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

dfx=pd.read_csv('/content/df_X.csv').iloc[:,1:]
dfx.head()
```


```
{"summary":"{\n  \"name\": \"dfx\",\n  \"rows\": 45211,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 18,\n
\"max\": 95,\n        \"num_unique_values\": 77,\n        \"samples\":
[\n          35,\n          34,\n          53\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 11,\n
\"samples\": [\n          \"admin.\",\n          \"management\",\n
\"housemaid\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"married\",\n          \"single\",\n
\"divorced\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"education\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"tertiary\",\n          \"secondary\",\n
\"primary\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"balance\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3044,\n
\"min\": -8019,\n        \"max\": 102127,\n
\"num_unique_values\": 7168,\n        \"samples\": [\n          3276,\
n          43\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"housing\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"no\",\n          \"yes\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"loan\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"contact\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"telephone\",\n
\"cellular\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"day_of_week\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 8,\n        \"min\": 1,\n
\"max\": 31,\n        \"num_unique_values\": 31,\n        \"samples\":
[\n          1,\n          27\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"month\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 12,\n
\"samples\": [\n          \"apr\",\n          \"mar\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"duration\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 257,\n
\"min\": 0,\n        \"max\": 4918,\n        \"num_unique_values\":
1573,\n        \"samples\": [\n          835,\n          1135\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"campaign\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
3,\n        \"min\": 1,\n        \"max\": 63,\n
\"num_unique_values\": 48,\n        \"samples\": [\n          41,\n
27\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"pdays\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 100,\n        \"min\": -1,\n        \"max\": 871,\n
\"num_unique_values\": 559,\n        \"samples\": [\n          249,\n
551\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"previous\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2,\n        \"min\": 0,\n
\"max\": 275,\n        \"num_unique_values\": 41,\n
\"samples\": [\n          17,\n          9\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"poutcome\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"failure\",\n
\"other\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe","variable_name":"dfx"}
```


```
dfy=pd.read_csv('/content/df_y.csv').iloc[:,1:]
dfy.head()
```


```
{"summary":"{\n  \"name\": \"dfy\",\n  \"rows\": 45211,\n  \"fields\":
[\n    {\n      \"column\": \"y\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"yes\",\n          \"no\"\n        ],\n
```

```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"dfy"}
```

Show specs:

```
dfx.shape
dfx.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   age          45211 non-null  int64
 1   job          44923 non-null  object
 2   marital      45211 non-null  object
 3   education    43354 non-null  object
 4   default      45211 non-null  object
 5   balance      45211 non-null  int64
 6   housing      45211 non-null  object
 7   loan         45211 non-null  object
 8   contact      32191 non-null  object
 9   day_of_week  45211 non-null  int64
 10  month        45211 non-null  object
 11  duration     45211 non-null  int64
 12  campaign     45211 non-null  int64
 13  pdays        45211 non-null  int64
 14  previous     45211 non-null  int64
 15  poutcome     8252 non-null   object
dtypes: int64(7), object(9)
memory usage: 5.5+ MB

dfy.shape
dfy.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 1 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   y       45211 non-null  object
dtypes: object(1)
memory usage: 353.3+ KB

df=pd.concat([dfx,dfy],axis=1)
df.rename(columns = {'y':'term deposit?'}, inplace = True)
df.head(3)

{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 45211,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
```

```
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 18,\n
\"max\": 95,\n        \"num_unique_values\": 77,\n        \"samples\":
[\n          35,\n          34,\n          53\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 11,\n
\"samples\": [\n          \"admin.\",\n          \"management\",\n
\"housemaid\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"married\",\n          \"single\",\n
\"divorced\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"education\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"tertiary\",\n          \"secondary\",\n
\"primary\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"balance\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3044,\n
\"min\": -8019,\n        \"max\": 102127,\n
\"num_unique_values\": 7168,\n        \"samples\": [\n          3276,\
n          43\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"housing\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"no\",\n          \"yes\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"loan\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"contact\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"telephone\",\n
\"cellular\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"day_of_week\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 8,\n        \"min\": 1,\n
\"max\": 31,\n        \"num_unique_values\": 31,\n        \"samples\":
[\n          1,\n          27\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"month\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 12,\n
```

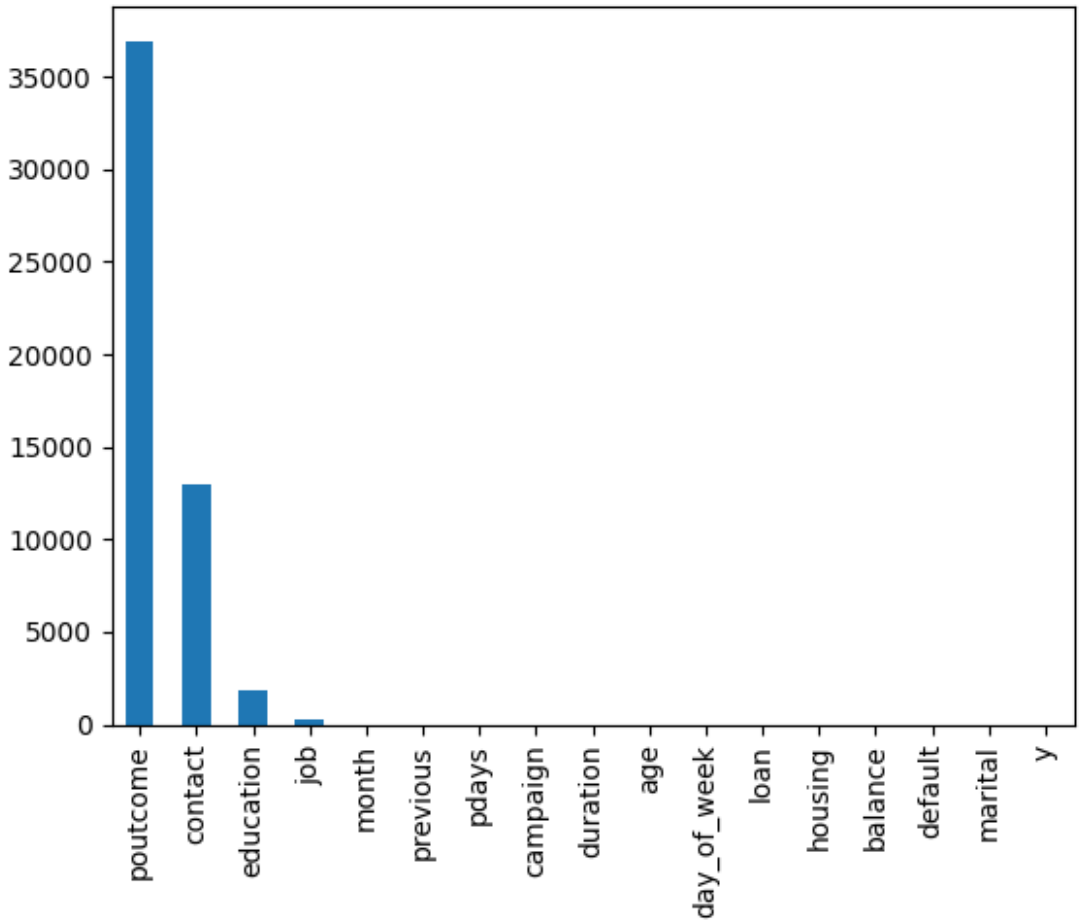
\"samples\": [\n   \"apr\",\n   \"mar\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"duration\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\": 257,\n
\"min\": 0,\n   \"max\": 4918,\n   \"num_unique_values\":
1573,\n   \"samples\": [\n   835,\n   1135\n
],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n
}\n   },\n   {\n   \"column\": \"campaign\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
3,\n   \"min\": 1,\n   \"max\": 63,\n
\"num_unique_values\": 48,\n   \"samples\": [\n   41,\n
27\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"pdays\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 100,\n   \"min\": -1,\n   \"max\": 871,\n
\"num_unique_values\": 559,\n   \"samples\": [\n   249,\n
551\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"previous\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 2,\n   \"min\": 0,\n
\"max\": 275,\n   \"num_unique_values\": 41,\n
\"samples\": [\n   17,\n   9\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"poutcome\",\n   \"properties\":
{\n   \"dtype\": \"category\",\n   \"num_unique_values\":
3,\n   \"samples\": [\n   \"failure\",\n
\"other\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"term deposit?\",\n   \"properties\": {\n   \"dtype\":
\"category\",\n   \"num_unique_values\": 2,\n   \"samples\":
[\n   \"yes\",\n   \"no\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.isnull().sum().sort_values(ascending=False).head()#.sum()
```

```
poutcome     36959
contact      13020
education     1857
job            288
month            0
dtype: int64
```

```
df.isnull().sum().sort_values(ascending=False).plot(kind='bar')
```

```
<Axes: >
```



Dropping:

```
df1=df.copy()
df1.drop(['poutcome', 'contact'],axis=1,inplace=True)
df1.head()
```


```
{"summary":"{\n  \"name\": \"df1\",\n  \"rows\": 45211,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 18,\n
\"max\": 95,\n        \"num_unique_values\": 77,\n        \"samples\":
[\n          35,\n          34,\n          53\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 11,\n
\"samples\": [\n          \"admin.\",\n          \"management\",\n
\"housemaid\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"marital\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"married\",\n          \"single\",\n
\"divorced\"\n        ],\n        \"semantic_type\": \"\",\n
```

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"education\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"tertiary\",\n          \"secondary\",\n
\"primary\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"balance\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3044,\n
\"min\": -8019,\n        \"max\": 102127,\n
\"num_unique_values\": 7168,\n        \"samples\": [\n          3276,\
n          43\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"housing\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"no\",\n          \"yes\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"loan\",\n      \"properties\": {\n
\"dtype\": \"category\",\n        \"num_unique_values\": 2,\n
\"samples\": [\n          \"yes\",\n          \"no\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"day_of_week\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
8,\n        \"min\": 1,\n        \"max\": 31,\n
\"num_unique_values\": 31,\n        \"samples\": [\n          1,\n
27\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"month\",\n      \"properties\": {\n        \"dtype\": \"category\",\
n        \"num_unique_values\": 12,\n        \"samples\": [\n
\"apr\",\n          \"mar\"\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"duration\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 257,\n        \"min\": 0,\n
\"max\": 4918,\n        \"num_unique_values\": 1573,\n
\"samples\": [\n          835,\n          1135\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"campaign\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3,\n
\"min\": 1,\n        \"max\": 63,\n        \"num_unique_values\": 48,\
n        \"samples\": [\n          41,\n          27\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"pdays\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 100,\n
\"min\": -1,\n        \"max\": 871,\n        \"num_unique_values\":
559,\n        \"samples\": [\n          249,\n          551\
n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"previous\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 2,\n   \"min\": 0,\n
\"max\": 275,\n   \"num_unique_values\": 41,\n
\"samples\": [\n   17,\n   9\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"term deposit?\",\n
\"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n   \"samples\": [\n   \"yes\",\
n   \"no\"\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"df1"}

```
df1.shape #ok, dropped
```

```
(45211, 15)
```

```
df1.corr(numeric_only=True)
```

{"summary":"{\n \"name\": \"df1\",\n \"rows\": 7,\n \"fields\": [\n
{\n   \"column\": \"age\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.37590459066917287,\n
\"min\": -0.023758014111728242,\n   \"max\": 1.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n   1.0,\n
0.09778273937134807,\n   -0.023758014111728242\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"balance\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
0.3715588608778324,\n   \"min\": -0.014578278850766218,\n
\"max\": 1.0,\n   \"num_unique_values\": 7,\n   \"samples\":
[\n   0.09778273937134807,\n   1.0,\n
0.003435321868106611\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"day_of_week\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.3874561457888938,\n   \"min\": -
0.09304407377294048,\n   \"max\": 1.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n   -
0.009120045633552305,\n   0.004502585129718555,\n   -
0.09304407377294048\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"duration\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.3856561009484518,\n   \"min\": -
0.08456950273006174,\n   \"max\": 1.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n   -
0.004648428470615691,\n   0.021560380494668893,\n   -
0.0015647704813434864\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"campaign\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.390450994307172,\n   \"min\": -
0.08862766791568427,\n   \"max\": 1.0,\n

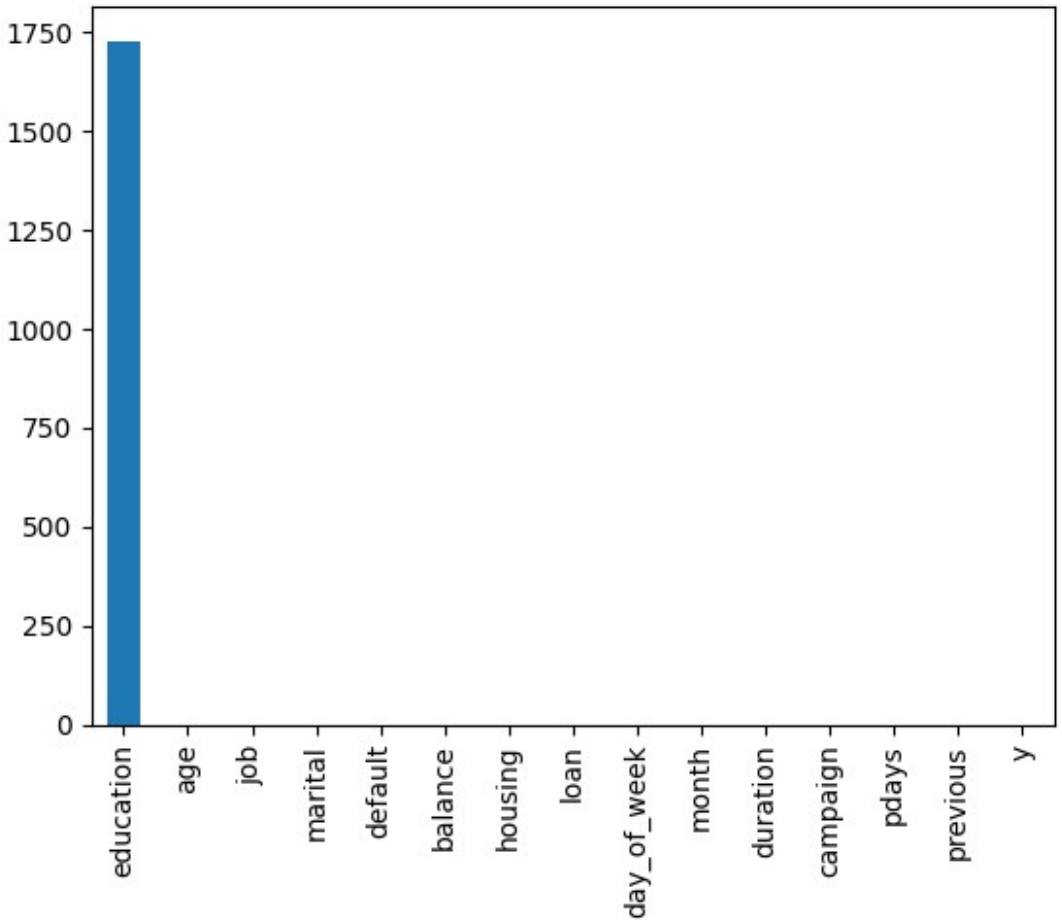
\"num_unique_values\": 7,\n   \"samples\": [\n
0.0047603117624537014,\n   -0.014578278850766218,\n   -
0.08862766791568427\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"pdays\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 0.4083068679638351,\n   \"min\": -0.09304407377294048,\n
\"max\": 1.0,\n   \"num_unique_values\": 7,\n   \"samples\":
[\n   -0.023758014111728242,\n   0.003435321868106611,\n
1.0\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"previous\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 0.3947809359853443,\n   \"min\": -
0.05171049673580366,\n   \"max\": 1.0,\n
\"num_unique_values\": 7,\n   \"samples\": [\n
0.0012883192026692438,\n   0.016673636728357952,\n
0.4548196354805043\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
df['job'].unique()
df1.dropna(subset=['job'],axis=0,inplace=True)
df1.reset_index(drop=True,inplace=True)
df1['job'].isna().sum()
```

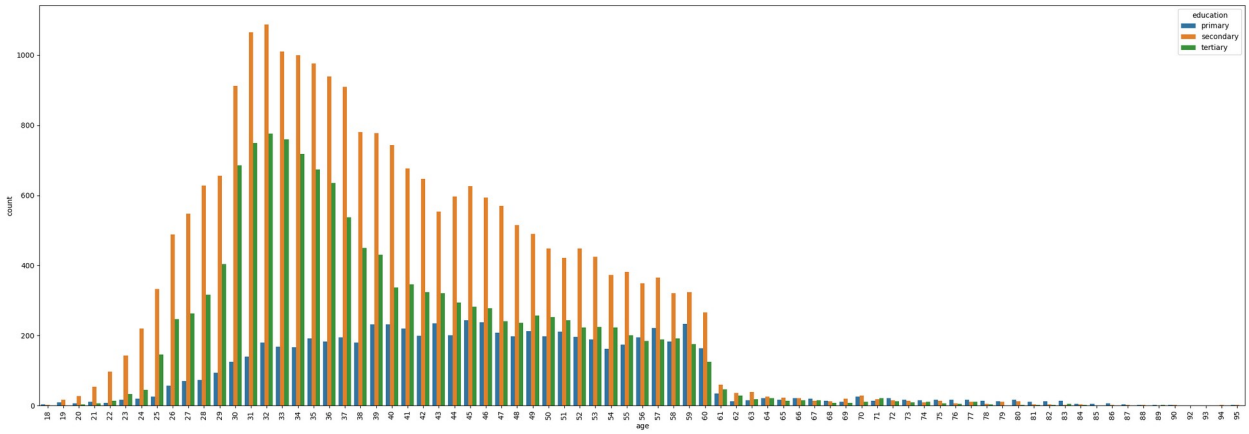
```
0
```

```
df1.isnull().sum().sort_values(ascending=False).plot(kind='bar')
```

```
<Axes: >
```




```
plt.figure(figsize=(30, 10))
sns.countplot(data=df, hue=df['education'], x=df['age'])
plt.xticks(rotation=90)
plt.show()
```



```
df1.dropna(subset=['education'],axis=0,inplace=True)
df1.reset_index(drop=True,inplace=True)
df1['education'].isna().sum()
```

```
0
```

```
df1.duplicated().sum()
# df.drop_duplicates()
# df.reset_index(drop=True)
```

```
0
```

Visualization:

```
count_col=list(df1.nunique().sort_values().index[:8])
count_col
```

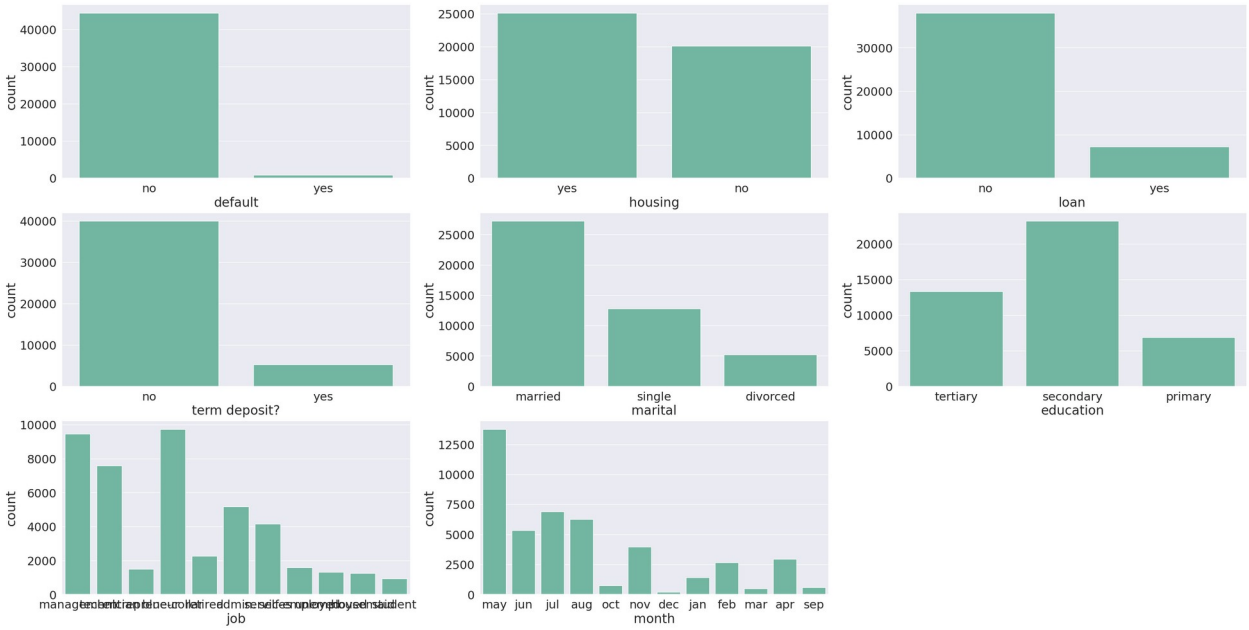
```
['default',
 'housing',
 'loan',
 'term deposit?',
 'marital',
 'education',
 'job',
 'month']
```

```
plt.figure(figsize=(40, 20))

for n, k in enumerate(count_col):
  plt.subplot(3,3,n+1)
  sns.countplot(data=df1,x=df1[k])

sns.set_theme(style='darkgrid',palette='Set2',font_scale=2)
plt.xticks(rotation=90)
plt.show()
```

```
<ipython-input-124-b8d569745506>:4: MatplotlibDeprecationWarning:
Auto-removal of overlapping axes is deprecated since 3.6 and will be
removed two minor releases later; explicitly call ax.remove() as
needed.
  plt.subplot(3,3,n+1)
```

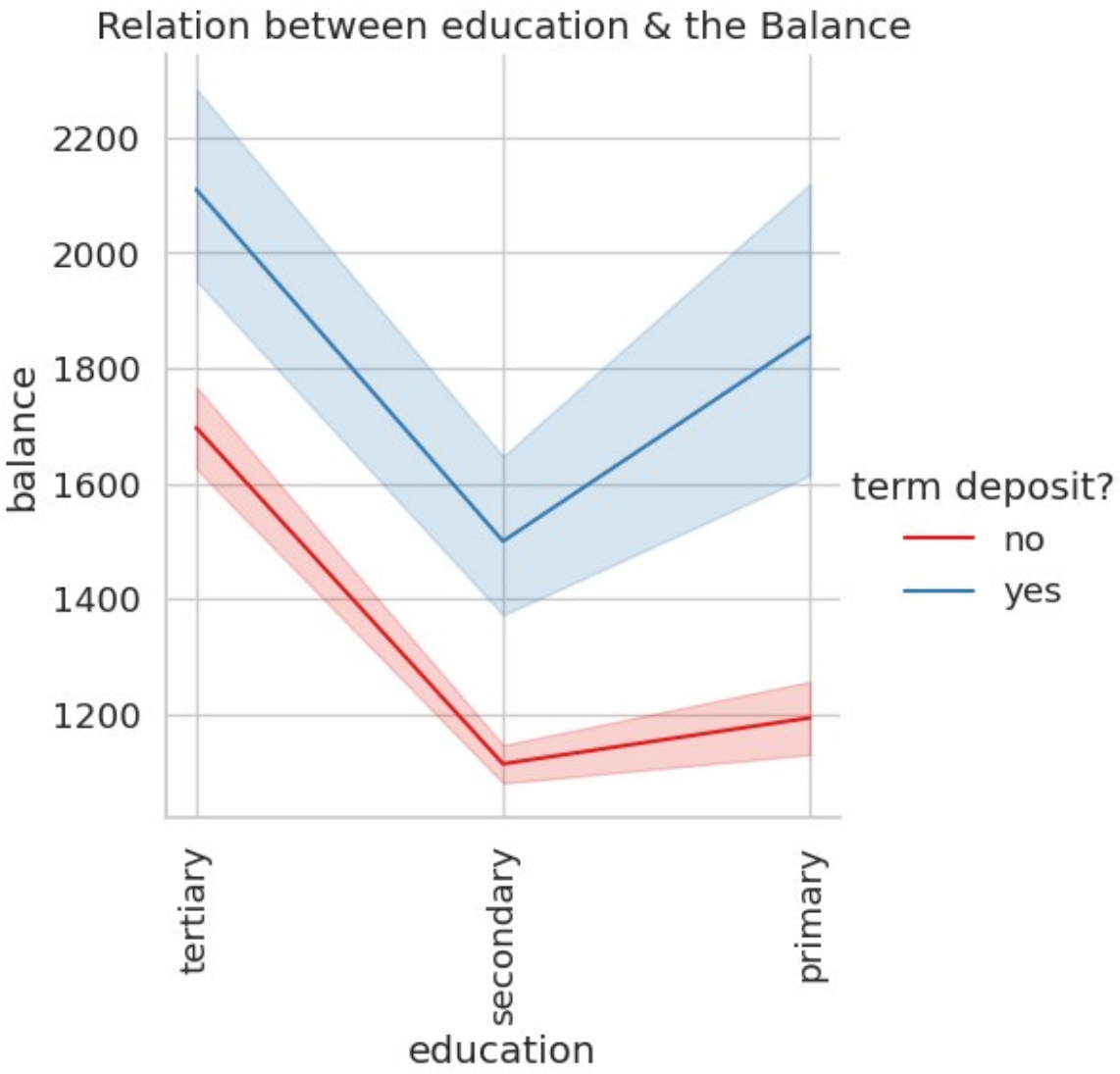



```
plt.figure(figsize=(10, 6))
sns.set_theme(style='whitegrid',palette='Set1',font_scale=1.2)
sns.relplot(data=df1, y=df1['balance'], x=df1['job'],
kind='line',hue=df1['term deposit?'])
plt.xticks(rotation=90)
plt.title("Relation between job & the Balance")
plt.show()
```

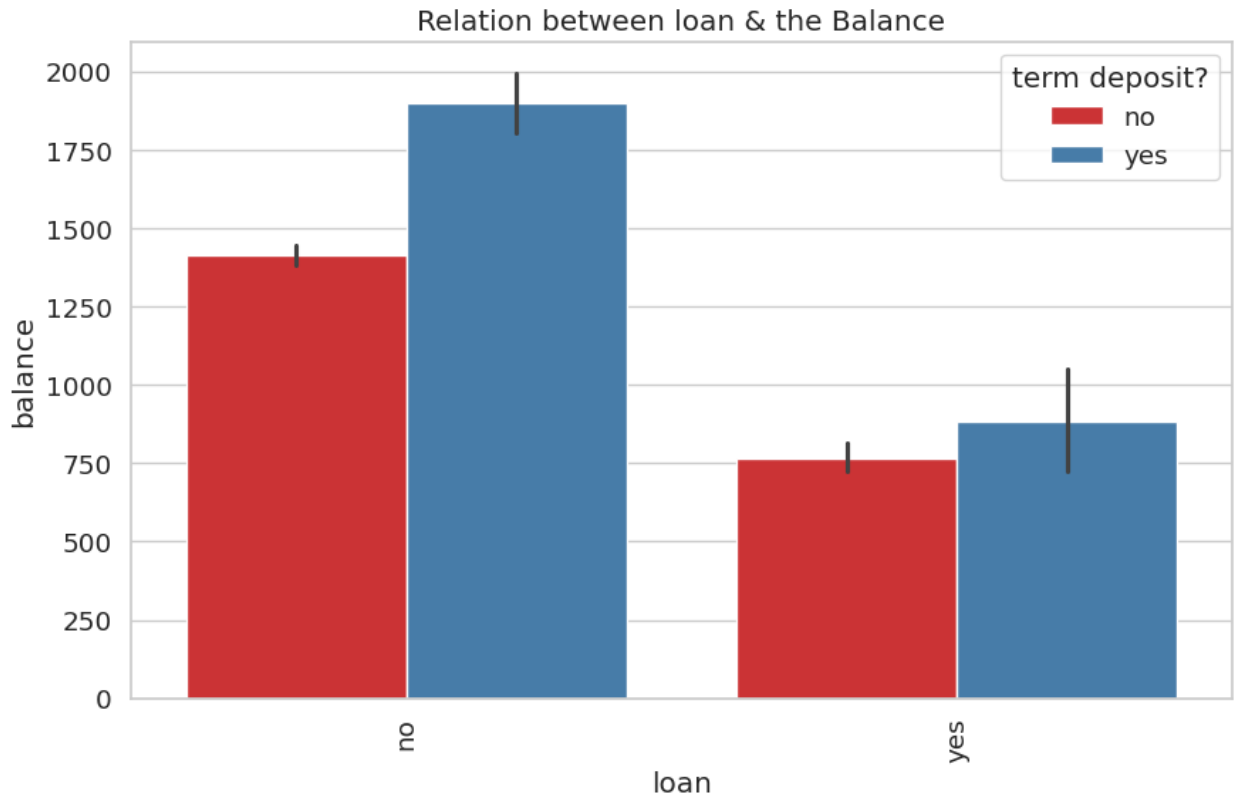
```
<Figure size 1000x600 with 0 Axes>
```


```
plt.figure(figsize=(10, 6))
sns.set_theme(style='whitegrid',palette='Set1',font_scale=1.2)
sns.relplot(data=df1, y=df1['balance'], x=df1['education'],
kind='line',hue=df1['term deposit?'])
plt.title("Relation between education & the Balance")
plt.xticks(rotation=90)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```




```
plt.figure(figsize=(10, 6))
sns.set_theme(style='whitegrid',palette='Set1',font_scale=1.2)
sns.barplot(data=df1, y=df1['balance'], x=df1['loan'],hue=df1['term
deposit?'])
plt.title("Relation between loan & the Balance")
plt.xticks(rotation=90)
plt.show()
```




```
plt.figure(figsize=(10, 6))
sns.set_theme(style='whitegrid',palette='Set2',font_scale=1.2)
sns.barplot(data=df1, y=df1['balance'], x=df1['housing'],hue=df1['term
deposit?'])
plt.title("Relation between housing & the Balance")
plt.xticks(rotation=90)
plt.show()
```


```
n_cols=df1.select_dtypes('object')
n_cols.nunique()
```

```
job              11
marital           3
education         3
default           2
housing           2
loan              2
month            12
term deposit?     2
dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder

for c in
['job','marital','education','default','housing','loan','month',
           'term deposit?']:
  lb= LabelEncoder()
  df1[c]=lb.fit_transform(df1[c])

df1.head(2)
```


```
{"summary":"{\n  \"name\": \"df1\",\n  \"rows\": 45211,\n  \"fields\":
[\n    {\n      \"column\": \"age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 10,\n        \"min\": 18,\n
```

```
\"max\": 95,\n        \"num_unique_values\": 77,\n        \"samples\":
[\n          35,\n          34,\n          53\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"job\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 3,\n        \"min\": 0,\n
\"max\": 11,\n        \"num_unique_values\": 12,\n        \"samples\":
[\n          3,\n          10,\n          4\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"marital\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 2,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1,\n          2,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"education\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1,\n
0,\n          2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"balance\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 3044,\n        \"min\": -8019,\n        \"max\":
102127,\n        \"num_unique_values\": 7168,\n        \"samples\": [\
n          3276,\n          43\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"housing\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"loan\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"day_of_week\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8,\n        \"min\": 1,\n
\"max\": 31,\n        \"num_unique_values\": 31,\n        \"samples\":
[\n          1,\n          27\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"month\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3,\n        \"min\": 0,\n
\"max\": 11,\n        \"num_unique_values\": 12,\n        \"samples\":
[\n          0,\n          7\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
```

\"column\": \"duration\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 257,\n   \"min\": 0,\n   \"max\": 4918,\n   \"num_unique_values\": 1573,\n   \"samples\": [\n   835,\n   1135\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"campaign\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 3,\n   \"min\": 1,\n   \"max\": 63,\n   \"num_unique_values\": 48,\n   \"samples\": [\n   41,\n   27\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"pdays\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 100,\n   \"min\": -1,\n   \"max\": 871,\n   \"num_unique_values\": 559,\n   \"samples\": [\n   249,\n   551\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"previous\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 2,\n   \"min\": 0,\n   \"max\": 275,\n   \"num_unique_values\": 41,\n   \"samples\": [\n   17,\n   9\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"term deposit?\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n   \"min\": 0,\n   \"max\": 1,\n   \"num_unique_values\": 2,\n   \"samples\": [\n   1,\n   0\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"df1"}

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

mscaler = MinMaxScaler(feature_range=(0,1))
y=df1['term deposit?']
x=df1.iloc[:,:-1]
x.head()
```

{"summary":"{\n \"name\": \"x\",\n \"rows\": 45211,\n \"fields\": [\n   {\n   \"column\": \"age\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 10,\n   \"min\": 18,\n   \"max\": 95,\n   \"num_unique_values\": 77,\n   \"samples\": [\n   35,\n   34,\n   53\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"job\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 3,\n   \"min\": 0,\n   \"max\": 11,\n   \"num_unique_values\": 12,\n   \"samples\": [\n   3,\n   10,\n   4\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"marital\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0,\n

```
\"min\": 0,\n        \"max\": 2,\n        \"num_unique_values\": 3,\n
\"samples\": [\n          1,\n          2,\n          0\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"education\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 3,\n
\"num_unique_values\": 4,\n        \"samples\": [\n          1,\n
0,\n          2\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"default\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 0,\n        \"min\": 0,\n        \"max\": 1,\n
\"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"balance\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 3044,\n        \"min\": -8019,\n        \"max\":
102127,\n        \"num_unique_values\": 7168,\n        \"samples\": [\
n          3276,\n          43\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"housing\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          0,\n          1\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"loan\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0,\n        \"min\": 0,\n
\"max\": 1,\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"day_of_week\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8,\n        \"min\": 1,\n
\"max\": 31,\n        \"num_unique_values\": 31,\n        \"samples\":
[\n          1,\n          27\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"month\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3,\n        \"min\": 0,\n
\"max\": 11,\n        \"num_unique_values\": 12,\n        \"samples\":
[\n          0,\n          7\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"duration\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 257,\n        \"min\": 0,\n
\"max\": 4918,\n        \"num_unique_values\": 1573,\n
\"samples\": [\n          835,\n          1135\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"campaign\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 3,\n
\"min\": 1,\n        \"max\": 63,\n        \"num_unique_values\": 48,\
n        \"samples\": [\n          41,\n          27\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"pdays\",\n      \"properties\": {\
n        \"dtype\": \"number\",\n        \"std\": 100,\n
\"min\": -1,\n        \"max\": 871,\n        \"num_unique_values\":
559,\n        \"samples\": [\n          249,\n          551\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"previous\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 2,\n        \"min\": 0,\n
\"max\": 275,\n        \"num_unique_values\": 41,\n
\"samples\": [\n          17,\n          9\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"x"}
```


```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
x,y , random_state=104,test_size=0.25, shuffle=True)

from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

mscaler.fit(X_train)
X_train =mscaler.transform(X_train)
X_test = mscaler.transform(X_test)

model = LinearRegression().fit(X_train,y_train)
y_pred = model.predict(X_test)
print(y_pred)
print(mean_squared_error(y_test,y_pred))
```

```
[0.06259213 0.04673475 0.50014908 ... 0.38824342 0.78825445
0.0487289 ]
0.08328111327660001
```