customer_segmentation

September 14, 2024

0.1 Citation Request

- This dataset is publicly available for research. The details are described in [Moro et al., 2014].
- Please include this citation if you plan to use this database:
  - [**Moro et al., 2014**] S. Moro, P. Cortez, and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, In press. `http://dx.doi.org/10.1016/j.dss.2014.03.001`
- Available at:
  - PDF
  - BibTeX

0.2 Metadata

1. **Title**: Bank Marketing (with social/economic context)
2. **Sources**:
   - Created by: Sérgio Moro (ISCTE-IUL), Paulo Cortez (Univ. Minho), and Paulo Rita (ISCTE-IUL) @ 2014
3. **Past Usage**:
   - The full dataset (bank-additional-full.csv) was described and analyzed in:
     - S. Moro, P. Cortez, and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems (2014). doi:10.1016/j.dss.2014.03.001
4. **Relevant Information**:
   - This dataset is based on the "Bank Marketing" UCI dataset (please check the description at: UCI Bank Marketing Dataset).
   - The data is enriched by the addition of five new social and economic features/attributes (national wide indicators from a ~10M population country), published by the Banco de Portugal and publicly available at: Banco de Portugal Statistics.
   - This dataset is almost identical to the one used in [Moro et al., 2014] (it does not include all attributes due to privacy concerns).
   - Using the rminer package and R tool (rminer package), we found that the addition of the five new social and economic attributes (made available here) leads to substantial improvement in the prediction of success, even when the duration of the call is not included. Note: the file can be read in R using: `d=read.table("bank-additional-full.csv", header=TRUE, sep=";")`.
   - The zip file includes two datasets:
     1. `bank-additional-full.csv` with all examples, ordered by date (from May 2008 to November 2010).

2. `bank-additional.csv` with 10% of the examples (4119), randomly selected from `bank-additional-full.csv`.

- The smallest dataset is provided to test more computationally demanding machine learning algorithms (e.g., SVM).
- The binary classification goal is to predict if the client will subscribe to a bank term deposit (variable `y`).

5. **Number of Instances**: 41,188 for `bank-additional-full.csv`
6. **Number of Attributes**: 20 + output attribute.
7. **Attribute Information**:
   - For more information, read [Moro et al., 2014].
   - Input variables:
     - **bank client data**:
       1. `age` (numeric)
       2. `job`: type of job (categorical: "admin.", "blue-collar", "entrepreneur", "housemaid", "management", "retired", "self-employed", "services", "student", "technician", "unemployed", "unknown")
       3. `marital`: marital status (categorical: "divorced", "married", "single", "unknown"; note: "divorced" means divorced or widowed)
       4. `education`: education level (categorical: "basic.4y", "basic.6y", "basic.9y", "high.school", "illiterate", "professional.course", "university.degree", "unknown")
       5. `default`: has credit in default? (categorical: "no", "yes", "unknown")
       6. `housing`: has housing loan? (categorical: "no", "yes", "unknown")
       7. `loan`: has personal loan? (categorical: "no", "yes", "unknown")
     - **related with the last contact of the current campaign**:
       1. `contact`: contact communication type (categorical: "cellular", "telephone")
       2. `month`: last contact month of year (categorical: "jan", "feb", "mar", ..., "nov", "dec")
       3. `day_of_week`: last contact day of the week (categorical: "mon", "tue", "wed", "thu", "fri")
       4. `duration`: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then `y="no"`). Yet, the duration is not known before a call is performed. Also, after the end of the call `y` is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
     - **other attributes**:
       1. `campaign`: number of contacts performed during this campaign and for this client (numeric, includes last contact)
       2. `pdays`: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
       3. `previous`: number of contacts performed before this campaign and for this client (numeric)
       4. `poutcome`: outcome of the previous marketing campaign (categorical: "failure", "nonexistent", "success")
     - **social and economic context attributes**:
       1. `emp.var.rate`: employment variation rate - quarterly indicator (numeric)
       2. `cons.price.idx`: consumer price index - monthly indicator (numeric)

3. `cons.conf.idx`: consumer confidence index - monthly indicator (numeric)
4. `euribor3m`: euribor 3 month rate - daily indicator (numeric)
5. `nr.employed`: number of employees - quarterly indicator (numeric)

8. **Output Variable (Desired Target)**:
   - y - has the client subscribed to a term deposit? (binary: "yes", "no")
9. **Missing Attribute Values**:
   - There are several missing values in some categorical attributes, all coded with the "unknown" label. These missing values can be treated as a possible class label or using deletion or imputation techniques.

0.3 Loading dependancies and data

[1]:
```
import calendar
from matplotlib.lines import Line2D
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import normaltest
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import MinMaxScaler, RobustScaler, StandardScaler
import tensorflow as tf
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.keras.layers import Input, Dense
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam, SGD
from warnings import filterwarnings
filterwarnings("ignore")
```

[2]:
```
pd.options.display.max_columns = None
```

[3]:
```
df = pd.read_csv("bank-additional-full.csv",
                 delimiter=";",
                 na_values=["unknown"],
                 false_values=["no"],
                 true_values=["yes"])
```

[4]:
```
df.head()
```

[4]:
```
   age        job  marital    education default housing   loan    contact  \
0   56  housemaid  married     basic.4y   False   False  False  telephone
1   57   services  married  high.school     NaN   False  False  telephone
2   37   services  married  high.school   False    True  False  telephone
3   40     admin.  married     basic.6y   False   False  False  telephone
4   56   services  married  high.school   False   False   True  telephone
```

```
  month day_of_week  duration  campaign  pdays  previous     poutcome  \
0   may         mon       261         1    999         0  nonexistent
1   may         mon       149         1    999         0  nonexistent
2   may         mon       226         1    999         0  nonexistent
3   may         mon       151         1    999         0  nonexistent
4   may         mon       307         1    999         0  nonexistent

   emp.var.rate  cons.price.idx  cons.conf.idx  euribor3m  nr.employed      y
0           1.1          93.994          -36.4      4.857       5191.0  False
1           1.1          93.994          -36.4      4.857       5191.0  False
2           1.1          93.994          -36.4      4.857       5191.0  False
3           1.1          93.994          -36.4      4.857       5191.0  False
4           1.1          93.994          -36.4      4.857       5191.0  False
```

0.4 Data Validation

- Renaming columns

[5]:
```
df.columns
```

[5]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'housing', 'loan',
       'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'emp.var.rate', 'cons.price.idx',
       'cons.conf.idx', 'euribor3m', 'nr.employed', 'y'],
      dtype='object')
```

- Action taking:
  - Replacing '.' by '_' to enhance usability of columns

[6]:
```
df.columns = df.columns.str.replace("\.", "_", regex=True)
```

- Data types

[7]:
```
df.dtypes
```

[7]:
```
age                int64
job               object
marital           object
education         object
default           object
housing           object
loan              object
contact           object
month             object
day_of_week       object
duration           int64
campaign           int64
pdays              int64
```

```
previous              int64
poutcome             object
emp_var_rate        float64
cons_price_idx      float64
cons_conf_idx       float64
euribor3m           float64
nr_employed         float64
y                      bool
dtype: object
```

- Data types were cast as expected
- Inspecting the values of categorical variables

[8]:

```
categorical_col = df.columns[df.dtypes=="object"]
for col in categorical_col:
    print()
    print(col.center(50,"-"))
    print(*sorted(filter(lambda x: x != "nan", map(str, df[col].unique()))),
 ↪sep=", ")
    print("="*50)
```

```
-----------------------job------------------------
admin., blue-collar, entrepreneur, housemaid, management, retired, self-
employed, services, student, technician, unemployed
==================================================

---------------------marital----------------------
divorced, married, single
==================================================

--------------------education---------------------
basic.4y, basic.6y, basic.9y, high.school, illiterate, professional.course,
university.degree
==================================================

---------------------default----------------------
False, True
==================================================

---------------------housing----------------------
False, True
==================================================

-----------------------loan-----------------------
False, True
==================================================
```

```
---------------------contact----------------------
cellular, telephone
==================================================

----------------------month-----------------------
apr, aug, dec, jul, jun, mar, may, nov, oct, sep
==================================================

-------------------day_of_week--------------------
fri, mon, thu, tue, wed
==================================================

---------------------poutcome---------------------
failure, nonexistent, success
==================================================
```

- Values are as expected

0.4.1 Handle missingness

[9]:
```python
nullity = (df.isna().mean() * 100).round(1)
cat_a = nullity[nullity >= 5].astype(str).to_dict()
cat_b = nullity[(nullity < 5) & (nullity > 0)].astype(str).to_dict()

print("Category A:", "`"+"`, `".join(cat_a.keys())+"`", f"ha(s\\ve) {'%, '.
↪join(cat_a.values())}% missingness")
print("Category B:", "`"+"`, `".join(cat_b.keys())+"`", f"ha(s\\ve) {'%, '.
↪join(cat_b.values())}% missingness")
```

```
Category A: `default` ha(s\ve) 20.9% missingness
Category B: `job`, `marital`, `education`, `housing`, `loan` ha(s\ve) 0.8%,
0.2%, 4.2%, 2.4%, 2.4% missingness
```

- Missing values in the first category will be further investigated.
- Rows with missing values in the second category will be trimmed.

[10]:
```python
df["default"].value_counts()
```

[10]:
```
False    32588
True         3
Name: default, dtype: int64
```

- Given the high proportion of missing values and the class imbalance in the `default` column, it may be more appropriate to remove the entire column.
- `default` column will be dropped.

[11]:
```python
df.drop(cat_a, axis=1, inplace=True)
df.dropna(subset=cat_b, inplace=True)
```

- Removing duplicates

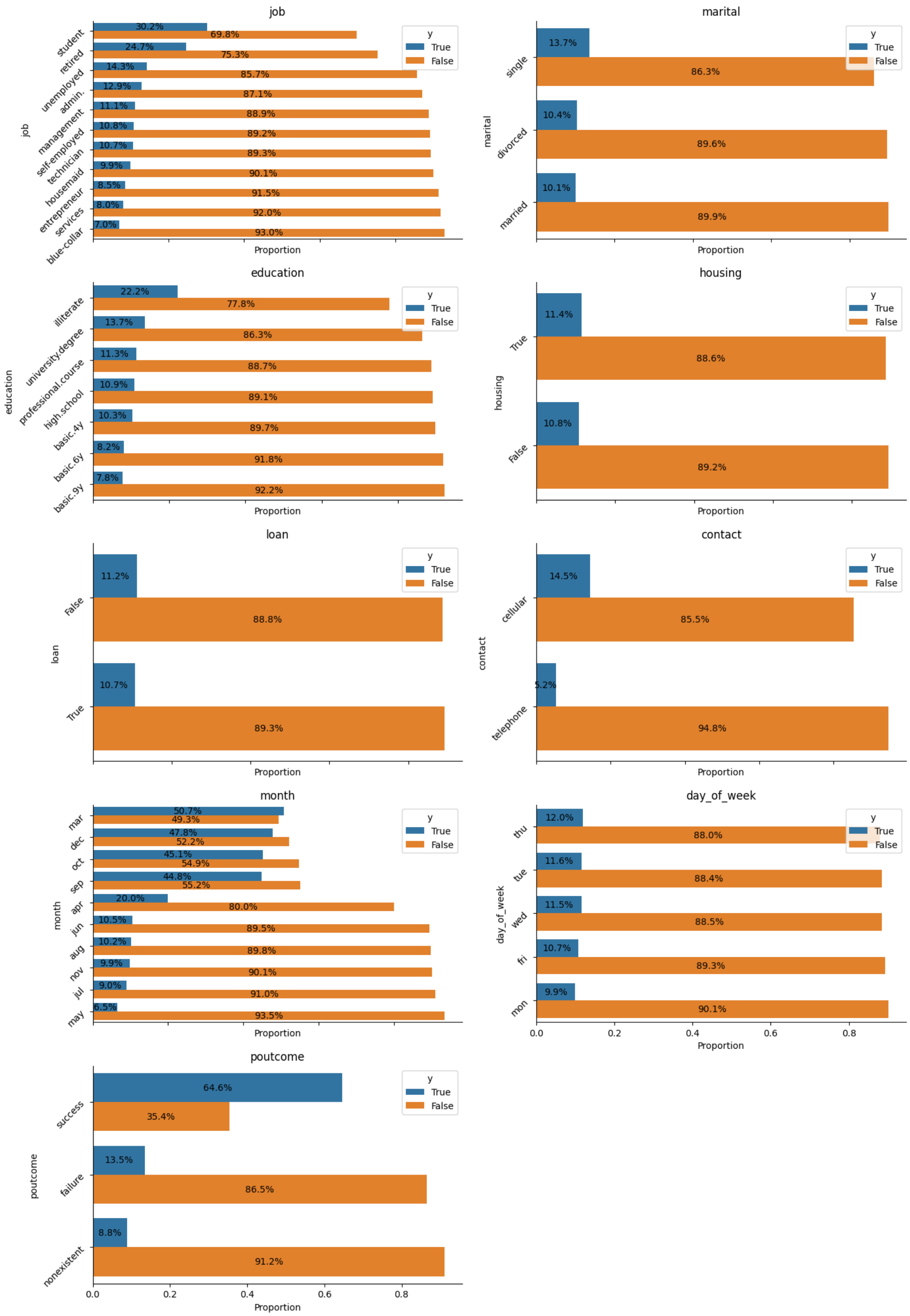
[12]:
```
duplicated = df.duplicated()
print(f"# Duplicates: {duplicated.sum()}")
```

```
# Duplicates: 13
```

[13]:
```
df.drop(df[duplicated].index, inplace=True)
df.reset_index(inplace=True, drop=True)
```

0.5 EDA

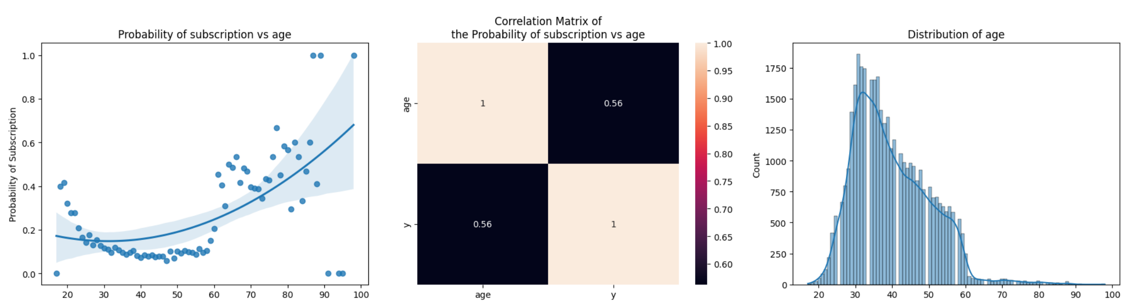
[14]:
```
i=1
n_axes = len(df.columns[df.dtypes == "object"])
n_rows=np.ceil(n_axes/2).astype(int)
fig = plt.figure(figsize=(16,5*n_rows))
# ax = ax.flatten()
for col in df.columns[df.dtypes == "object"]:
    ax = plt.subplot(n_rows,2,i)
    s = df.groupby(col)["y"].value_counts(normalize=True)
    s.name = "count"
    data = s.reset_index()
    data.iloc[:, :-1] = data.iloc[:, :-1].astype(str)
    order = data[data["y"] == "True"].sort_values("count",␣
 ↪ascending=False)[col].values

    sns.barplot(data=data, x="count", y=col, hue="y", hue_order=["True",␣
 ↪"False"], ax=ax, order=order)

    # set bar labels for client who subscribed to a term deposit as percentage
    ax.bar_label(ax.containers[0], data[data["y"] == "True"]["count"].
 ↪sort_values(ascending=False)\
                 .map(lambda x: f"{x:.1%}"), label_type="center")
    # set bar labels for client who didn't subscribed to a term deposit as␣
 ↪percentage
    ax.bar_label(ax.containers[1], data[data["y"] == "False"]["count"].
 ↪sort_values(ascending=True)\
                 .map(lambda x: f"{x:.1%}"), label_type="center")

    ax.tick_params("y", labelrotation=45)
    ax.spines["right"].set_visible(False)
    ax.spines["top"].set_visible(False)
    ax.set_title(col)
    ax.set_xlabel("Proportion")
    if i+2 <= n_axes:
        [tick.set_visible(False) for tick in ax.get_xticklabels()]
    i+=1
plt.show()
```

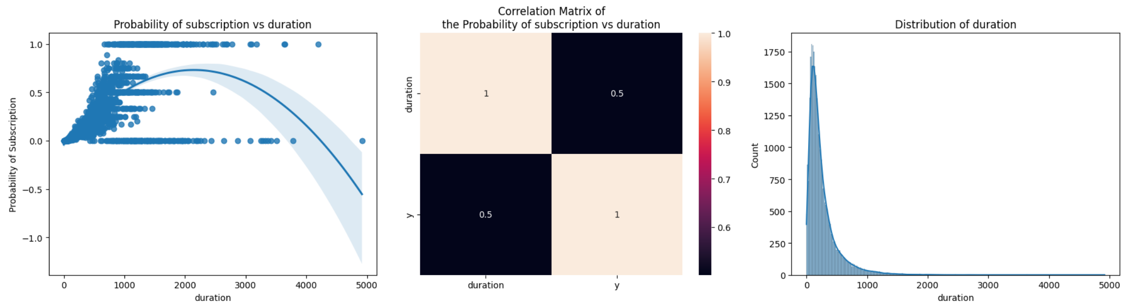

job
student
retired
unemployed
admin.
management
self-employed
technician
housemaid
entrepreneur
services
blue-collar
30.2%
69.8%
24.7%
75.3%
14.3%
85.7%
12.9%
87.1%
11.1%
88.9%
10.8%
89.2%
10.7%
89.3%
9.9%
90.1%
8.5%
91.5%
8.0%
92.0%
7.0%
93.0%
marital
single
divorced
married
13.7%
86.3%
10.4%
89.6%
10.1%
89.9%
education
illiterate
university.degree
professional.course
high.school
basic.4y
basic.6y
basic.9y
22.2%
77.8%
13.7%
86.3%
11.3%
88.7%
10.9%
89.1%
10.3%
89.7%
8.2%
91.8%
7.8%
92.2%
housing
True
False
11.4%
88.6%
10.8%
89.2%
loan
False
True
11.2%
88.8%
10.7%
89.3%
contact
cellular
telephone
14.5%
85.5%
5.2%
94.8%
month
mar
dec
oct
sep
apr
jun
aug
nov
jul
may
50.7%
49.3%
47.8%
52.2%
45.1%
54.9%
44.8%
55.2%
20.0%
80.0%
10.5%
89.5%
10.2%
89.8%
9.9%
90.1%
9.0%
91.0%
6.5%
93.5%
day_of_week
thu
tue
wed
fri
mon
12.0%
88.0%
11.6%
88.4%
11.5%
88.5%
10.7%
89.3%
9.9%
90.1%
poutcome
success
failure
nonexistent
64.6%
35.4%
13.5%
86.5%
8.8%
91.2%
y
True
False
Proportion
0.0
0.2
0.4
0.6
0.8

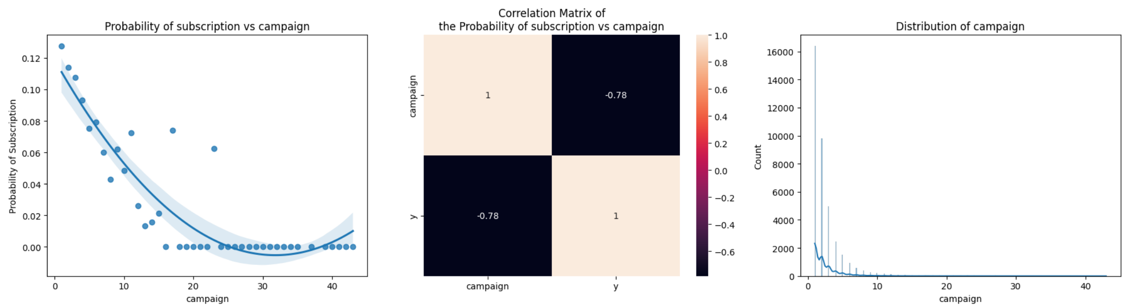
- **Job**:
  - Student: 30% success rate
  - Retired: 25% success rate
- **Education**:
  - Illiterate: 22% success rate
- **Contact**:
  - Cellular: 14.5% success rate
  - Phone: 5.2% success rate
- **Month**:
  - March: 50.7% success rate
  - December: 47.8% success rate
  - October and September: 45% success rate each
- **Poutcome**:
  - Success: 64.6%
  - Failure: 13.5%
  - Nonexistent: 8.8%

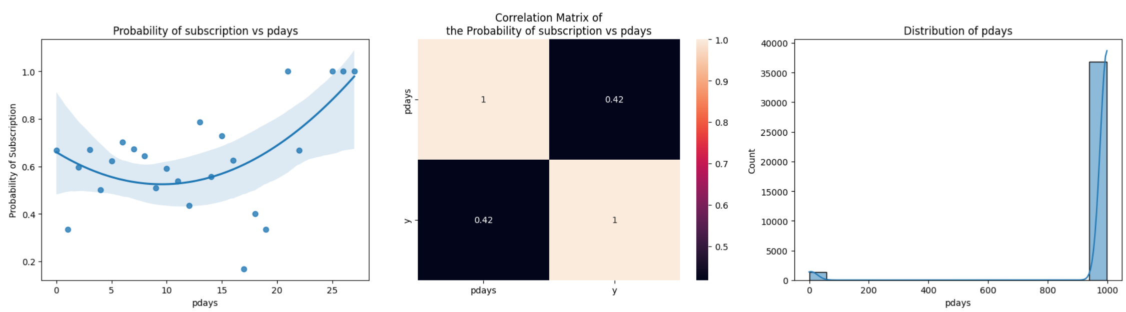
[15]:

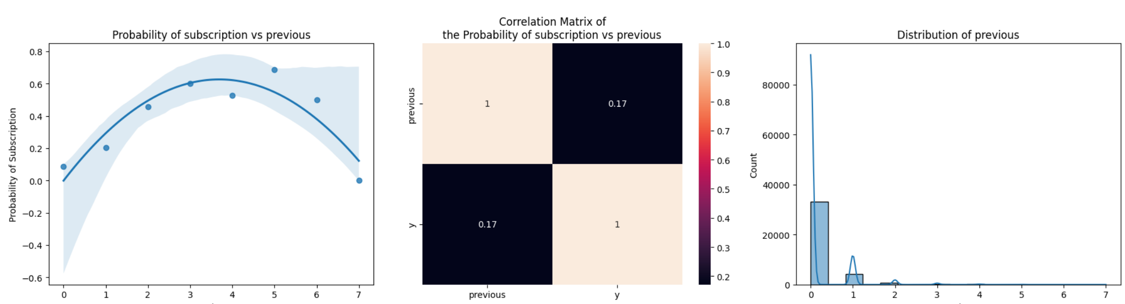
```python
columns = df.columns[df.dtypes != "object"].tolist()
columns.remove("y")
for col in columns:
    fig, (ax1, ax2, ax3) = plt.subplots(1,3, figsize=(18,5))
    if col == "pdays":
        data = df[df["pdays"] != 999].groupby(col)["y"].mean().reset_index()
    else:
        data = df.groupby(col)["y"].mean().reset_index()
    sns.regplot(data, x=col, y="y", order=2, ax=ax1)
    ax1.set_title(f"Probability of subscription vs {col}")
    ax1.set_ylabel("Probability of Subscription")
    sns.heatmap(data.corr(), annot=True, ax=ax2)
    ax2.set_title(f"Correlation Matrix of\nthe Probability of subscription vs {col}")
    sns.histplot(df[col], ax=ax3, kde=True)
    ax3.set_title(f"Distribution of {col}")
    fig.tight_layout()
    plt.show()
```

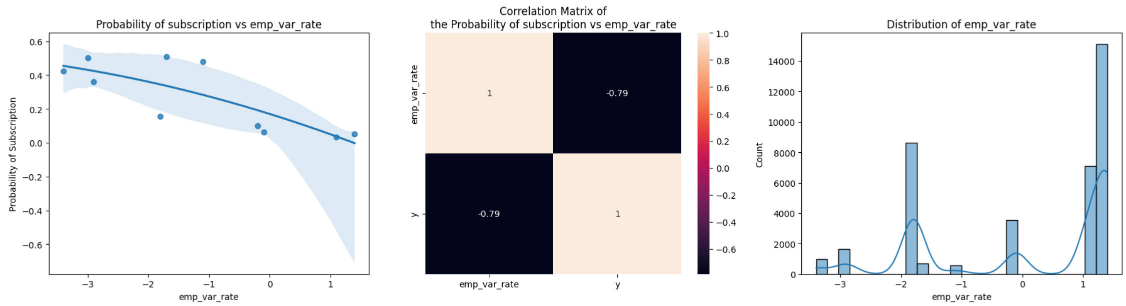


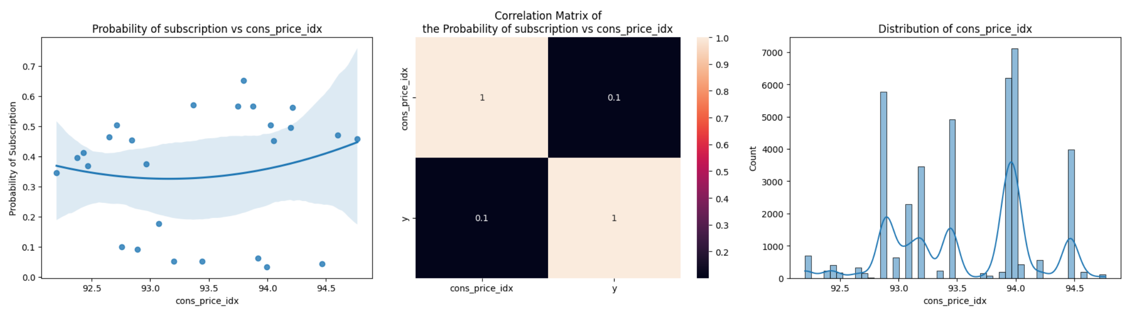

Probability of subscription vs duration
Probability of Subscription
duration
Correlation Matrix of
the Probability of subscription vs duration
duration
y
Distribution of duration
Count
duration

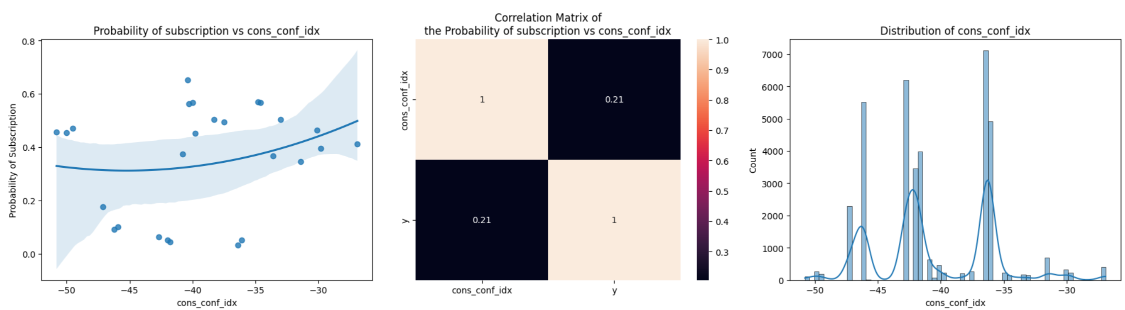

Probability of subscription vs campaign
Probability of Subscription
campaign
Correlation Matrix of
the Probability of subscription vs campaign
campaign
y
Distribution of campaign
Count
campaign

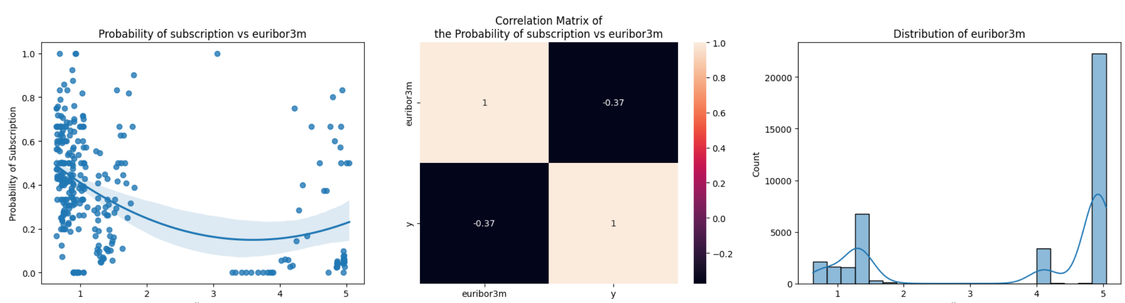

Probability of subscription vs pdays
Probability of Subscription
pdays
Correlation Matrix of
the Probability of subscription vs pdays
pdays
y
Distribution of pdays
Count
pdays

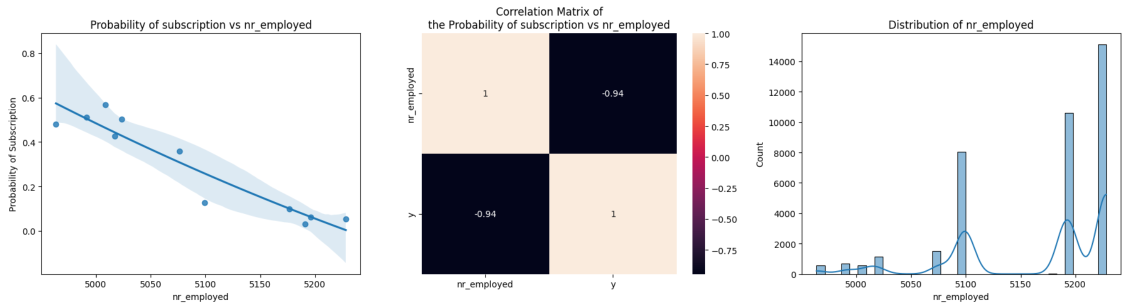

Probability of subscription vs previous
Probability of Subscription
previous
Correlation Matrix of
the Probability of subscription vs previous
previous
y
Distribution of previous
Count
previous


Probability of subscription vs emp_var_rate
Probability of Subscription
emp_var_rate
Correlation Matrix of
the Probability of subscription vs emp_var_rate
emp_var_rate
y
1
-0.79
-0.79
1
Distribution of emp_var_rate
Count
emp_var_rate


Probability of subscription vs cons_price_idx
Probability of Subscription
cons_price_idx
Correlation Matrix of
the Probability of subscription vs cons_price_idx
cons_price_idx
y
1
0.1
0.1
1
Distribution of cons_price_idx
Count
cons_price_idx


Probability of subscription vs cons_conf_idx
Probability of Subscription
cons_conf_idx
Correlation Matrix of
the Probability of subscription vs cons_conf_idx
cons_conf_idx
y
1
0.21
0.21
1
Distribution of cons_conf_idx
Count
cons_conf_idx


Probability of subscription vs euribor3m
Probability of Subscription
euribor3m
Correlation Matrix of
the Probability of subscription vs euribor3m
euribor3m
y
1
-0.37
-0.37
1
Distribution of euribor3m
Count
euribor3m



- age
  1. `age` and the proportion of the response **moderately correlated**
  2. The distribution of `age` is **right skewed** >meaning that most individuals are younger, but there is a long tail of older individuals.
- duration
  1. `duration` and the proportion of the response **moderately correlated**
  2. `duration` exhibits **exponential distribution** >suggests that most contacts are of short length, with fewer long-duration calls
- campaign
  1. `campaign` and the proportion of the response **negatively correlated**
  2. `campaign` exhibits **geometric distribution** >meaning the likelihood of a response decreases as the number of contacts increases.
- pdays
  1. `pdays` and the proportion of the response **moderately correlated**
  2. The distribution of `pdays` suggests that the majority of engagements weren't preceded by any contact in the previous campaign.
- previous
  1. `previous` and the proportion of the response **weakly correlated**
  2. `previous` exhibits **geometric distribution**
- emp_var_rate
  1. `emp_var_rate` and the proportion of the response **negatively correlated**
- cons_price_idx
  1. `cons_price_idx` and the proportion of the response **weakly correlated**
- cons_conf_idx
  1. `cons_conf_idx` and the proportion of the response **weaklly correlated**
- euribor3m
  1. `euribor3m` and the proportion of the response **moderately correlated**
- nr_employed
  1. `nr_employed` and the proportion of the response **negatively correlated**

From sight we can initially say taht all distributions are not following normal distribution. This issue will be handled by using sequential robust min-max scaler.

[16]:
```
df.select_dtypes(include=["number", "bool"]).corr().loc[:, ["y"]].sort_values("y", ascending=False).drop("y")
```

[16]:
```
                      y
duration         0.405856
previous         0.221178
cons_conf_idx    0.051363
age              0.030123
campaign        -0.065125
cons_price_idx  -0.133000
emp_var_rate    -0.292209
euribor3m       -0.300540
pdays           -0.319386
nr_employed     -0.347816
```

- `age` and response **weakly correlated**
- `duration` and response **moderately correlated**
- `campaign` and response **negatively weakly correlated**
- `pdays` and response **-ve moderately correlated**
- `previous` and response **weakly correlated**
- `emp_var_rate` and response **-ve moderately correlated**
- `cons_price_idx` and response **-ve weakly correlated**
- `cons_conf_idx` and response **weaklly correlated**
- `euribor3m` and response **-ve moderately correlated**
- `nr_employed` and response **-ve moderately correlated**

conclusions

- **Demographic Factors:**
  - Students, Retired, and Unemployed: These groups have a higher likelihood of subscribing to a term deposit compared to others. This could be due to different financial stability or investment interests.
- **Age Factor:**
  - Individuals at both ends of the age spectrum are more likely to subscribe. This might reflect different financial priorities or investment strategies among younger and older people.
- **Communication Channel:**
  - Reaching out to clients via cellular communication increases the likelihood of subscription. This suggests that personal and direct communication might be more effective than other methods.
- **Seasonal Trends:**
  - Subscription rates fluctuate depending on the month, possibly due to economic conditions or financial behaviors that vary throughout the year.
- **Previous Campaign Interactions:**
  - People who were contacted in previous campaigns are more likely to subscribe. Furthermore, those who had successful outcomes in prior campaigns are even more likely to subscribe. This implies that past engagement and success can positively influence future decisions.
- **Frequency of Contact:**
  - Contacting individuals multiple times within the same campaign increases subscription rates, but there's a limit to its effectiveness (about 25 contacts). This suggests that

while persistence can be beneficial, there's a diminishing return after a certain point.

0.6 Feature Engineering

0.6.1 Variables with Potential Spurious Correlations:

- **Duration of Contact**: It is not known before contact
- **Month**: Could reflect seasonal or economic effects rather than individual customer behavior.
- **Day of the Week**: May show operational patterns, not true customer preferences.
- **Contact Method (Telephone vs. Cellular)**: Might be influenced by demographic factors.
- **Job Category**: Could be a proxy for socioeconomic status rather than a direct influence.
- **Education Level**: Could be a proxy for financial literacy or product accessibility.
- **Housing/Personal Loans**: Might indicate financial conditions rather than the likelihood of subscription.

0.6.2 Variables with Strong Potential for Spurious Associations:

1. **`duration`**
2. **`month`**
3. **`day_of_week`**

- I decided to remove these three variables to avoid spurious correlations and improve model reliability.

[17]:
```
df_cleaned = df.copy().drop(["duration", "month", "day_of_week"], axis=1)
```

[20]:
```
df_cleaned["education_job"] = df_cleaned["education"] + "_" + df_cleaned["job"]
df_cleaned.drop(["education", "job"], axis=1, inplace=True)
```

[21]:
```
onehot_cols = df_cleaned.dtypes[df_cleaned.dtypes=="object"].index.to_list()
label_cols = ["y"]
df_cleaned = pd.get_dummies(df_cleaned, columns=onehot_cols, drop_first=True)
df_cleaned[label_cols] = df[label_cols].astype(np.uint8)
```

[22]:
```
df_cleaned.head()
```

[22]:
```
   age  campaign  pdays  previous  emp_var_rate  cons_price_idx  \
0   56         1    999         0           1.1          93.994
1   57         1    999         0           1.1          93.994
2   37         1    999         0           1.1          93.994
3   40         1    999         0           1.1          93.994
4   56         1    999         0           1.1          93.994

   cons_conf_idx  euribor3m  nr_employed  y  marital_married  marital_single  \
0          -36.4      4.857       5191.0  0                1               0
1          -36.4      4.857       5191.0  0                1               0
2          -36.4      4.857       5191.0  0                1               0
3          -36.4      4.857       5191.0  0                1               0
4          -36.4      4.857       5191.0  0                1               0
```

	housing_True	loan_True	contact_telephone	poutcome_nonexistent	\
0	0	0	1	1	
1	0	0	1	1	
2	1	0	1	1	
3	0	0	1	1	
4	0	1	1	1	

	poutcome_success	education_job_basic.4y_blue-collar	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	education_job_basic.4y_entrepreneur	education_job_basic.4y_housemaid	\
0	0	1	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	education_job_basic.4y_management	education_job_basic.4y_retired	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	education_job_basic.4y_self-employed	education_job_basic.4y_services	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	education_job_basic.4y_student	education_job_basic.4y_technician	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	

	education_job_basic.4y_unemployed	education_job_basic.6y_admin.	\
0	0	0	
1	0	0	
2	0	0	

```
3                                 0                              1
4                                 0                              0

   education_job_basic.6y_blue-collar  education_job_basic.6y_entrepreneur  \
0                                   0                                    0
1                                   0                                    0
2                                   0                                    0
3                                   0                                    0
4                                   0                                    0

   education_job_basic.6y_housemaid  education_job_basic.6y_management  \
0                                 0                                  0
1                                 0                                  0
2                                 0                                  0
3                                 0                                  0
4                                 0                                  0

   education_job_basic.6y_retired  education_job_basic.6y_self-employed  \
0                               0                                     0
1                               0                                     0
2                               0                                     0
3                               0                                     0
4                               0                                     0

   education_job_basic.6y_services  education_job_basic.6y_student  \
0                                0                               0
1                                0                               0
2                                0                               0
3                                0                               0
4                                0                               0

   education_job_basic.6y_technician  education_job_basic.6y_unemployed  \
0                                  0                                  0
1                                  0                                  0
2                                  0                                  0
3                                  0                                  0
4                                  0                                  0

   education_job_basic.9y_admin.  education_job_basic.9y_blue-collar  \
0                              0                                   0
1                              0                                   0
2                              0                                   0
3                              0                                   0
4                              0                                   0

   education_job_basic.9y_entrepreneur  education_job_basic.9y_housemaid  \
0                                    0                                 0
```

```
1                                    0                                  0
2                                    0                                  0
3                                    0                                  0
4                                    0                                  0

   education_job_basic.9y_management  education_job_basic.9y_retired  \
0                                  0                               0
1                                  0                               0
2                                  0                               0
3                                  0                               0
4                                  0                               0

   education_job_basic.9y_self-employed  education_job_basic.9y_services  \
0                                     0                                0
1                                     0                                0
2                                     0                                0
3                                     0                                0
4                                     0                                0

   education_job_basic.9y_student  education_job_basic.9y_technician  \
0                               0                                  0
1                               0                                  0
2                               0                                  0
3                               0                                  0
4                               0                                  0

   education_job_basic.9y_unemployed  education_job_high.school_admin.  \
0                                  0                                  0
1                                  0                                  0
2                                  0                                  0
3                                  0                                  0
4                                  0                                  0

   education_job_high.school_blue-collar  \
0                                      0
1                                      0
2                                      0
3                                      0
4                                      0

   education_job_high.school_entrepreneur  \
0                                       0
1                                       0
2                                       0
3                                       0
4                                       0
```

```
   education_job_high.school_housemaid  education_job_high.school_management  \
0                                    0                                     0
1                                    0                                     0
2                                    0                                     0
3                                    0                                     0
4                                    0                                     0

   education_job_high.school_retired  education_job_high.school_self-employed  \
0                                  0                                        0
1                                  0                                        0
2                                  0                                        0
3                                  0                                        0
4                                  0                                        0

   education_job_high.school_services  education_job_high.school_student  \
0                                   0                                  0
1                                   1                                  0
2                                   1                                  0
3                                   0                                  0
4                                   1                                  0

   education_job_high.school_technician  education_job_high.school_unemployed  \
0                                     0                                     0
1                                     0                                     0
2                                     0                                     0
3                                     0                                     0
4                                     0                                     0

   education_job_illiterate_admin.  education_job_illiterate_blue-collar  \
0                                0                                     0
1                                0                                     0
2                                0                                     0
3                                0                                     0
4                                0                                     0

   education_job_illiterate_entrepreneur  education_job_illiterate_housemaid  \
0                                      0                                   0
1                                      0                                   0
2                                      0                                   0
3                                      0                                   0
4                                      0                                   0

   education_job_illiterate_retired  education_job_illiterate_self-employed  \
0                                 0                                       0
1                                 0                                       0
2                                 0                                       0
3                                 0                                       0
```

```
4                              0                                    0

   education_job_professional.course_admin.  \
0                                         0
1                                         0
2                                         0
3                                         0
4                                         0

   education_job_professional.course_blue-collar  \
0                                              0
1                                              0
2                                              0
3                                              0
4                                              0

   education_job_professional.course_entrepreneur  \
0                                               0
1                                               0
2                                               0
3                                               0
4                                               0

   education_job_professional.course_housemaid  \
0                                            0
1                                            0
2                                            0
3                                            0
4                                            0

   education_job_professional.course_management  \
0                                             0
1                                             0
2                                             0
3                                             0
4                                             0

   education_job_professional.course_retired  \
0                                          0
1                                          0
2                                          0
3                                          0
4                                          0

   education_job_professional.course_self-employed  \
0                                                0
1                                                0
```

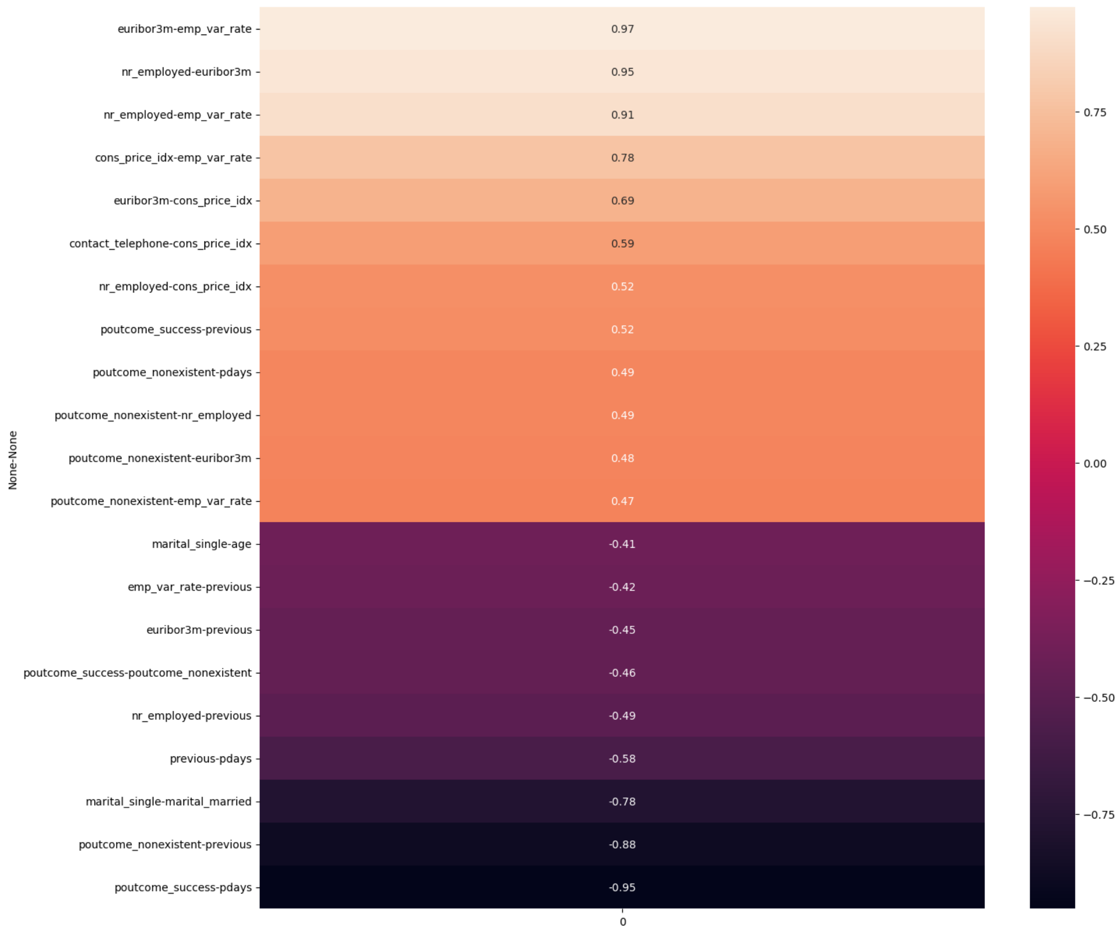
```
2                                               0
3                                               0
4                                               0

   education_job_professional.course_services  \
0                                          0
1                                          0
2                                          0
3                                          0
4                                          0

   education_job_professional.course_student  \
0                                         0
1                                         0
2                                         0
3                                         0
4                                         0

   education_job_professional.course_technician  \
0                                            0
1                                            0
2                                            0
3                                            0
4                                            0

   education_job_professional.course_unemployed  \
0                                            0
1                                            0
2                                            0
3                                            0
4                                            0

   education_job_university.degree_admin.  \
0                                      0
1                                      0
2                                      0
3                                      0
4                                      0

   education_job_university.degree_blue-collar  \
0                                           0
1                                           0
2                                           0
3                                           0
4                                           0

   education_job_university.degree_entrepreneur  \
```

```
0                                              0
1                                              0
2                                              0
3                                              0
4                                              0

   education_job_university.degree_housemaid  \
0                                          0
1                                          0
2                                          0
3                                          0
4                                          0

   education_job_university.degree_management  \
0                                           0
1                                           0
2                                           0
3                                           0
4                                           0

   education_job_university.degree_retired  \
0                                        0
1                                        0
2                                        0
3                                        0
4                                        0

   education_job_university.degree_self-employed  \
0                                              0
1                                              0
2                                              0
3                                              0
4                                              0

   education_job_university.degree_services  \
0                                         0
1                                         0
2                                         0
3                                         0
4                                         0

   education_job_university.degree_student  \
0                                        0
1                                        0
2                                        0
3                                        0
4                                        0
```

```
   education_job_university.degree_technician  \
0                                           0
1                                           0
2                                           0
3                                           0
4                                           0

   education_job_university.degree_unemployed
0                                           0
1                                           0
2                                           0
3                                           0
4                                           0
```

[23]:

```
corr_matrix = df_cleaned.corr().drop("y", axis=1)
corr_matrix = corr_matrix[:][(corr_matrix >= .4)|(corr_matrix <= -.4)]
corr_matrix = corr_matrix.dropna(how="all").dropna(how="all", axis=1)
shape = corr_matrix.values.shape[0]
indices = [[True if j < i else False for j in range(shape)] for i in␣
 ↪range(shape)]
corr_matrix = corr_matrix[pd.DataFrame(indices, index=corr_matrix.index,␣
 ↪columns=corr_matrix.columns)]

fig, ax = plt.subplots(figsize=(15,15))
vmax=corr_matrix.max().max()
vmin=corr_matrix.min().min()
sns.heatmap(corr_matrix.stack().sort_values(ascending=False).to_frame(),␣
 ↪vmax=vmax, vmin=vmin, ax=ax, annot=True)
plt.show()
```



[24]:

```
corr_matrix.stack().sort_values(ascending=False).to_frame()
```

[24]:

		0
euribor3m	emp_var_rate	0.972421
nr_employed	euribor3m	0.945328
	emp_var_rate	0.907898
cons_price_idx	emp_var_rate	0.775385
euribor3m	cons_price_idx	0.689554
contact_telephone	cons_price_idx	0.592909
nr_employed	cons_price_idx	0.524188
poutcome_success	previous	0.519892
poutcome_nonexistent	pdays	0.486286
	nr_employed	0.485282
	euribor3m	0.482562
	emp_var_rate	0.468716
marital_single	age	-0.408988
emp_var_rate	previous	-0.419750
euribor3m	previous	-0.450753

```
poutcome_success     poutcome_nonexistent -0.463107
nr_employed          previous             -0.494700
previous             pdays                -0.581296
marital_single       marital_married      -0.776228
poutcome_nonexistent previous             -0.881786
poutcome_success     pdays                -0.952692
```

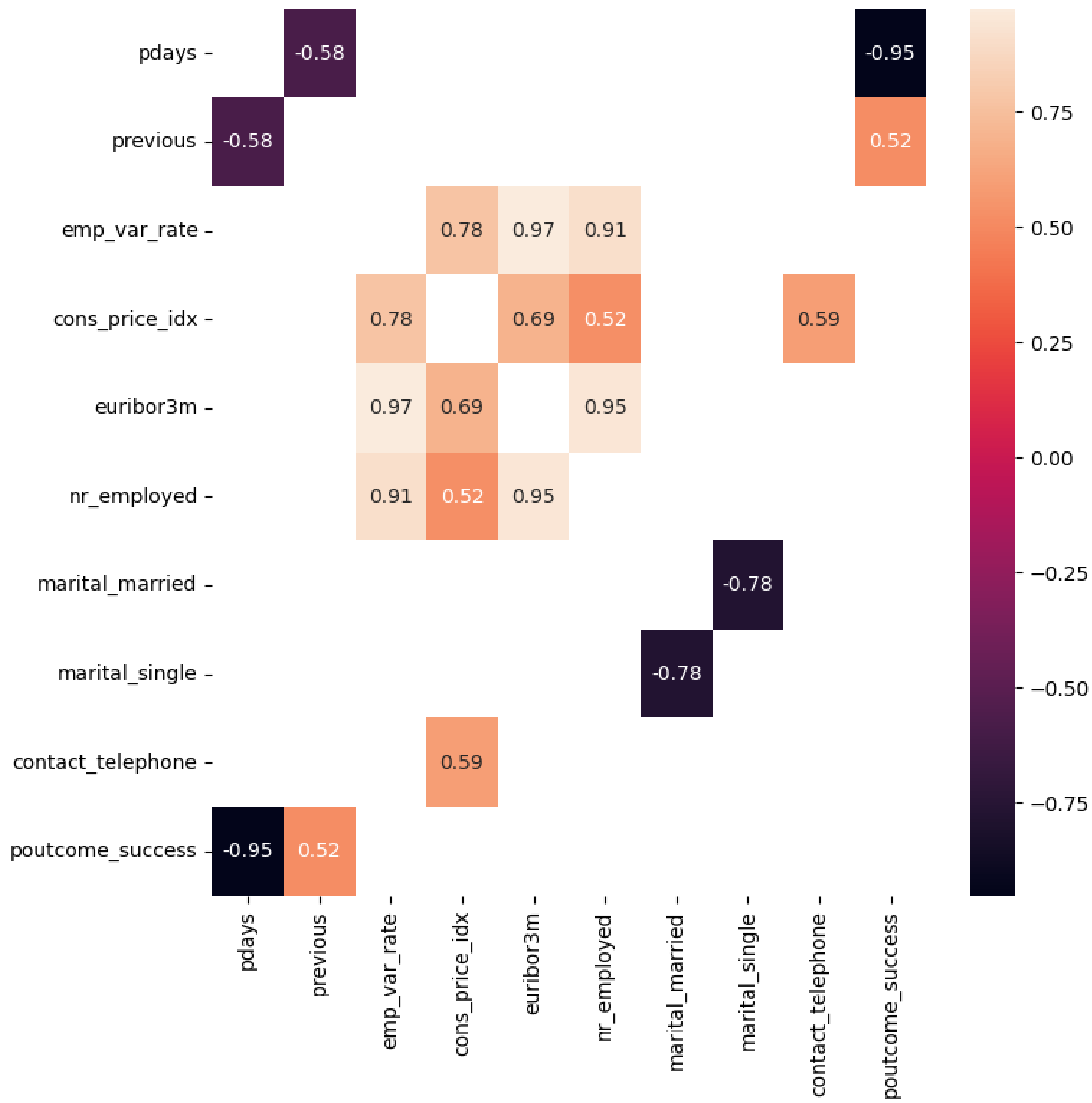
1. **`emp_var_rate`** has high correlations with:
   - `euribor3m` (0.972)
   - `nr_employed` (0.908)
2. **`euribor3m`** has a high correlation with:
   - `nr_employed` (0.945)
3. **`poutcome_nonexistent`** has a strong negative correlation with:
   - `previous` (-0.882)

0.6.3 Actions:

1. **Feature Reduction**: Dropping `poutcome_nonexistent`.
2. **Feature Engineering**: Create new features that capture underlying information without redundancy. For example, combining `emp_var_rate`, `euribor3m`, and `nr_employed` into a single composite index might be useful. > It was found to negatively impact model performance, so this approach was discarded.

[25]:
```
cloumns_to_drop = ["poutcome_nonexistent"]
df_cleaned.drop(cloumns_to_drop, axis=1, inplace=True, errors="ignore")
```

[26]:
```
corr_matrix = df_cleaned.corr().drop("y", axis=1)
corr_matrix = corr_matrix[:][corr_matrix != 1]
corr_matrix = corr_matrix[:][(corr_matrix >= .5)|(corr_matrix <= -.5)]
corr_matrix = corr_matrix.dropna(how="all").dropna(how="all", axis=1)
fig, ax = plt.subplots(figsize=(8,8))
vmax=corr_matrix.max().max()
vmin=corr_matrix.min().min()
sns.heatmap(corr_matrix, vmax=vmax, vmin=vmin, ax=ax, annot=True)
plt.show()
```



0.6.4 Feature Scaling

[27]:
```
(normaltest(df_cleaned).pvalue <= .05)
```

[27]:
```
array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True,  True,  True,  True,
```

```
        True,  True,  True,  True,  True,  True,  True,  True,  True,
        True,  True,  True,  True,  True,  True])
```

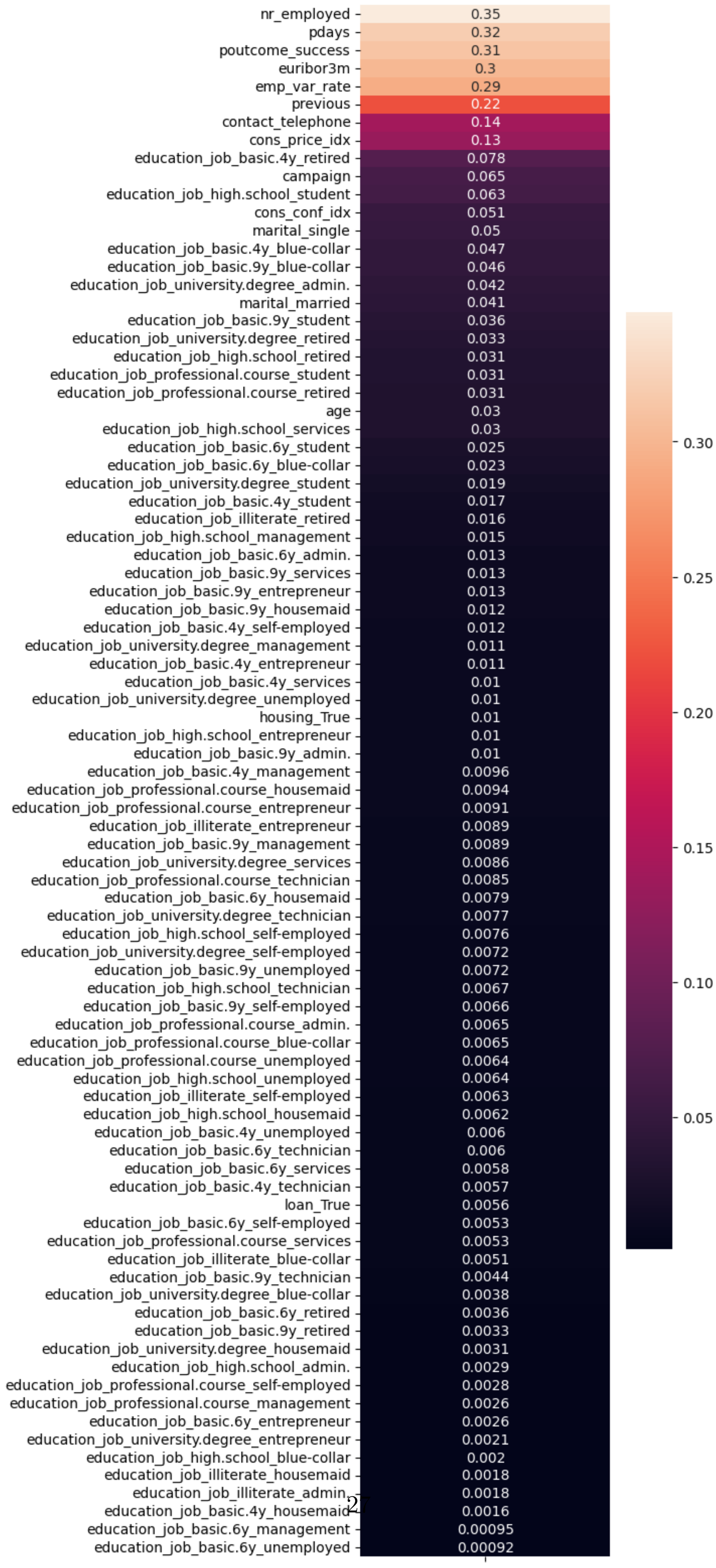
- Since the p-values from the normality tests indicate that none of the features follow a normal distribution (i.e., all p-values are 0.05), I have decided to use Robust min-max scaling.

[28]:
```
robust_scaler = RobustScaler()
minmax_scaler = MinMaxScaler()
def scaler(data):
    data = robust_scaler.fit_transform(data)
    data = minmax_scaler.fit_transform(data)
    return data
```

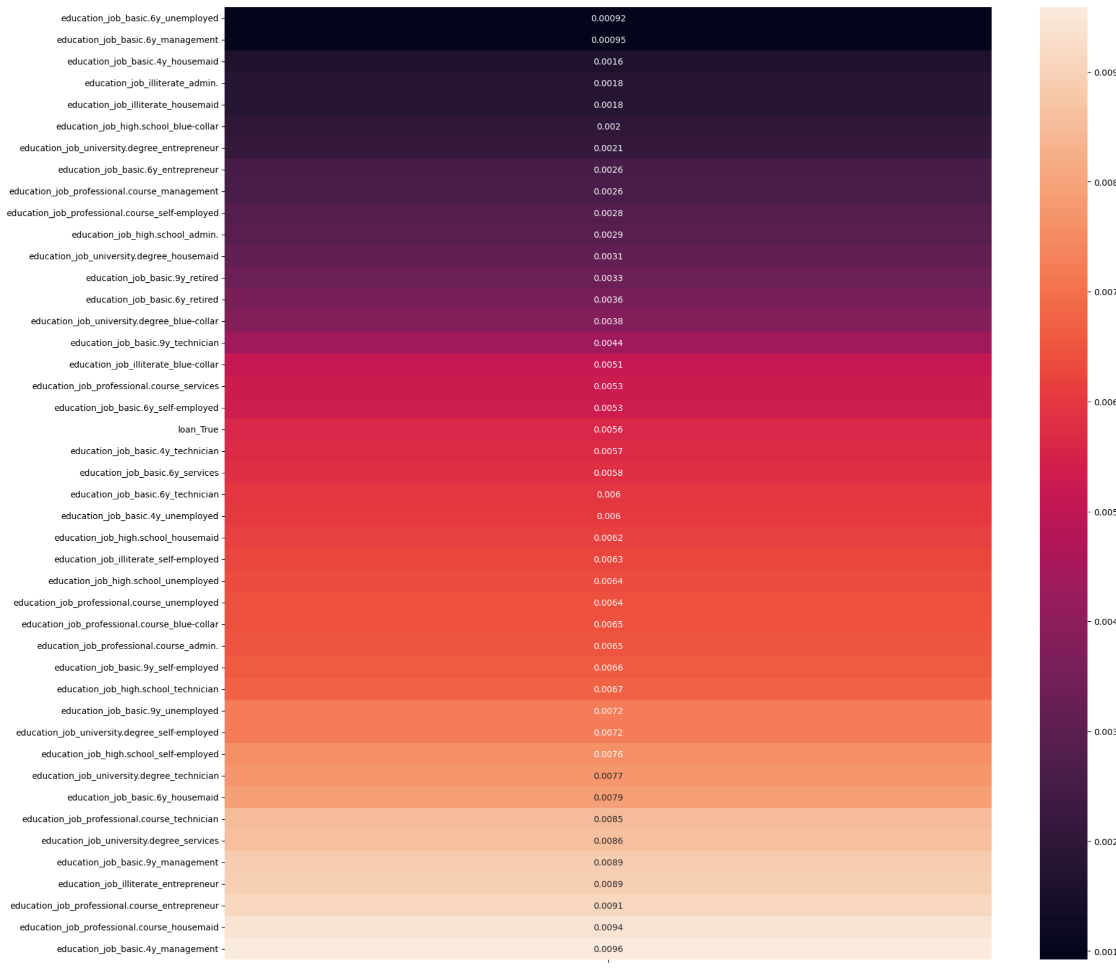
[29]:
```
df_cleaned[:] = scaler(df_cleaned)
```

[30]:
```
## discarded
# pca = PCA(n_components=1)
# df_cleaned["composite_feature"] = scaler(pca.
 ↪fit_transform(df_cleaned[["emp_var_rate", "euribor3m", "nr_employed"]])).
 ↪flatten()

# df_cleaned.drop(["emp_var_rate", "euribor3m", "nr_employed"], axis=1,␣
 ↪inplace=True, errors="ignore")
```

[31]:
```
corr_y = df_cleaned.corr().drop("y")["y"].abs().sort_values(ascending=False)
vmax = corr_y.max()
vmin = corr_y.min()
plt.figure(figsize=(4,20))
sns.heatmap(corr_y.to_frame(), annot=True, vmax=vmax, vmin=vmin)
plt.show()
```


nr_employed 0.35
pdays 0.32
poutcome_success 0.31
euribor3m 0.3
emp_var_rate 0.29
previous 0.22
contact_telephone 0.14
cons_price_idx 0.13
education_job_basic.4y_retired 0.078
campaign 0.065
education_job_high.school_student 0.063
cons_conf_idx 0.051
marital_single 0.05
education_job_basic.4y_blue-collar 0.047
education_job_basic.9y_blue-collar 0.046
education_job_university.degree_admin. 0.042
marital_married 0.041
education_job_basic.9y_student 0.036
education_job_university.degree_retired 0.033
education_job_high.school_retired 0.031
education_job_professional.course_student 0.031
education_job_professional.course_retired 0.031
age 0.03
education_job_high.school_services 0.03
education_job_basic.6y_student 0.025
education_job_basic.6y_blue-collar 0.023
education_job_university.degree_student 0.019
education_job_basic.4y_student 0.017
education_job_illiterate_retired 0.016
education_job_high.school_management 0.015
education_job_basic.6y_admin. 0.013
education_job_basic.9y_services 0.013
education_job_basic.9y_entrepreneur 0.013
education_job_basic.9y_housemaid 0.012
education_job_basic.4y_self-employed 0.012
education_job_university.degree_management 0.011
education_job_basic.4y_entrepreneur 0.011
education_job_basic.4y_services 0.01
education_job_university.degree_unemployed 0.01
housing_True 0.01
education_job_high.school_entrepreneur 0.01
education_job_basic.9y_admin. 0.01
education_job_basic.4y_management 0.0096
education_job_professional.course_housemaid 0.0094
education_job_professional.course_entrepreneur 0.0091
education_job_illiterate_entrepreneur 0.0089
education_job_basic.9y_management 0.0089
education_job_university.degree_services 0.0086
education_job_professional.course_technician 0.0085
education_job_basic.6y_housemaid 0.0079
education_job_university.degree_technician 0.0077
education_job_high.school_self-employed 0.0076
education_job_university.degree_self-employed 0.0072
education_job_basic.9y_unemployed 0.0072
education_job_high.school_technician 0.0067
education_job_basic.9y_self-employed 0.0066
education_job_professional.course_admin. 0.0065
education_job_professional.course_blue-collar 0.0065
education_job_professional.course_unemployed 0.0064
education_job_high.school_unemployed 0.0064
education_job_illiterate_self-employed 0.0063
education_job_high.school_housemaid 0.0062
education_job_basic.4y_unemployed 0.006
education_job_basic.6y_technician 0.006
education_job_basic.6y_services 0.0058
education_job_basic.4y_technician 0.0057
loan_True 0.0056
education_job_basic.6y_self-employed 0.0053
education_job_professional.course_services 0.0053
education_job_illiterate_blue-collar 0.0051
education_job_basic.9y_technician 0.0044
education_job_university.degree_blue-collar 0.0038
education_job_basic.6y_retired 0.0036
education_job_basic.9y_retired 0.0033
education_job_university.degree_housemaid 0.0031
education_job_high.school_admin. 0.0029
education_job_professional.course_self-employed 0.0028
education_job_professional.course_management 0.0026
education_job_basic.6y_entrepreneur 0.0026
education_job_university.degree_entrepreneur 0.0021
education_job_high.school_blue-collar 0.002
education_job_illiterate_housemaid 0.0018
education_job_illiterate_admin. 0.0018
education_job_basic.4y_housemaid 0.0016
education_job_basic.6y_management 0.00095
education_job_basic.6y_unemployed 0.00092
y
0.30
0.25
0.20
0.15
0.10
0.05

[32]:
```
corr_y = df_cleaned.corr().abs()
corr_y = corr_y[corr_y < .01]
corr_y.dropna(how="all", inplace=True)
corr_y.dropna(how="all", axis=1, inplace=True)
corr_y = corr_y[["y"]].sort_values("y").dropna()
vmax = corr_y.max().max()
vmin = corr_y.min().min()
plt.figure(figsize=(20,20))
sns.heatmap(corr_y, annot=True, vmax=vmax, vmin=vmin)
plt.show()
```



[33]:
```
df_cleaned.drop(corr_y.index, axis=1, inplace=True)
```

0.7 Model building

[34]:
```
df_cleaned.drop(["cluster","y"], axis=1, errors="ignore", inplace=True)

input_dim = df_cleaned.shape[1],
encoding_dim = 200

input_layer = Input(shape=input_dim)
encoded = Dense(encoding_dim, activation="relu")(input_layer)
encoded = Dense(encoding_dim // 2, activation="relu")(encoded)
encoded = Dense(encoding_dim // 3, activation="relu")(encoded)
encoded = Dense(encoding_dim // 6, activation="relu")(encoded)
encoded = Dense(encoding_dim // 12, activation="relu")(encoded)

encoded = Dense(encoding_dim // 25, activation="relu")(encoded)

decoded = Dense(encoding_dim // 12, activation="relu")(encoded)
decoded = Dense(encoding_dim // 6, activation="relu")(decoded)
decoded = Dense(encoding_dim // 3, activation="relu")(decoded)
decoded = Dense(encoding_dim // 2, activation="relu")(decoded)
decoded = Dense(encoding_dim, activation="relu")(decoded)
decoded = Dense(input_dim[0], activation="sigmoid")(decoded)

autoencoder = Model(input_layer, decoded)
encoder = Model(input_layer, encoded)

optimizer = Adam(learning_rate=0.0001)
autoencoder.compile(optimizer=optimizer, loss="binary_crossentropy")

autoencoder.summary()
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 42)]              0
_________________________________________________________________
dense (Dense)                (None, 200)               8600
_________________________________________________________________
dense_1 (Dense)              (None, 100)               20100
_________________________________________________________________
dense_2 (Dense)              (None, 66)                6666
_________________________________________________________________
dense_3 (Dense)              (None, 33)                2211
_________________________________________________________________
dense_4 (Dense)              (None, 16)                544
_________________________________________________________________
dense_5 (Dense)              (None, 8)                 136
```

```
_________________________________________________________________
dense_6 (Dense)              (None, 16)                144

_________________________________________________________________
dense_7 (Dense)              (None, 33)                561

_________________________________________________________________
dense_8 (Dense)              (None, 66)                2244

_________________________________________________________________
dense_9 (Dense)              (None, 100)               6700

_________________________________________________________________
dense_10 (Dense)             (None, 200)               20200

_________________________________________________________________
dense_11 (Dense)             (None, 42)                8442
=================================================================
Total params: 76,548
Trainable params: 76,548
Non-trainable params: 0

_________________________________________________________________
```

[35]:
```
df_cleaned.drop("cluster", axis=1, errors="ignore", inplace=True)

early_stopping = EarlyStopping(monitor="val_loss", patience=10,␣
 ↪restore_best_weights=True, mode="min")

batch_size = 256
epochs = 1000
tf.random.set_seed(42)
history = autoencoder.fit(df_cleaned, df_cleaned,
                          epochs=epochs,
                          batch_size=batch_size,
                          shuffle=True,
                          validation_split=0.3,
                          callbacks=[early_stopping])
```

```
Epoch 1/1000
105/105 [==============================] - 7s 26ms/step - loss: 0.5983 -
val_loss: 0.4047
Epoch 2/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.2017 -
val_loss: 0.3214
Epoch 3/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.1866 -
val_loss: 0.3167
Epoch 4/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.1850 -
val_loss: 0.3102
Epoch 5/1000
105/105 [==============================] - 2s 17ms/step - loss: 0.1832 -
val_loss: 0.3084
```

```
Epoch 6/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.1809 -
val_loss: 0.3098
Epoch 7/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.1722 -
val_loss: 0.2901
Epoch 8/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.1538 -
val_loss: 0.2805
Epoch 9/1000
105/105 [==============================] - 1s 12ms/step - loss: 0.1424 -
val_loss: 0.2675
Epoch 10/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.1293 -
val_loss: 0.2493
Epoch 11/1000
105/105 [==============================] - 2s 21ms/step - loss: 0.1207 -
val_loss: 0.2412
Epoch 12/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.1174 -
val_loss: 0.2375
Epoch 13/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.1145 -
val_loss: 0.2326
Epoch 14/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.1109 -
val_loss: 0.2240
Epoch 15/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.1068 -
val_loss: 0.2168
Epoch 16/1000
105/105 [==============================] - 2s 17ms/step - loss: 0.1026 -
val_loss: 0.2088
Epoch 17/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0991 -
val_loss: 0.2018
Epoch 18/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0961 -
val_loss: 0.1953
Epoch 19/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.0934 -
val_loss: 0.1869
Epoch 20/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0910 -
val_loss: 0.1799
Epoch 21/1000
105/105 [==============================] - 2s 17ms/step - loss: 0.0891 -
val_loss: 0.1761
```

```
Epoch 22/1000
105/105 [==============================] - 2s 17ms/step - loss: 0.0875 -
val_loss: 0.1753
Epoch 23/1000
105/105 [==============================] - 2s 20ms/step - loss: 0.0861 -
val_loss: 0.1719
Epoch 24/1000
105/105 [==============================] - 2s 17ms/step - loss: 0.0848 -
val_loss: 0.1686
Epoch 25/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.0837 -
val_loss: 0.1664
Epoch 26/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0827 -
val_loss: 0.1629
Epoch 27/1000
105/105 [==============================] - 3s 27ms/step - loss: 0.0818 -
val_loss: 0.1632
Epoch 28/1000
105/105 [==============================] - 3s 29ms/step - loss: 0.0808 -
val_loss: 0.1602
Epoch 29/1000
105/105 [==============================] - 3s 26ms/step - loss: 0.0800 -
val_loss: 0.1605
Epoch 30/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0791 -
val_loss: 0.1586
Epoch 31/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0782 -
val_loss: 0.1563
Epoch 32/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.0775 -
val_loss: 0.1564
Epoch 33/1000
105/105 [==============================] - 3s 26ms/step - loss: 0.0768 -
val_loss: 0.1543
Epoch 34/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0762 -
val_loss: 0.1549
Epoch 35/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0755 -
val_loss: 0.1511
Epoch 36/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0750 -
val_loss: 0.1510
Epoch 37/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0746 -
val_loss: 0.1508
```

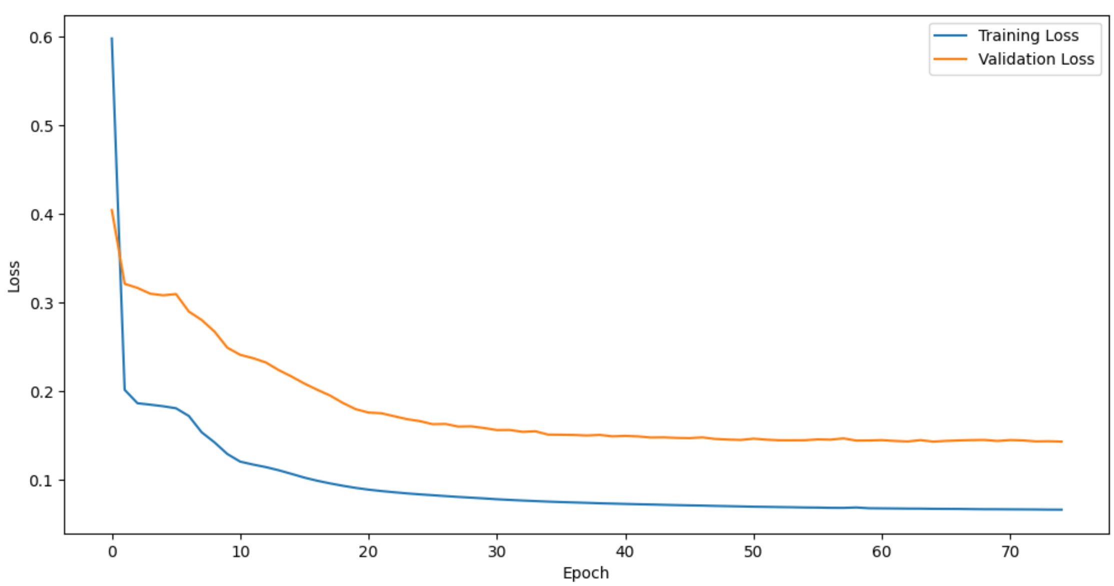
```
Epoch 38/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.0742 -
val_loss: 0.1502
Epoch 39/1000
105/105 [==============================] - 2s 16ms/step - loss: 0.0737 -
val_loss: 0.1509
Epoch 40/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0733 -
val_loss: 0.1492
Epoch 41/1000
105/105 [==============================] - 2s 20ms/step - loss: 0.0729 -
val_loss: 0.1497
Epoch 42/1000
105/105 [==============================] - 2s 21ms/step - loss: 0.0726 -
val_loss: 0.1491
Epoch 43/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0723 -
val_loss: 0.1479
Epoch 44/1000
105/105 [==============================] - 2s 20ms/step - loss: 0.0719 -
val_loss: 0.1481
Epoch 45/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0716 -
val_loss: 0.1475
Epoch 46/1000
105/105 [==============================] - 2s 22ms/step - loss: 0.0713 -
val_loss: 0.1472
Epoch 47/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0710 -
val_loss: 0.1481
Epoch 48/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0707 -
val_loss: 0.1463
Epoch 49/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0704 -
val_loss: 0.1456
Epoch 50/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0701 -
val_loss: 0.1452
Epoch 51/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0698 -
val_loss: 0.1466
Epoch 52/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0696 -
val_loss: 0.1455
Epoch 53/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0694 -
val_loss: 0.1448
```

```
Epoch 54/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0692 -
val_loss: 0.1448
Epoch 55/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0689 -
val_loss: 0.1448
Epoch 56/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0688 -
val_loss: 0.1458
Epoch 57/1000
105/105 [==============================] - 2s 22ms/step - loss: 0.0685 -
val_loss: 0.1455
Epoch 58/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0685 -
val_loss: 0.1470
Epoch 59/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0689 -
val_loss: 0.1444
Epoch 60/1000
105/105 [==============================] - 2s 24ms/step - loss: 0.0681 -
val_loss: 0.1445
Epoch 61/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0679 -
val_loss: 0.1449
Epoch 62/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0678 -
val_loss: 0.1440
Epoch 63/1000
105/105 [==============================] - 3s 25ms/step - loss: 0.0677 -
val_loss: 0.1434
Epoch 64/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0676 -
val_loss: 0.1450
Epoch 65/1000
105/105 [==============================] - 2s 20ms/step - loss: 0.0674 -
val_loss: 0.1432
Epoch 66/1000
105/105 [==============================] - 2s 18ms/step - loss: 0.0673 -
val_loss: 0.1441
Epoch 67/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0673 -
val_loss: 0.1446
Epoch 68/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0671 -
val_loss: 0.1450
Epoch 69/1000
105/105 [==============================] - 1s 14ms/step - loss: 0.0669 -
val_loss: 0.1452
```

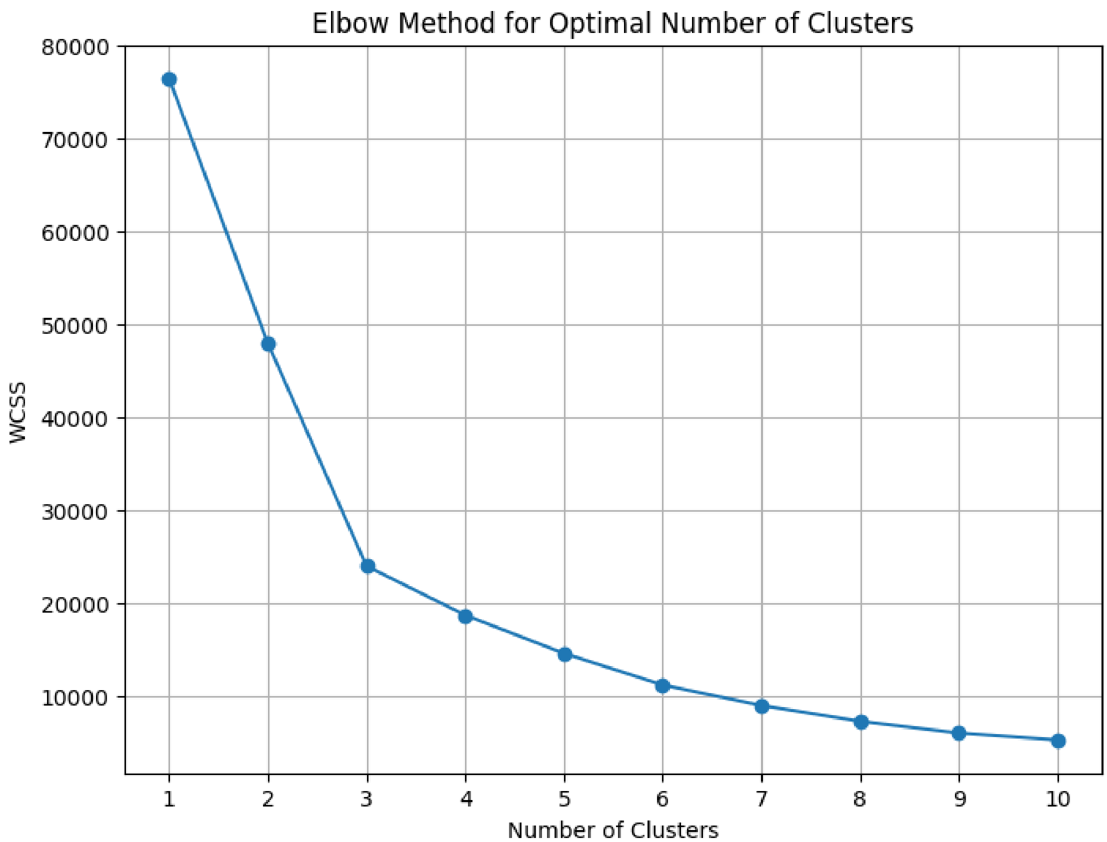
```
Epoch 70/1000
105/105 [==============================] - 1s 13ms/step - loss: 0.0669 -
val_loss: 0.1440
Epoch 71/1000
105/105 [==============================] - 2s 17ms/step - loss: 0.0668 -
val_loss: 0.1450
Epoch 72/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0668 -
val_loss: 0.1446
Epoch 73/1000
105/105 [==============================] - 2s 15ms/step - loss: 0.0667 -
val_loss: 0.1435
Epoch 74/1000
105/105 [==============================] - 2s 14ms/step - loss: 0.0665 -
val_loss: 0.1437
Epoch 75/1000
105/105 [==============================] - 2s 19ms/step - loss: 0.0665 -
val_loss: 0.1433
```

[36]:

```python
plt.figure(figsize=(12, 6))
plt.plot(history.history["loss"], label="Training Loss")
plt.plot(history.history["val_loss"], label="Validation Loss")
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.legend()
plt.show()
```



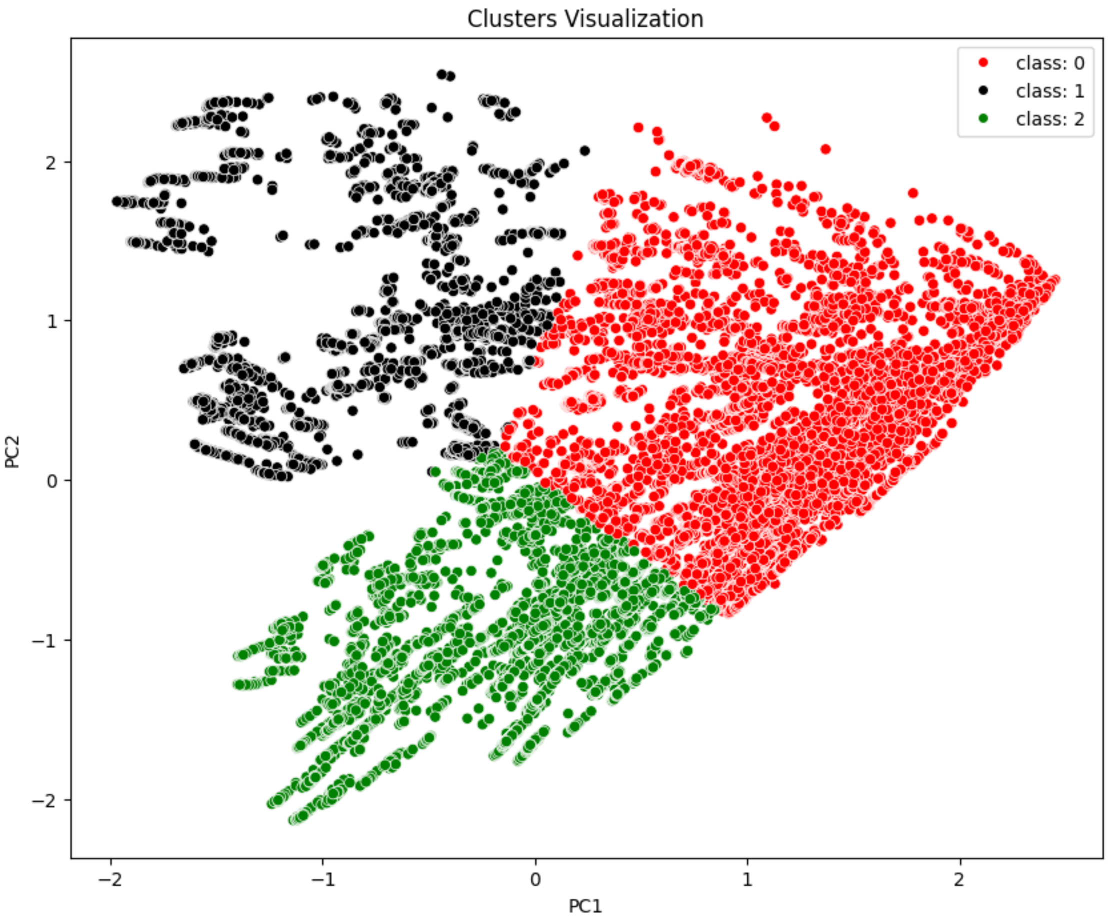
[37]:
```
df_cleaned.drop(["cluster","y", "duration"], axis=1, errors="ignore",␣
 ↪inplace=True)
encoded_data = encoder.predict(df_cleaned)
encoded_data = scaler(encoded_data)
pca = PCA(n_components=2, random_state=42, whiten=True)
encoded_data_pca = pca.fit_transform(encoded_data)

cluster_range = range(1, 11)
wcss = []

for n_clusters in cluster_range:
    kmeans = KMeans(n_clusters=n_clusters, random_state=42, init="k-means++",␣
 ↪n_init=10)
    kmeans.fit(encoded_data_pca)
    wcss.append(kmeans.inertia_)

plt.figure(figsize=(8, 6))
plt.plot(cluster_range, wcss, marker="o")
plt.xlabel("Number of Clusters")
plt.ylabel("WCSS")
plt.title("Elbow Method for Optimal Number of Clusters")
plt.gca().set_xticks(cluster_range, cluster_range)
plt.grid(True)
plt.show()
```



[38]:
```
pca = PCA(n_components=2, random_state=42, whiten=True)
encoded_data_pca = pca.fit_transform(encoded_data)

kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
clusters = kmeans.fit_predict(encoded_data_pca)

df_cleaned["cluster"] = clusters

df_viz = pd.DataFrame(encoded_data_pca, columns=["PC1", "PC2"])
df_viz["cluster"] = clusters
```

[49]:
```
handles = [Line2D([0], [0], markerfacecolor="r", lw=0, label="class: 0",␣
 ↪marker="o", color="w"),
           Line2D([0], [0], markerfacecolor="k", lw=0, label="class: 1",␣
 ↪marker="o", color="w"),
           Line2D([0], [0], markerfacecolor="g", lw=0, label="class: 2",␣
 ↪marker="o", color="w")
          ]

plt.figure(figsize=(10, 8))
```

```
labels = ["red", "black", "green"]
colors = pd.cut(df_viz["cluster"], bins=[-.01,0,1,2], labels=labels)
sns.scatterplot(data=df_viz, x="PC1", y="PC2", c=colors)
plt.title("Clusters Visualization")
plt.legend(handles=handles)
plt.show()
```



[48]:
```
# Evaluate clustering
silhouette_avg = silhouette_score(encoded_data_pca, clusters)
print(f"Silhouette Score: {silhouette_avg}")
```

```
Silhouette Score: 0.47355520725250244
```

Silhouette Score: suggests that your clusters are **moderately separated**

[41]:
```
df_cleaned[["y", "duration"]] = df[["y", "duration"]]
y_true = df_cleaned["y"]
clusters = df_cleaned["cluster"]
crosstab = pd.crosstab(y_true, clusters)
```

```
display((crosstab.T/ crosstab.values.sum(axis=1)).T)

purity = np.amax(crosstab.values, axis=0).sum() / crosstab.sum().sum()

print(f"Cluster Purity: {purity:.4f}")
```

```
cluster         0         1         2
y
False    0.337277  0.278764  0.383959
True     0.711064  0.105238  0.183697
```

```
Cluster Purity: 0.8887
```

[42]:
```
df_cleaned.groupby("cluster").agg(["mean", "median", "std"])
```

[42]:
```
              age                        campaign                          pdays  \
             mean    median       std       mean   median       std       mean
cluster
0        0.289336  0.259259  0.147990  0.026689  0.02381  0.043233  0.906334
1        0.303416  0.296296  0.110042  0.043003  0.02381  0.075864  0.999900
2        0.259573  0.234568  0.109969  0.044352  0.02381  0.075703  1.000000

                            previous                  emp_var_rate            \
        median       std        mean median       std         mean    median
cluster
0          1.0  0.290415  0.060206    0.0  0.100467     0.377731  0.333333
1          1.0  0.009980  0.001858    0.0  0.017292     0.939602  1.000000
2          1.0  0.000000  0.002769    0.0  0.020789     0.936527  1.000000

                    cons_price_idx                      cons_conf_idx            \
              std             mean    median       std          mean    median
cluster
0        0.232465         0.341346  0.269680  0.187531      0.346957  0.192469
1        0.134114         0.693088  0.698753  0.146470      0.490692  0.602510
2        0.143581         0.620773  0.669135  0.150677      0.471295  0.376569

                    euribor3m                     nr_employed            \
              std        mean    median       std        mean    median
cluster
0        0.242141  0.259956  0.150759  0.291139    0.510931  0.512287
1        0.130989  0.935194  0.958966  0.155279    0.915782  1.000000
2        0.134699  0.930596  0.980957  0.158064    0.938573  1.000000

                    marital_married                   marital_single            \
              std             mean median       std            mean median
cluster
0        0.241672         0.597266    1.0  0.490465        0.290142    0.0
1        0.124772         0.996471    1.0  0.059300        0.000000    0.0
```

```
2        0.131128        0.335503    0.0  0.472183       0.473928    0.0

                 housing_True                 contact_telephone         \
             std          mean median       std              mean median
cluster
0        0.453844     0.570689    1.0  0.494995          0.041833    0.0
1        0.000000     0.433310    0.0  0.495557          0.764896    1.0
2        0.499338     0.579084    1.0  0.493724          0.405583    0.0

                 poutcome_success                  \
             std              mean median       std
cluster
0        0.200215          0.085807    0.0  0.280088
1        0.424085          0.000000    0.0  0.000000
2        0.491022          0.000000    0.0  0.000000

        education_job_basic.4y_blue-collar                   \
                                      mean median       std
cluster
0                                 0.043352    0.0  0.203656
1                                 0.130558    0.0  0.336933
2                                 0.023794    0.0  0.152412

        education_job_basic.4y_entrepreneur                   \
                                       mean median       std
cluster
0                                  0.005315    0.0  0.072716
1                                  0.005444    0.0  0.073587
2                                  0.000072    0.0  0.008504

        education_job_basic.4y_retired                   \
                                  mean median       std
cluster
0                             0.031479    0.0  0.174613
1                             0.012098    0.0  0.109329
2                             0.000362    0.0  0.019013

        education_job_basic.4y_self-employed                   \
                                        mean median       std
cluster
0                                   0.003245    0.0  0.056870
1                                   0.003428    0.0  0.058450
2                                   0.000362    0.0  0.019013

        education_job_basic.4y_services                   \
                                   mean median       std
cluster
```

```
0                             0.006075    0.0  0.077707
1                             0.002722    0.0  0.052105
2                             0.000940    0.0  0.030649

        education_job_basic.4y_student                  \
                                 mean median       std
cluster
0                            0.001174    0.0  0.034238
1                            0.000000    0.0  0.000000
2                            0.000579    0.0  0.024048

        education_job_basic.6y_admin.                   \
                                mean median       std
cluster
0                           0.006006    0.0  0.077267
1                           0.004436    0.0  0.066458
2                           0.001229    0.0  0.035044

        education_job_basic.6y_blue-collar                  \
                                     mean median       std
cluster
0                                0.034999    0.0  0.183784
1                                0.051013    0.0  0.220036
2                                0.026832    0.0  0.161597

        education_job_basic.6y_student                  \
                                 mean median       std
cluster
0                            0.000897    0.0  0.029945
1                            0.000000    0.0  0.000000
2                            0.000000    0.0  0.000000

        education_job_basic.9y_admin.                   \
                                mean median       std
cluster
0                           0.017741    0.0  0.132014
1                           0.012199    0.0  0.109778
2                           0.008100    0.0  0.089639

        education_job_basic.9y_blue-collar                  \
                                     mean median       std
cluster
0                                0.076902    0.0  0.266445
1                                0.177437    0.0  0.382058
2                                0.046793    0.0  0.211202

        education_job_basic.9y_entrepreneur                  \
```

```
                                        mean median       std
cluster
0                                   0.008284    0.0  0.090641
1                                   0.007965    0.0  0.088892
2                                   0.000579    0.0  0.024048

        education_job_basic.9y_housemaid                   \
                                     mean median       std
cluster
0                                0.004280    0.0  0.065284
1                                0.002520    0.0  0.050143
2                                0.000362    0.0  0.019013

        education_job_basic.9y_services                   \
                                    mean median       std
cluster
0                               0.011874    0.0  0.108321
1                               0.011291    0.0  0.105665
2                               0.006726    0.0  0.081739

        education_job_basic.9y_student                   \
                                   mean median       std
cluster
0                              0.005661    0.0  0.075027
1                              0.000000    0.0  0.000000
2                              0.000868    0.0  0.029448

        education_job_high.school_entrepreneur                   \
                                           mean median       std
cluster
0                                      0.009181    0.0  0.095381
1                                      0.006351    0.0  0.079446
2                                      0.002242    0.0  0.047298

        education_job_high.school_management                   \
                                         mean median       std
cluster
0                                    0.013185    0.0  0.114071
1                                    0.004638    0.0  0.067945
2                                    0.003905    0.0  0.062373

        education_job_high.school_retired                   \
                                      mean median       std
cluster
0                                 0.013944    0.0  0.117265
1                                 0.005142    0.0  0.071524
2                                 0.001013    0.0  0.031805
```

```
        education_job_high.school_services                  \
                                      mean median       std
cluster
0                                 0.060748    0.0  0.238876
1                                 0.069866    0.0  0.254934
2                                 0.075287    0.0  0.263864

        education_job_high.school_student                  \
                                     mean median       std
cluster
0                                0.020088    0.0  0.140307
1                                0.000101    0.0  0.010041
2                                0.004122    0.0  0.064076

        education_job_illiterate_retired                  \
                                    mean median      std
cluster
0                               0.000207    0.0  0.01439
1                               0.000000    0.0  0.00000
2                               0.000000    0.0  0.00000

        education_job_professional.course_retired                  \
                                             mean median       std
cluster
0                                        0.013461    0.0  0.115243
1                                        0.003226    0.0  0.056710
2                                        0.000579    0.0  0.024048

        education_job_professional.course_student                  \
                                             mean median       std
cluster
0                                        0.002692    0.0  0.051819
1                                        0.000000    0.0  0.000000
2                                        0.000217    0.0  0.014729

        education_job_university.degree_admin.                  \
                                          mean median       std
cluster
0                                     0.118252    0.0  0.322918
1                                     0.000000    0.0  0.000000
2                                     0.282491    0.0  0.450227

        education_job_university.degree_management                  \
                                              mean median       std
cluster
0                                         0.067997    0.0  0.251749
```

```
1                                          0.000000    0.0   0.000000
2                                          0.074058    0.0   0.261875

        education_job_university.degree_retired                   \
                                      mean median       std
cluster
0                                      0.016844    0.0   0.128691
1                                      0.003226    0.0   0.056710
2                                      0.000072    0.0   0.008504

        education_job_university.degree_student                   \
                                      mean median       std
cluster
0                                      0.007939    0.0   0.088748
1                                      0.000000    0.0   0.000000
2                                      0.003616    0.0   0.060027

        education_job_university.degree_unemployed                          y  \
                                         mean median       std      mean
cluster
0                                         0.012633    0.0   0.111688  0.208960
1                                         0.003629    0.0   0.060138  0.045166
2                                         0.002459    0.0   0.049529  0.056556

                           duration
        median       std        mean median          std
cluster
0          0.0   0.406580  268.910327  195.0   254.041490
1          0.0   0.207678  253.818933  176.0   264.035590
2          0.0   0.231001  250.245606  167.0   262.392689
```

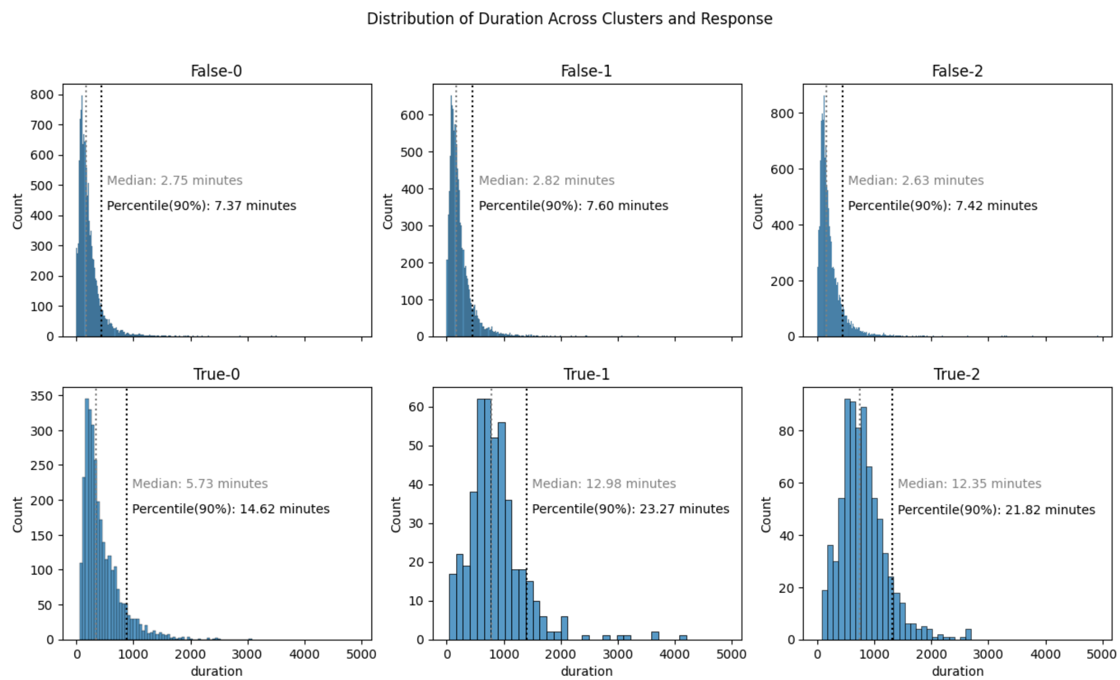
[43]:

```
df = df.assign(cluster=clusters)
```

[44]:

```
groups = df.groupby(["y", "cluster"])["duration"]

fig, axes = plt.subplots(2, 3, figsize=(15,8), sharex=True)
axes = axes.flatten()
i=0
for lab, group in groups:
    ax=axes[i]
    sns.histplot(group, ax=ax)
    median = np.median(group).astype(int)
    percentile90 = np.quantile(group,.9).astype(int)
    ax.axvline(median, linestyle="dotted", color="gray")
    ax.axvline(percentile90, linestyle="dotted", color="black")
    ax.annotate(f"Median: {median/60:.2f} minutes",
```

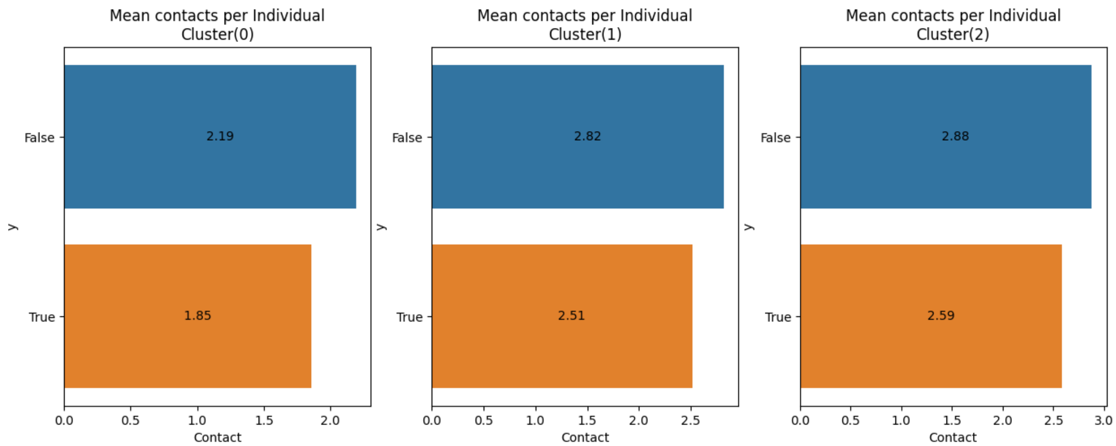
```
                    (median, ax.get_ylim()[1]*.6), color="gray", xytext=(percentile90+100, ax.get_ylim()[1]*.6))
    ax.annotate(f"\nPercentile(90%): {percentile90/60:.2f} minutes",
                    (median, ax.get_ylim()[1]*.6), color="black", xytext=(percentile90+100, ax.get_ylim()[1]*.5))
    ax.set_title(f"{lab[0]}-{lab[1]}")
    i+=1
plt.suptitle("Distribution of Duration Across Clusters and Response")
plt.show()
```



- The duration of contact is similar across all clusters for clients who did not subscribe.
- For subscribing clients, the contact duration in cluster 0 follows an exponential distribution, with half of the contacts under 6 minutes and 90% under 15 minutes.
- In cluster 1, half of the contacts are under 13 minutes and 90% are under 24 minutes.
- In cluster 2, half of the contacts are under 13 minutes and 90% are under 22 minutes.

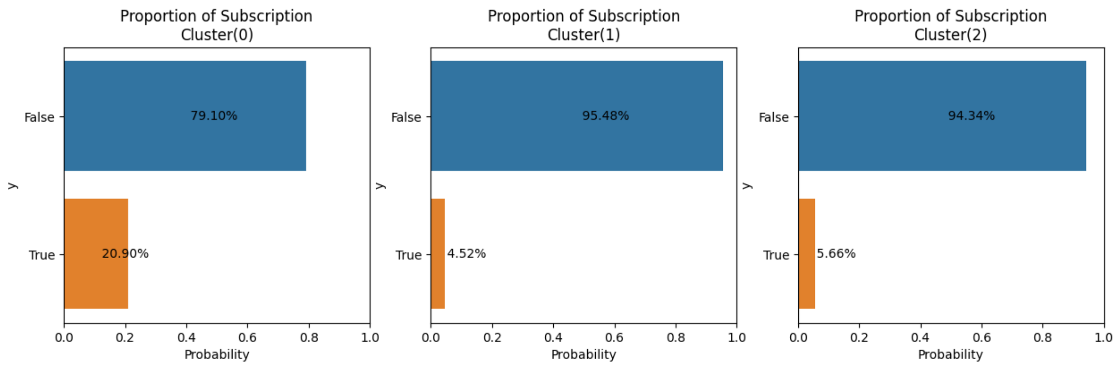
[45]:
```
plt.figure(figsize=(15,4))
i=1
mean_contacts = df.groupby(["cluster", "y"])["campaign"].mean().unstack()
for j in range(3):
    ax = plt.subplot(1,3,i)
    sns.barplot(y=mean_contacts.loc[j].index.astype(str), x=mean_contacts.loc[j].values)
    ax.bar_label(ax.containers[0], mean_contacts.loc[j].map(lambda x: f"{x:.2f}"), label_type ="center")
```

```
    ax.set_xlabel(f"Contact")
    ax.set_title(f"Mean contacts per Individual\nCluster({j})")
    i+=1
plt.subplots_adjust(bottom=1, top=2)
plt.show()
```



- For clients who subscribed, cluster 0 requires fewer contacts compared to cluster 2, and cluster 1 requires fewer contacts than cluster 2.

[46]:
```
plt.figure(figsize=(15,4))
i=1
proportions = df.groupby("cluster")["y"].value_counts(normalize=True).unstack()
for j in range(3):
    ax = plt.subplot(1,3,i)
    sns.barplot(y=proportions.loc[j].index.astype(str), x=proportions.loc[j].
↪values)
    ax.bar_label(ax.containers[0], proportions.loc[j].map(lambda x: f"
↪    {x:.2%}"), label_type ="center")
    ax.set_xlabel(f"Probability")
    ax.set_title(f"Proportion of Subscription\nCluster({j})")
    ax.set_xlim(0,1)
    i+=1

plt.show()
```



- The probability of subscribing to a term deposit is 20.9% in cluster 0, 4.5% in cluster 1, and 5.7% in cluster 2.
- Cluster 0 has the highest probability of subscribing to a term deposit, making it the most promising cluster for conversions. In contrast, cluster 1 has the lowest probability, indicating it is the least effective for term deposit subscriptions.

[47]:
```
autoencoder.save("final_model.h5")
```

0.8 Summary Report

Cluster Characteristics:

1. Cluster 0:
   - Duration of Contact: For subscribing clients, contact durations are typically shorter, with a median duration of 6 minutes and 90% of contacts under 15 minutes.
   - Mean Contacts: Fewer contacts are required to achieve a subscription compared to other clusters.
   - Subscription Probability: Highest probability of subscribing to a term deposit at 20.9%.
2. Cluster 1:
   - Duration of Contact: Contacts in this cluster have a median duration of 13 minutes and 90% are under 24 minutes.
   - Mean Contacts: Requires more contacts compared to Cluster 0 but fewer than Cluster 2.
   - Subscription Probability: Lowest probability of subscribing to a term deposit at 4.5%.
3. Cluster 2:
   - Duration of Contact: Median duration of 13 minutes and 90% of contacts are under 22 minutes.
   - Mean Contacts: Requires more contacts than Cluster 0 and 1.
   - Subscription Probability: Moderate probability of subscribing to a term deposit at 5.6%.

Key Insights:

1. Cluster 0 is the most promising for term deposit subscriptions due to its highest subscription probability and fewer required contacts.

2. Cluster 1 represents the least effective segment for conversions, indicated by the lowest probability and highest contact duration and frequency.
3. Cluster 2 falls in between, with moderate probabilities and contact requirements.

0.8.1 Recommendations

Targeted Marketing:

1. Focus on Cluster 0: Implement targeted campaigns with personalized offers for clients in this cluster to maximize conversion rates.
   - Use shorter contact durations as a benchmark for effectiveness.
2. Revise Strategies for Cluster 1: Given the low conversion rates and higher contact requirements, reassess the marketing strategies for this segment. Consider testing different approaches or reducing contact attempts.
3. Optimize Efforts for Cluster 2: Apply strategies that balance between cost and effectiveness. Tailor marketing efforts to improve the subscription probability without excessive resource expenditure.