import pandas as pd import numpy as np

df = pd.read\_csv("/content/bank-additional-full.csv", sep=";") df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutco** |
| **0** 56 | housemaid | married | basic.4y | no | no | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **1** 57 | services | married | high.school | unknown | no | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **2** 37 | services | married | high.school | no | yes | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **3** 40 | admin. | married | basic.6y | no | no | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **4** 56 | services | married | high.school | no | no | yes | telephone | may | mon ... 1 | 999 | 0 | nonexiste |

5 rows × 21 columns

df.tail()

## age job marital education default housing loan contact month day\_of\_week ... campaign pdays previous

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **41183** | 73 | retired | married | professional.course | no | yes | no | cellular | nov | fri | ... | 1 | 999 | 0 |
| **41184** | 46 | blue- | married | professional.course | no | no | no | cellular | nov | fri | ... | 1 | 999 | 0 |
|  |  | collar |  |  |  |  |  |  |  |  | | |  |  |
| **41185** | 56 | retired | married | university.degree | no | yes | no | cellular | nov | fri ... 2 | | | 999 | 0 |
| **41186** | 44 | technician | married | professional.course | no | no | no | cellular | nov | fri ... 1 | | | 999 | 0 |
| **41187** | 74 | retired | married | professional.course | no | yes | no | cellular | nov | fri ... 3 | | | 999 | 1 |

5 rows × 21 columns

def replace\_marital(val):

if val=="single":

return 0 else:

return 1

df["marital"]=df["marital"].apply(replace\_marital,1) df.head()

 <ipython-input-3-e3f3028052ce>:6: FutureWarning: the convert\_dtype parameter is deprecated and will be removed in a future version. df["marital"]=df["marital"].apply(replace\_marital,1)

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutco** |
| **0** 56 | housemaid | 1 | basic.4y | no | no | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **1** 57 | services | 1 | high.school | unknown | no | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **2** 37 | services | 1 | high.school | no | yes | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **3** 40 | admin. | 1 | basic.6y | no | no | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **4** 56 | services | 1 | high.school | no | no | yes | telephone | may | mon ... 1 | 999 | 0 | nonexiste |

5 rows × 21 columns

df["housing"]=df["housing"].map({ "no":0,

"yes":1

}.get)

df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutco** |
| **0** 56 | housemaid | 1 | basic.4y | no | 0.0 | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **1** 57 | services | 1 | high.school | unknown | 0.0 | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **2** 37 | services | 1 | high.school | no | 1.0 | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **3** 40 | admin. | 1 | basic.6y | no | 0.0 | no | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **4** 56 | services | 1 | high.school | no | 0.0 | yes | telephone | may | mon ... 1 | 999 | 0 | nonexiste |

5 rows × 21 columns

df["loan"]=df["loan"].replace({ "no":0,

"yes":1

})

df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutco** |
| **0** 56 | housemaid | 1 | basic.4y | no | 0.0 | 0 | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **1** 57 | services | 1 | high.school | unknown | 0.0 | 0 | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **2** 37 | services | 1 | high.school | no | 1.0 | 0 | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **3** 40 | admin. | 1 | basic.6y | no | 0.0 | 0 | telephone | may | mon ... 1 | 999 | 0 | nonexiste |
| **4** 56 | services | 1 | high.school | no | 0.0 | 1 | telephone | may | mon ... 1 | 999 | 0 | nonexiste |

5 rows × 21 columns

df["job"].unique() #to find unique value of column job

 array(['housemaid', 'services', 'admin.', 'blue-collar', 'technician',

'retired', 'management', 'unemployed', 'self-employed', 'unknown', 'entrepreneur', 'student'], dtype=object)

df["job"].replace({ 'unknown':np.nan,

'unemployed':0, 'services':1, 'management':2, 'blue-collar':3, 'self-employed':4, 'technician':5, 'entrepreneur':6,

'admin.':7, 'student':8,

'housemaid':9, 'retired':10

},inplace=True) df.head()

 <ipython-input-7-b2109e05a277>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co

df["job"].replace({

<ipython-input-7-b2109e05a277>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future ve df["job"].replace({

## age job marital education default housing loan contact month day\_of\_week ... campaign pdays previous poutcome em

1. 56 9.0 1 basic.4y no 0.0 0 telephone may mon ... 1 999 0 nonexistent
2. 57 1.0 1 high.school unknown 0.0 0 telephone may mon ... 1 999 0 nonexistent
3. 37 1.0 1 high.school no 1.0 0 telephone may mon ... 1 999 0 nonexistent
4. 40 7.0 1 basic.6y no 0.0 0 telephone may mon ... 1 999 0 nonexistent
5. 56 1.0 1 high.school no 0.0 1 telephone may mon ... 1 999 0 nonexistent 5 rows × 21 columns

df["education"].unique()

 array(['basic.4y', 'high.school', 'basic.6y', 'basic.9y',

'professional.course', 'unknown', 'university.degree', 'illiterate'], dtype=object)

df["education"].replace({

'basic.4y':1, 'high.school':2, 'basic.6y':3, 'basic.9y':4, 'professional.course':5, 'university.degree':6, 'illiterate':0,'unknown':np

} ,inplace=True) df.head()

 <ipython-input-9-e00d0648f3b4>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co

df["education"].replace({

<ipython-input-9-e00d0648f3b4>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future ve df["education"].replace({

## age job marital education default housing loan contact month day\_of\_week ... campaign pdays previous poutcome emp

1. 56 9.0 1 1.0 no 0.0 0 telephone may mon ... 1 999 0 nonexistent
2. 57 1.0 1 2.0 unknown 0.0 0 telephone may mon ... 1 999 0 nonexistent
3. 37 1.0 1 2.0 no 1.0 0 telephone may mon ... 1 999 0 nonexistent
4. 40 7.0 1 3.0 no 0.0 0 telephone may mon ... 1 999 0 nonexistent
5. 56 1.0 1 2.0 no 0.0 1 telephone may mon ... 1 999 0 nonexistent 5 rows × 21 columns

df.contact.replace({"unknown":np.nan, "telephone":0, "cellular":1}, inplace=True)

df.head()

 <ipython-input-10-b74c0a009967>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co

df.contact.replace({"unknown":np.nan, "telephone":0, "cellular":1},

<ipython-input-10-b74c0a009967>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future v df.contact.replace({"unknown":np.nan, "telephone":0, "cellular":1},

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.** |
| **0** 56 | 9.0 | 1 | 1.0 | no | 0.0 | 0 | 0 | may | mon ... 1 | 999 | 0 | nonexistent |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | may | mon ... 1 | 999 | 0 | nonexistent |  |
| **2** 37 | 1.0 | 1 | 2.0 | no | 1.0 | 0 | 0 | may | mon ... 1 | 999 | 0 | nonexistent |  |
| **3** 40 | 7.0 | 1 | 3.0 | no | 0.0 | 0 | 0 | may | mon ... 1 | 999 | 0 | nonexistent |  |
| **4** 56 | 1.0 | 1 | 2.0 | no | 0.0 | 1 | 0 | may | mon ... 1 | 999 | 0 | nonexistent |  |

1. rows × 21 columns

df.contact.unique()  array([0, 1])

df.month.unique()

 array(['may', 'jun', 'jul', 'aug', 'oct', 'nov', 'dec', 'mar', 'apr', 'sep'], dtype=object)

df.month=df.month.map({'oct':10, 'may':5, 'apr':4, 'jun':6, 'feb':2, 'aug':8, 'jan':1, 'jul':7, 'nov':11,

'sep':9, 'mar':3, 'dec':12}) df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.** |
| **0** 56 | 9.0 | 1 | 1.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | nonexistent |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | nonexistent |  |
| **2** 37 | 1.0 | 1 | 2.0 | no | 1.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | nonexistent |  |
| **3** 40 | 7.0 | 1 | 3.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | nonexistent |  |
| **4** 56 | 1.0 | 1 | 2.0 | no | 0.0 | 1 | 0 | 5 | mon ... 1 | 999 | 0 | nonexistent |  |

5 rows × 21 columns

df.poutcome.unique()

array(['nonexistent', 'failure', 'success'], dtype=object)

df.poutcome=df.poutcome.map({'unknown':np.nan, 'failure':0, 'other':1, 'success':2})

df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.v** |
| **0** 56 | 9.0 | 1 | 1.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | NaN |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | NaN |  |
| **2** 37 | 1.0 | 1 | 2.0 | no | 1.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | NaN |  |
| **3** 40 | 7.0 | 1 | 3.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 999 | 0 | NaN |  |
| **4** 56 | 1.0 | 1 | 2.0 | no | 0.0 | 1 | 0 | 5 | mon ... 1 | 999 | 0 | NaN |  |

5 rows × 21 columns

df.pdays=df.pdays.apply(lambda v:(v-df.pdays.min())/(df.pdays.max()- df.pdays.min()))

df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.v** |
| **0** 56 | 9.0 | 1 | 1.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **2** 37 | 1.0 | 1 | 2.0 | no | 1.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **3** 40 | 7.0 | 1 | 3.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **4** 56 | 1.0 | 1 | 2.0 | no | 0.0 | 1 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |

5 rows × 21 columns

df.y.unique()

 array(['no', 'yes'], dtype=object)

df.y.replace({'no':0, 'yes':1}, inplace=True) df.head()

 <ipython-input-26-ce21d4741977>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co

df.y.replace({'no':0, 'yes':1}, inplace=True)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.v** |
| **0** 56 | 9.0 | 1 | 1.0 | no | 0.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **2** 37 | 1.0 | 1 | 2.0 | no | 1.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **3** 40 | 7.0 | 1 | 3.0 | no | 0.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **4** 56 | 1.0 | 1 | 2.0 | no | 0.0 | 1 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |

5 rows × 21 columns

df.duration=df.duration.apply(lambda v:(v-df.duration.min())/(df.duration.max()-df.duration.min())) df.head()



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.v** |
| **0** 56 | 9.0 | 1 | 1.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **2** 37 | 1.0 | 1 | 2.0 | no | 1.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **3** 40 | 7.0 | 1 | 3.0 | no | 0.0 | 0 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |
| **4** 56 | 1.0 | 1 | 2.0 | no | 0.0 | 1 | 0 | 5 | mon ... 1 | 1.0 | 0 | NaN |  |

5 rows × 21 columns

df.day\_of\_week.unique()

 array(['mon', 'tue', 'wed', 'thu', 'fri'], dtype=object)

df.day\_of\_week=df.day\_of\_week.map({'mon':1, 'tue':2, 'wed':3, 'thu':4, 'fri':5}) df.head()

## age job marital education default housing loan contact month day\_of\_week ... campaign pdays previous poutcome emp.v

**0** 56 9.0 1 1.0 no 0.0 0 0 5 1 ... 1 1.0 0 NaN

**1** 57 1.0 1 2.0 unknown 0.0 0 0 5 1 ... 1 1.0 0 NaN

**2** 37 1.0 1 2.0 no 1.0 0 0 5 1 ... 1 1.0 0 NaN

**3** 40 7.0 1 3.0 no 0.0 0 0 5 1 ... 1 1.0 0 NaN

**4** 56 1.0 1 2.0 no 0.0 1 0 5 1 ... 1 1.0 0 NaN

5 rows × 21 columns

df["default"].replace({ "no":0,

"yes":1

},inplace=True) df.head()

 <ipython-input-29-83ce5d7dfc10>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained as The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co

df["default"].replace({

## age job marital education default housing loan contact month day\_of\_week ... campaign pdays previous poutcome emp.v

**0** 56 9.0 1 1.0 0 0.0 0 0 5 1 ... 1 1.0 0 NaN

**1** 57 1.0 1 2.0 unknown 0.0 0 0 5 1 ... 1 1.0 0 NaN

**2** 37 1.0 1 2.0 0 1.0 0 0 5 1 ... 1 1.0 0 NaN

**3** 40 7.0 1 3.0 0 0.0 0 0 5 1 ... 1 1.0 0 NaN

**4** 56 1.0 1 2.0 0 0.0 1 0 5 1 ... 1 1.0 0 NaN

5 rows × 21 columns

df.describe()



|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **age** | **job** | **marital** | **education** | **housing** | **contact** | **month** | **day\_of\_week** | **duration** | **c** |
| **count** | 41188.00000 | 40858.000000 | 41188.000000 | 39457.000000 | 40198.000000 | 41188.000000 | 41188.000000 | 41188.000000 | 41188.000000 | 4118 |
| **mean** | 40.02406 | 4.709188 | 0.719141 | 3.889931 | 0.536743 | 0.634748 | 6.607896 | 2.979581 | 0.052518 |  |
| **std** | 10.42125 | 2.528108 | 0.449424 | 1.826720 | 0.498654 | 0.481507 | 2.040998 | 1.411514 | 0.052720 |  |
| **min** | 17.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 3.000000 | 1.000000 | 0.000000 |  |
| **25%** | 32.00000 | 3.000000 | 0.000000 | 2.000000 | 0.000000 | 0.000000 | 5.000000 | 2.000000 | 0.020740 |  |
| **50%** | 38.00000 | 5.000000 | 1.000000 | 4.000000 | 1.000000 | 1.000000 | 6.000000 | 3.000000 | 0.036600 |  |
| **75%** | 47.00000 | 7.000000 | 1.000000 | 6.000000 | 1.000000 | 1.000000 | 8.000000 | 4.000000 | 0.064864 |  |
| **max** | 98.00000 | 10.000000 | 1.000000 | 6.000000 | 1.000000 | 1.000000 | 12.000000 | 5.000000 | 1.000000 | 5 |

df.shape

 (41188, 21)

df.to\_csv("/content/bank-additional-new.csv",index=False) new\_df=pd.read\_csv("/content/bank-additional-new.csv")

new\_df.head()

# 

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **age** | **job** | **marital** | **education** | **default** | **housing** | **loan** | **contact** | **month** | **day\_of\_week ... campaign** | **pdays** | **previous** | **poutcome** | **emp.v** |
| **0** 56 | 9.0 | 1 | 1.0 | 0 | 0.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **1** 57 | 1.0 | 1 | 2.0 | unknown | 0.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **2** 37 | 1.0 | 1 | 2.0 | 0 | 1.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **3** 40 | 7.0 | 1 | 3.0 | 0 | 0.0 | 0 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |
| **4** 56 | 1.0 | 1 | 2.0 | 0 | 0.0 | 1 | 0 | 5 | 1 ... 1 | 1.0 | 0 | NaN |  |

5 rows × 21 columns

new\_df.corr()



ValueError Traceback (most recent call last)

<ipython-input-33-326e7bbec5b0> in <cell line: 1>()

----> 1 new\_df.corr()

 3 frames

/usr/local/lib/python3.10/dist-packages/pandas/core/internals/managers.py in \_interleave(self, dtype, na\_value) 1751 else:

1752 arr = blk.get\_values(dtype)

-> 1753 result[rl.indexer] = arr

1754 itemmask[rl.indexer] = 1

1755

ValueError: could not convert string to float: 'unknown'

Next steps:

new\_df = pd.get\_dummies(new\_df, drop\_first=True) # This will create dummy/indicator variables for categorical columns.

Explain error

numeric\_df = new\_df.select\_dtypes(include=['float64', 'int64']) numeric\_df.corr()



|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **age** | **job** | **marital** | **education** | **housing** | **contact** | **month** | **day\_of\_week** | **duration** | **campaign** | **pdays** | **pr** |
| **age** | 1.000000 | 0.150802 | 0.411703 | -0.116074 | -0.001636 | -0.007021 | 0.077265 | -0.018486 | -0.000866 | 0.004594 | -0.034369 | 0. |
| **job** | 0.150802 | 1.000000 | -0.072637 | 0.134969 | 0.009771 | 0.085828 | 0.086367 | -0.002940 | 0.000821 | -0.000159 | -0.083395 | 0. |
| **marital** | 0.411703 | -0.072637 | 1.000000 | -0.106687 | -0.014681 | -0.071159 | 0.017394 | -0.010839 | -0.007808 | 0.007624 | 0.042015 | -0. |
| **education** | -0.116074 | 0.134969 | -0.106687 | 1.000000 | 0.020255 | 0.096494 | 0.113639 | 0.007433 | -0.016067 | -0.001964 | -0.028592 | 0. |
| **housing** | -0.001636 | 0.009771 | -0.014681 | 0.020255 | 1.000000 | 0.083022 | 0.032084 | -0.009083 | -0.007806 | -0.011168 | -0.010649 | 0. |
| **contact** | -0.007021 | 0.085828 | -0.071159 | 0.096494 | 0.083022 | 1.000000 | 0.324315 | -0.019583 | 0.026657 | -0.077368 | -0.117970 | 0. |
| **month** | 0.077265 | 0.086367 | 0.017394 | 0.113639 | 0.032084 | 0.324315 | 1.000000 | -0.006959 | -0.019302 | -0.030635 | -0.079556 | 0. |
| **day\_of\_week** | -0.018486 | -0.002940 | -0.010839 | 0.007433 | -0.009083 | -0.019583 | -0.006959 | 1.000000 | 0.010549 | 0.015098 | 0.006765 | 0. |
| **duration** | -0.000866 | 0.000821 | -0.007808 | -0.016067 | -0.007806 | 0.026657 | -0.019302 | 0.010549 | 1.000000 | -0.071699 | -0.047577 | 0. |
| **campaign** | 0.004594 | -0.000159 | 0.007624 | -0.001964 | -0.011168 | -0.077368 | -0.030635 | 0.015098 | -0.071699 | 1.000000 | 0.052584 | -0. |
| **pdays** | -0.034369 | -0.083395 | 0.042015 | -0.028592 | -0.010649 | -0.117970 | -0.079556 | 0.006765 | -0.047577 | 0.052584 | 1.000000 | -0. |
| **previous** | 0.024365 | 0.066687 | -0.048485 | 0.016876 | 0.021656 | 0.212848 | 0.063754 | 0.004013 | 0.020640 | -0.079141 | -0.587514 | 1. |
| **poutcome** | 0.070651 | 0.155432 | -0.052542 | 0.073859 | 0.000916 | -0.007434 | 0.118006 | -0.028250 | 0.130641 | -0.058456 | -0.936492 | 0. |
| **emp.var.rate** | -0.000371 | -0.081441 | 0.099403 | -0.028934 | -0.060917 | -0.393584 | 0.058874 | -0.004401 | -0.027968 | 0.150754 | 0.271004 | -0. |
| **cons.price.idx** | 0.000857 | -0.070022 | 0.063013 | -0.081933 | -0.081396 | -0.591474 | -0.150350 | -0.004586 | 0.005312 | 0.127836 | 0.078889 | -0. |
| **cons.conf.idx** | 0.129372 | 0.108297 | 0.056186 | 0.063389 | -0.034167 | -0.251614 | 0.264227 | -0.000099 | -0.008173 | -0.013733 | -0.091342 | -0. |
| **euribor3m** | 0.010767 | -0.082316 | 0.109479 | -0.019943 | -0.059978 | -0.399773 | 0.163411 | -0.005552 | -0.032897 | 0.135133 | 0.296899 | -0. |
| **nr.employed** | -0.017725 | -0.103960 | 0.102382 | -0.017918 | -0.046455 | -0.269155 | 0.132697 | -0.000734 | -0.044703 | 0.144095 | 0.372605 | -0. |
| **y** | 0.030399 | 0.096900 | -0.054133 | 0.038306 | 0.011662 | 0.144773 | 0.037187 | 0.010051 | 0.405274 | -0.066357 | -0.324914 | 0. |

new\_df.replace('unknown', np.nan, inplace=True) # Replace 'unknown' with NaN new\_df.corr()

# 

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **age** | **job** | **marital** | **education** | **housing** | **contact** | **month** | **day\_of\_week** | **duration** | **campaign ... emp.v** |
| **age** | 1.000000 | 0.150802 | 0.411703 | -0.116074 | -0.001636 | -0.007021 | 0.077265 | -0.018486 | -0.000866 | 0.004594 ... -0 |
| **job** | 0.150802 | 1.000000 | -0.072637 | 0.134969 | 0.009771 | 0.085828 | 0.086367 | -0.002940 | 0.000821 | -0.000159 ... -0 |
| **marital** | 0.411703 | -0.072637 | 1.000000 | -0.106687 | -0.014681 | -0.071159 | 0.017394 | -0.010839 | -0.007808 | 0.007624 ... 0 |
| **education** | -0.116074 | 0.134969 | -0.106687 | 1.000000 | 0.020255 | 0.096494 | 0.113639 | 0.007433 | -0.016067 | -0.001964 ... -0 |
| **housing** | -0.001636 | 0.009771 | -0.014681 | 0.020255 | 1.000000 | 0.083022 | 0.032084 | -0.009083 | -0.007806 | -0.011168 ... -0 |
| **contact** | -0.007021 | 0.085828 | -0.071159 | 0.096494 | 0.083022 | 1.000000 | 0.324315 | -0.019583 | 0.026657 | -0.077368 ... -0 |
| **month** | 0.077265 | 0.086367 | 0.017394 | 0.113639 | 0.032084 | 0.324315 | 1.000000 | -0.006959 | -0.019302 | -0.030635 ... 0 |
| **day\_of\_week** | -0.018486 | -0.002940 | -0.010839 | 0.007433 | -0.009083 | -0.019583 | -0.006959 | 1.000000 | 0.010549 | 0.015098 ... -0 |
| **duration** | -0.000866 | 0.000821 | -0.007808 | -0.016067 | -0.007806 | 0.026657 | -0.019302 | 0.010549 | 1.000000 | -0.071699 ... -0 |
| **campaign** | 0.004594 | -0.000159 | 0.007624 | -0.001964 | -0.011168 | -0.077368 | -0.030635 | 0.015098 | -0.071699 | 1.000000 ... 0 |
| **pdays** | -0.034369 | -0.083395 | 0.042015 | -0.028592 | -0.010649 | -0.117970 | -0.079556 | 0.006765 | -0.047577 | 0.052584 ... 0 |
| **previous** | 0.024365 | 0.066687 | -0.048485 | 0.016876 | 0.021656 | 0.212848 | 0.063754 | 0.004013 | 0.020640 | -0.079141 ... -0 |
| **poutcome** | 0.070651 | 0.155432 | -0.052542 | 0.073859 | 0.000916 | -0.007434 | 0.118006 | -0.028250 | 0.130641 | -0.058456 ... -0 |
| **emp.var.rate** | -0.000371 | -0.081441 | 0.099403 | -0.028934 | -0.060917 | -0.393584 | 0.058874 | -0.004401 | -0.027968 | 0.150754 ... 1 |
| **cons.price.idx** | 0.000857 | -0.070022 | 0.063013 | -0.081933 | -0.081396 | -0.591474 | -0.150350 | -0.004586 | 0.005312 | 0.127836 ... 0 |
| **cons.conf.idx** | 0.129372 | 0.108297 | 0.056186 | 0.063389 | -0.034167 | -0.251614 | 0.264227 | -0.000099 | -0.008173 | -0.013733 ... 0 |
| **euribor3m** | 0.010767 | -0.082316 | 0.109479 | -0.019943 | -0.059978 | -0.399773 | 0.163411 | -0.005552 | -0.032897 | 0.135133 ... 0 |
| **nr.employed** | -0.017725 | -0.103960 | 0.102382 | -0.017918 | -0.046455 | -0.269155 | 0.132697 | -0.000734 | -0.044703 | 0.144095 ... 0 |
| **y** | 0.030399 | 0.096900 | -0.054133 | 0.038306 | 0.011662 | 0.144773 | 0.037187 | 0.010051 | 0.405274 | -0.066357 ... -0 |
| **default\_1** | 0.001891 | -0.004664 | 0.005334 | 0.000525 | -0.003524 | 0.006474 | 0.010003 | -0.005923 | -0.005101 | -0.003803 ... 0 |
| **default\_unknown** | 0.165001 | -0.094225 | 0.123565 | -0.158554 | -0.015793 | -0.135604 | -0.084801 | -0.004040 | -0.011588 | 0.033007 ... 0 |
| **loan\_1** | -0.007198 | 0.007436 | -0.004999 | 0.008268 | 0.046462 | 0.013367 | -0.001696 | 0.001850 | 0.000121 | 0.005294 ... |
| **loan\_unknown** | -0.001092 | -0.005715 | -0.000688 | -0.006104 | NaN | -0.022189 | -0.011869 | 0.002607 | -0.004897 | -0.000396 ... 0 |

23 rows × 23 columns

import matplotlib.pyplot as plt

%matplotlib inline

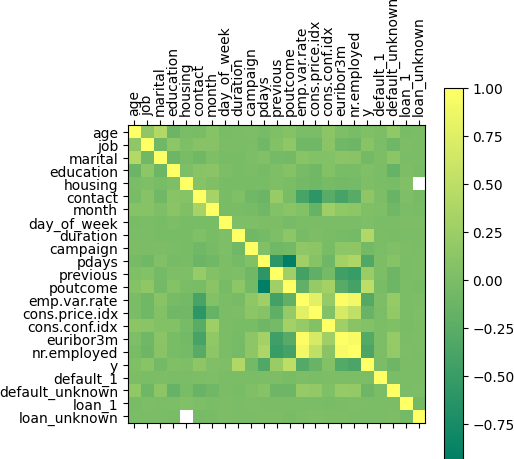
plt.matshow(new\_df.corr(), cmap='summer') plt.colorbar()

plt.xticks(list(range(len(new\_df.columns))), new\_df.columns, rotation='vertical')

plt.yticks(list(range(len(new\_df.columns))), new\_df.columns, rotation='horizontal')

plt.show()





new\_df.corr()["y"].sort\_values(ascending=False)

# 

|  |  |
| --- | --- |
|  | **y** |
| **y** | 1.000000 |
| **poutcome** | 0.494368 |
| **duration** | 0.405274 |
| **previous** | 0.230181 |
| **contact** | 0.144773 |
| **job** | 0.096900 |
| **cons.conf.idx** | 0.054878 |
| **education** | 0.038306 |
| **month** | 0.037187 |
| **age** | 0.030399 |
| **housing** | 0.011662 |
| **day\_of\_week** | 0.010051 |
| **loan\_unknown** | -0.002270 |
| **default\_1** | -0.003041 |
| **loan\_1** | -0.004466 |
| **marital** | -0.054133 |
| **campaign** | -0.066357 |
| **default\_unknown** | -0.099293 |
| **cons.price.idx** | -0.136211 |
| **emp.var.rate** | -0.298334 |
| **euribor3m** | -0.307771 |
| **pdays** | -0.324914 |
| **nr.employed** | -0.354678 |

new\_df.plot.box()

plt.xticks(list(range(len(new\_df.columns))), new\_df.columns, rotation='vertical')



<matplotlib.axis.XTick at 0x79e76aecd600>,

<matplotlib.axis.XTick at 0x79e76af7f490>,

<matplotlib.axis.XTick at 0x79e76af04ee0>,

<matplotlib.axis.XTick at 0x79e76af05990>,

<matplotlib.axis.XTick at 0x79e76af05150>,

<matplotlib.axis.XTick at 0x79e76b0455d0>,

<matplotlib.axis.XTick at 0x79e76b046da0>,

<matplotlib.axis.XTick at 0x79e76b047b50>,

<matplotlib.axis.XTick at 0x79e76af050f0>,

<matplotlib.axis.XTick at 0x79e76b0462f0>,

<matplotlib.axis.XTick at 0x79e76b0478e0>,

<matplotlib.axis.XTick at 0x79e76aed83a0>,

<matplotlib.axis.XTick at 0x79e76aeda2f0>,

<matplotlib.axis.XTick at 0x79e76b044850>,

<matplotlib.axis.XTick at 0x79e76aed9960>,

<matplotlib.axis.XTick at 0x79e76aedb8e0>,

<matplotlib.axis.XTick at 0x79e76aedad70>,

<matplotlib.axis.XTick at 0x79e76aed8f40>,

<matplotlib.axis.XTick at 0x79e76af7e110>,

<matplotlib.axis.XTick at 0x79e76af7fac0>,

<matplotlib.axis.XTick at 0x79e76af7d990>,

<matplotlib.axis.XTick at 0x79e76af7d390>], [Text(0, 0, 'age'),

Text(1, 0, 'job'),

Text(2, 0, 'marital'),

Text(3, 0, 'education'),

Text(4, 0, 'housing'),

Text(5, 0, 'contact'),

Text(6, 0, 'month'),

Text(7, 0, 'day\_of\_week'),

Text(8, 0, 'duration'),

Text(9, 0, 'campaign'),

Text(10, 0, 'pdays'),

Text(11, 0, 'previous'),

Text(12, 0, 'poutcome'),

Text(13, 0, 'emp.var.rate'),

Text(14, 0, 'cons.price.idx'),

Text(15, 0, 'cons.conf.idx'),

Text(16, 0, 'euribor3m'),

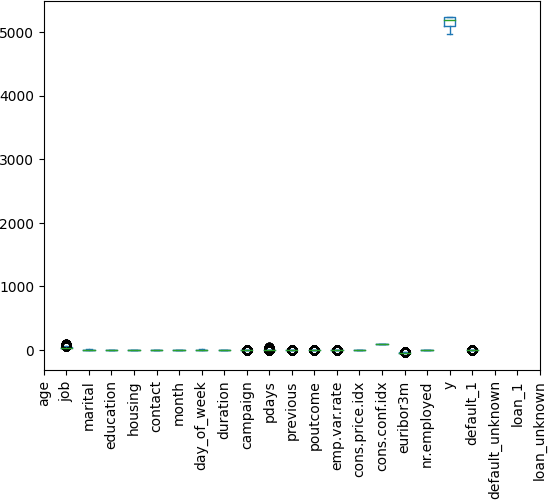
Text(17, 0, 'nr.employed'),

Text(18, 0, 'y'),

Text(19, 0, 'default\_1'),

Text(20, 0, 'default\_unknown'),

Text(21, 0, 'loan\_1'),

Text(22, 0, 'loan\_unknown')])

