachh** , però no entre **thalachh**-**log_oldpeak**. Serà necessari comprovar si el problema és tan greu com per haver d'eliminar una de les dues mitjançant el càlcul del VIF .

```
# calculate VIF
collineallity_problem = data_heart[['thalachh', 'log_oldpeak']]
vif = pd.DataFrame()
vif['VIF'] = [variance_inflation_factor(collineallity_problem.values,
i) for i in range(collineallity_problem.shape[1])]
vif['variable'] = collineallity_problem.columns
```

```
print(vif)
```

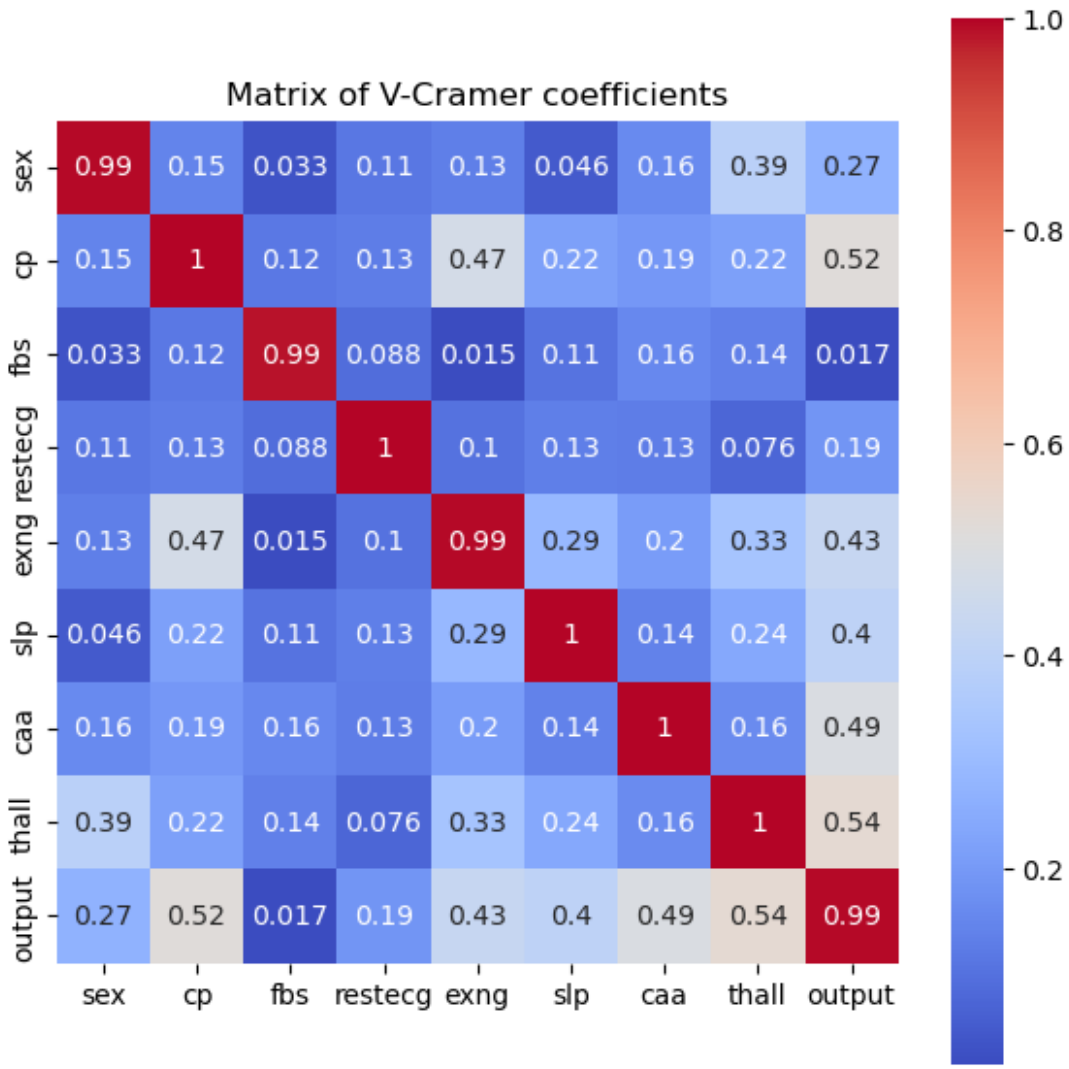
```
        VIF     variable
0  1.894564     thalachh
1  1.894564  log_oldpeak
```

Com podem veure el VIF obtingut es troba dins del rang entre 1 i 5, per tant, es descarta que pugui haver-hi problemes de multicol·linealitat i seleccionem les dues variables numèriques.

Matriu de coeficients V-Cramer de les variables categòriques.

A continuació realitzarem una matriu de coeficients V-Cramer que ens permetrà determinar com es relacionen les diferents variables categòriques entre si i amb la variable **Output**.

```
categorical_data_heart = data_heart[categorical_variable_names]
correlation_matrix = pd.DataFrame(index=categorical_variable_names,
columns=categorical_variable_names)

for var1 in categorical_variable_names:
    for var2 in categorical_variable_names:

        contingency_table = pd.crosstab(data_heart[var1],
data_heart[var2])

        chi2 = chi2_contingency(contingency_table)[0]
        total_values = contingency_table.sum().sum()
        minDim = min(contingency_table.shape)-1
        vcramer = np.sqrt((chi2 / total_values)/minDim)

        correlation_matrix.loc[var1, var2] = vcramer

# Convert the correlation matrix values to float
correlation_matrix = correlation_matrix.astype(float)

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(7, 7))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
square=True)
plt.title('Matrix of V-Cramer coefficients')
plt.show()
```



Podem observar que l'associació entre les diferents variables categòriques explicatives és majoritàriament dèbil, ja que els coeficients V-Cramers obtinguts es troben dins del rang de [0,0.3], encara que es pot apreciar una associació mitjana per les variables **exng**-**thall**, **cp**-**exng** i **sex**-**thall**. Aquesta associació mitjana de primeres no caldria tenir-la en compte en dur a terme el model de regressió, ja que no hauria de generar problemes de col·linealitat que afectessin l'exactitud del model. Així i tot, es comprovarà aquest fet generant dos models de regressió i comparant l'exactitud assolida.

Si ens fixem en els coeficients assolits per les diferents variables categòriques respecte a la variable **Output**, podem veure que només les variables **sex**, **fbs** i **restecg** presenten una associació dèbil. Per tant, aquestes variables seran descartades per a la creació del model de regressió logística.

En definitiva, les variables **cp**, **exng**, **slp**, **ca** i **thall**, ja que són les úniques que es poden considerar com a rellevants per explicar la variable a predir.

Regresió logística.

A continuació durem a terme la generació d'un model que ens permetra predir el output de futurs pacients. Gracies a aquest model es podrà determinar com de bones son les variables seleccionades per predir la variable de sortida i com es relacionen amb aquesta.

```
X = data_heart[['cp','exng', 'slp', 'caa', 'thall','thalachh',
'log_oldpeak']]
y = data_heart[['output']]
y = y.values.ravel()
X = pd.get_dummies(X, drop_first=True)

# generate train and test set
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=48)


# initialize regresion model
logistic_model = LogisticRegression(max_iter = 1000)

logistic_model.fit(X_train, y_train)
y_pred = logistic_model.predict(X_test)

# evaluate the accuracy
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Percentage of accuracy from the logistic regresion model:
{:.2f} %.  \n".format(accuracy*100))

# Obtain the coeficients
coefficients = logistic_model.coef_

coeff_df = pd.DataFrame(coefficients, columns=X.columns)
print("Coefficient of each variable in the regresion logístic model: \
n\n {}".format(coeff_df))
```

```
Percentage of accuracy from the logistic regresion model: 85.71 %.

Coefficient of each variable in the regresion logístic model:

          cp      exng       slp       caa     thall  thalachh
log_oldpeak
0  0.735234 -0.920938  0.214352 -0.586752 -1.050391   0.01778    -
1.029652
```

Podem observar que el model de regressió logística generat per predir la variable **Output** segons totes les variables seleccionades com a rellevants en les anàlisis anteriors presenta una elevada exatitud (85.71%). Per tant, es pot determinar que les variables explicatives seleccionades generen un model valid pera predir el tipus de probabilitat que té el pacient de patir un atac cardíac.

A continuació, com hem dit en la matriu de V-Cramer, comprovarem si en eliminar la variable **exng** i **thall**, millora o no l'exactitud del model generat.

```
X_2 = data_heart[['cp', 'slp', 'caa','thalachh', 'log_oldpeak']]
y_2 = data_heart[['output']]
y_2 = y_2.values.ravel()
X_2 = pd.get_dummies(X_2, drop_first=True)

# generate train and test set
X_train, X_test, y_train, y_test = train_test_split(X_2, y_2,
test_size=0.3, random_state=48)

# initialize regresion model 2
logistic_model_2 = LogisticRegression(max_iter = 1000)

logistic_model_2.fit(X_train, y_train)
y_pred_2 = logistic_model_2.predict(X_test)

# evaluate the accuracy
accuracy = metrics.accuracy_score(y_test, y_pred_2)
print("Percentage of accuracy from the logistic regresion model:
{:.2f} %.  \n".format(accuracy*100))

# Obtain the coeficients
coefficients = logistic_model_2.coef_
coeff_df = pd.DataFrame(coefficients, columns=X_2.columns)
print("Coefficient of each variable in the regresion logístic model: \
n\n {}".format(coeff_df))
```

```
Percentage of accuracy from the logistic regresion model: 82.42 %.

Coefficient of each variable in the regresion logístic model:

          cp       slp       caa  thalachh  log_oldpeak
0  0.871597  0.261667 -0.618158  0.020345    -1.098831
```

Podem observar que el nou model de regressió logística generat per predir la variable **Output** presenta una menor exactitud respecte al model de regressió logística generat anteriorment. En conseqüència, ens quedarem amb el model de regressió logística anterior.

6. Resolució del problema.

Després d'haver realitzat l'estudi sobre el joc de dades, podem afirmar que les variables explicatives **cp**, **exng**, **slp**, **caa**, **thall**, **thalachh** i **log_oldpeak** són els paràmetres clau per determinar si un pacient té menor o major probabilitat de patir un atac de cor.

Podem observar que a mesura que els valors de les variables **cp**, **slp** i **thalachh** augmentin, la probabilitat d'obtenir la categoria "1.Increased probability of heart attack" en

comparació a la categoria "0.Lower probability of heart attack" en la variable **Output** augmentarà.

D'altra banda, a mesura que els valors de les variables **exng**, **thall**, **caa** i **log_oldpeak** augmentin, la probabilitat d'aconseguir la categoria "1.Increased probability of heart attack" en comparació a la categoria "0.Lower probability of heart attack" en la variable **Output** disminuirà.

És a dir, hem trobat que la presència de dolor de pit, una major freqüència cardíaca i un major pendent en l'ona ST augmenten el risc de patir un atac de cor. Per altra banda, un major número de vasos majors, una major depressió en l'ona ST, la presència d'angina induïda per l'exercici i la presencia de talassèmia redueixen aquesta probabilitat.

En conclusió, gràcies a les relacions establertes entre les variables explicatives seleccionades i la variable a predir **Output**, ara tenim un model que ens permet detectar amb una exactitud del 85.71% quins pacients tenen una major probabilitat d'atac de cor. Aquest model permetrà als metges centrar els recursos específics en aquests casos, reduint despeses econòmiques i millorant l'atenció especialitzada al pacient.

7. Taula de contribucions

Cadascun dels membres del grup ha participat en el projecte segons la taula següent.

Contribucions	Signatura
Investigació prèvia	Marina, Jordi
Redacció de les respostes	Marina, Jordi
Desenvolupament del codi	Marina, Jordi
Participació al vídeo	Marina, Jordi