creditCardsDefault

June 25, 2021

1 Setup

[2]:
```
!pip install wget
!pip install graphviz
!pip install thundersvm
import wget
import urllib
import os
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import graphviz
import keras
from keras.models import Sequential
from sklearn.dummy import DummyClassifier
from keras.wrappers.scikit_learn import KerasClassifier
from keras.layers import Dense, Input, Dropout
from keras.optimizers import Adam
from keras.initializers import glorot_uniform, Orthogonal, RandomNormal
from sklearn.metrics import f1_score, confusion_matrix, mean_squared_error
from sklearn.model_selection import train_test_split, GridSearchCV,␣
↪StratifiedKFold
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn import tree
from xgboost import XGBClassifier
import tensorflow as tf
from keras.callbacks import Callback,ModelCheckpoint
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
warnings.filterwarnings("ignore", category=DeprecationWarning)

## Constants
randSeed = 42
nSplitsGS = 3
```

```
confidenceLevel = 0.95

## Utility
models = {}
skf = StratifiedKFold(n_splits=nSplitsGS, shuffle=True, random_state=randSeed)
```

```
Processing /root/.cache/pip/wheels/40/15/30/7d8f7cea2902b4db79e3fea550d7d7b85ecb
27ef992b618f3f/wget-3.2-cp37-none-any.whl
Installing collected packages: wget
Successfully installed wget-3.2
Requirement already satisfied: graphviz in /usr/local/lib/python3.7/dist-
packages (0.10.1)
Collecting thundersvm
  Downloading https://files.pythonhosted.org/packages/7d/16/281a54f6d1f70c
59df242f2f93e5cc04daf01b9c9809c2b154d15ea6a346/thundersvm-0.3.12-py3-none-
any.whl (507kB)
     |                  | 512kB 11.0MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-
packages (from thundersvm) (1.19.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages
(from thundersvm) (1.4.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-
packages (from thundersvm) (0.22.2.post1)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-
packages (from scikit-learn->thundersvm) (1.0.1)
Installing collected packages: thundersvm
Successfully installed thundersvm-0.3.12
```

[3]:
```
url = 'https://raw.githubusercontent.com/DTrimarchi10/confusion_matrix/master/
↪cf_matrix.py'
filename = 'cf_matrix.py'

if not os.path.isfile(filename):
    wget.download(url)

sys.path.append(os.path.abspath("./cf_matrix.py"))
from cf_matrix import make_confusion_matrix
```

[4]:
```
!wget https://developer.nvidia.com/compute/cuda/9.0/Prod/local_installers/
↪cuda-repo-ubuntu1704-9-0-local_9.0.176-1_amd64-deb
!ls  # Check if required cuda 9.0 amd64-deb file is downloaded
!dpkg -i cuda-repo-ubuntu1704-9-0-local_9.0.176-1_amd64-deb
!ls /var/cuda-repo-9-0-local | grep .pub
!apt-key add /var/cuda-repo-9-0-local/7fa2af80.pub
!apt-get update
!sudo apt-get install cuda-9.0
```

```
--2021-06-25 15:22:30--
```

```
https://developer.nvidia.com/compute/cuda/9.0/Prod/local_installers/cuda-repo-
ubuntu1704-9-0-local_9.0.176-1_amd64-deb
Resolving developer.nvidia.com (developer.nvidia.com)… 152.199.16.29
Connecting to developer.nvidia.com (developer.nvidia.com)|152.199.16.29|:443…
connected.
HTTP request sent, awaiting response… 301 Moved Permanently
Location:
https://developer.nvidia.com/compute/cuda/9.0/prod/local_installers/cuda-repo-
ubuntu1704-9-0-local_9.0.176-1_amd64-deb [following]
--2021-06-25 15:22:30--
https://developer.nvidia.com/compute/cuda/9.0/prod/local_installers/cuda-repo-
ubuntu1704-9-0-local_9.0.176-1_amd64-deb
Reusing existing connection to developer.nvidia.com:443.
HTTP request sent, awaiting response… 302 Found
Location: https://developer.download.nvidia.com/compute/cuda/9.0/secure/Prod/loc
al_installers/cuda-repo-
ubuntu1704-9-0-local_9.0.176-1_amd64.deb?cCTPIaj0PUNEdygzF-dhsEsQ1VBDTaUqrZkaDOe
5fPdcnIB7vx-2PxrZM6OMDL6KB9qZ1vA5-Zv89N2WHiQ7jwOoE_1XqdlQqxviOlMHxONQh3ZiM1b02GZ
F8aeo-MuWicqYDZxR66-4zMBYboT5Z_kF0Lliun_jJPCjZUkkXegUrBsr3Gv7niUkgJ3HfxqdfEcaUfp
jFTgCw63qUeSB [following]
--2021-06-25 15:22:31--  https://developer.download.nvidia.com/compute/cuda/9.0/
secure/Prod/local_installers/cuda-repo-
ubuntu1704-9-0-local_9.0.176-1_amd64.deb?cCTPIaj0PUNEdygzF-dhsEsQ1VBDTaUqrZkaDOe
5fPdcnIB7vx-2PxrZM6OMDL6KB9qZ1vA5-Zv89N2WHiQ7jwOoE_1XqdlQqxviOlMHxONQh3ZiM1b02GZ
F8aeo-MuWicqYDZxR66-4zMBYboT5Z_kF0Lliun_jJPCjZUkkXegUrBsr3Gv7niUkgJ3HfxqdfEcaUfp
jFTgCw63qUeSB
Resolving developer.download.nvidia.com (developer.download.nvidia.com)…
152.199.20.126
Connecting to developer.download.nvidia.com
(developer.download.nvidia.com)|152.199.20.126|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 1216133170 (1.1G) [application/x-deb]
Saving to: ‘cuda-repo-ubuntu1704-9-0-local_9.0.176-1_amd64-deb’

cuda-repo-ubuntu170 100%[===================>]   1.13G   161MB/s    in 7.2s

2021-06-25 15:22:38 (161 MB/s) - ‘cuda-repo-
ubuntu1704-9-0-local_9.0.176-1_amd64-deb’ saved [1216133170/1216133170]

cf_matrix.py                                        __pycache__
cuda-repo-ubuntu1704-9-0-local_9.0.176-1_amd64-deb  sample_data
Selecting previously unselected package cuda-repo-ubuntu1704-9-0-local.
(Reading database … 160772 files and directories currently installed.)
Preparing to unpack cuda-repo-ubuntu1704-9-0-local_9.0.176-1_amd64-deb …
Unpacking cuda-repo-ubuntu1704-9-0-local (9.0.176-1) …
Setting up cuda-repo-ubuntu1704-9-0-local (9.0.176-1) …
7fa2af80.pub
OK
```

```
Get:1 file:/var/cuda-repo-9-0-local  InRelease
Ign:1 file:/var/cuda-repo-9-0-local  InRelease
Get:2 file:/var/cuda-repo-9-0-local  Release [574 B]
Get:2 file:/var/cuda-repo-9-0-local  Release [574 B]
Get:3 file:/var/cuda-repo-9-0-local  Release.gpg [819 B]
Get:3 file:/var/cuda-repo-9-0-local  Release.gpg [819 B]
Hit:4 http://archive.ubuntu.com/ubuntu bionic InRelease
Hit:5 http://ppa.launchpad.net/c2d4u.team/c2d4u4.0+/ubuntu bionic InRelease
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Hit:7 http://ppa.launchpad.net/cran/libgit2/ubuntu bionic InRelease
Hit:8 https://cloud.r-project.org/bin/linux/ubuntu bionic-cran40/ InRelease
Get:9 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic InRelease [15.9 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:11 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic InRelease
[21.3 kB]
Get:12 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Ign:13
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
InRelease
Get:14 file:/var/cuda-repo-9-0-local  Packages [15.8 kB]
Ign:15 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  InRelease
Get:16
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Release [697 B]
Hit:17 https://developer.download.nvidia.com/compute/machine-
learning/repos/ubuntu1804/x86_64  Release
Get:18
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Release.gpg [836 B]
Get:19 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages
[2,652 kB]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages
[2,188 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages
[505 kB]
Get:22 http://ppa.launchpad.net/deadsnakes/ppa/ubuntu bionic/main amd64 Packages
[40.9 kB]
Get:23 http://ppa.launchpad.net/graphics-drivers/ppa/ubuntu bionic/main amd64
Packages [41.5 kB]
Get:24 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64
Packages [473 kB]
Get:26 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages
[1,418 kB]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages
[2,220 kB]
Ign:28
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
```

```
Packages
Get:28
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64
Packages [599 kB]
Fetched 10.4 MB in 3s (3,161 kB/s)
Reading package lists… Done
Reading package lists… Done
Building dependency tree
Reading state information… Done
Note, selecting 'cuda-9-0' for regex 'cuda-9.0'
Note, selecting 'libcuda-9.0-1' for regex 'cuda-9.0'
The following additional packages will be installed:
  cuda-command-line-tools-9-0 cuda-core-9-0 cuda-cublas-9-0
  cuda-cublas-dev-9-0 cuda-cudart-9-0 cuda-cudart-dev-9-0 cuda-cufft-9-0
  cuda-cufft-dev-9-0 cuda-curand-9-0 cuda-curand-dev-9-0 cuda-cusolver-9-0
  cuda-cusolver-dev-9-0 cuda-cusparse-9-0 cuda-cusparse-dev-9-0
  cuda-demo-suite-9-0 cuda-documentation-9-0 cuda-driver-dev-9-0
  cuda-libraries-9-0 cuda-libraries-dev-9-0 cuda-license-9-0
  cuda-misc-headers-9-0 cuda-npp-9-0 cuda-npp-dev-9-0 cuda-nvgraph-9-0
  cuda-nvgraph-dev-9-0 cuda-nvml-dev-9-0 cuda-nvrtc-9-0 cuda-nvrtc-dev-9-0
  cuda-runtime-9-0 cuda-samples-9-0 cuda-toolkit-9-0 cuda-visual-tools-9-0
The following NEW packages will be installed:
  cuda-9-0 cuda-command-line-tools-9-0 cuda-core-9-0 cuda-cublas-9-0
  cuda-cublas-dev-9-0 cuda-cudart-9-0 cuda-cudart-dev-9-0 cuda-cufft-9-0
  cuda-cufft-dev-9-0 cuda-curand-9-0 cuda-curand-dev-9-0 cuda-cusolver-9-0
  cuda-cusolver-dev-9-0 cuda-cusparse-9-0 cuda-cusparse-dev-9-0
  cuda-demo-suite-9-0 cuda-documentation-9-0 cuda-driver-dev-9-0
  cuda-libraries-9-0 cuda-libraries-dev-9-0 cuda-license-9-0
  cuda-misc-headers-9-0 cuda-npp-9-0 cuda-npp-dev-9-0 cuda-nvgraph-9-0
  cuda-nvgraph-dev-9-0 cuda-nvml-dev-9-0 cuda-nvrtc-9-0 cuda-nvrtc-dev-9-0
  cuda-runtime-9-0 cuda-samples-9-0 cuda-toolkit-9-0 cuda-visual-tools-9-0
0 upgraded, 33 newly installed, 0 to remove and 70 not upgraded.
Need to get 0 B/1,097 MB of archives.
After this operation, 2,315 MB of additional disk space will be used.
Get:1 file:/var/cuda-repo-9-0-local  cuda-license-9-0 9.0.176-1 [22.0 kB]
Get:2 file:/var/cuda-repo-9-0-local  cuda-misc-headers-9-0 9.0.176-1 [684 kB]
Get:3 file:/var/cuda-repo-9-0-local  cuda-core-9-0 9.0.176-1 [16.9 MB]
Get:4 file:/var/cuda-repo-9-0-local  cuda-cudart-9-0 9.0.176-1 [106 kB]
Get:5 file:/var/cuda-repo-9-0-local  cuda-driver-dev-9-0 9.0.176-1 [10.9 kB]
Get:6 file:/var/cuda-repo-9-0-local  cuda-cudart-dev-9-0 9.0.176-1 [767 kB]
Get:7 file:/var/cuda-repo-9-0-local  cuda-command-line-tools-9-0 9.0.176-1 [25.4
MB]
Get:8 file:/var/cuda-repo-9-0-local  cuda-nvrtc-9-0 9.0.176-1 [6,348 kB]
Get:9 file:/var/cuda-repo-9-0-local  cuda-nvrtc-dev-9-0 9.0.176-1 [9,334 B]
Get:10 file:/var/cuda-repo-9-0-local  cuda-cusolver-9-0 9.0.176-1 [26.2 MB]
Get:11 file:/var/cuda-repo-9-0-local  cuda-cusolver-dev-9-0 9.0.176-1 [5,317 kB]
Get:12 file:/var/cuda-repo-9-0-local  cuda-cublas-9-0 9.0.176-1 [25.0 MB]
Get:13 file:/var/cuda-repo-9-0-local  cuda-cublas-dev-9-0 9.0.176-1 [49.4 MB]
```

```
Get:14 file:/var/cuda-repo-9-0-local  cuda-cufft-9-0 9.0.176-1 [84.1 MB]
Get:15 file:/var/cuda-repo-9-0-local  cuda-cufft-dev-9-0 9.0.176-1 [73.7 MB]
Get:16 file:/var/cuda-repo-9-0-local  cuda-curand-9-0 9.0.176-1 [38.8 MB]
Get:17 file:/var/cuda-repo-9-0-local  cuda-curand-dev-9-0 9.0.176-1 [57.9 MB]
Get:18 file:/var/cuda-repo-9-0-local  cuda-cusparse-9-0 9.0.176-1 [25.2 MB]
Get:19 file:/var/cuda-repo-9-0-local  cuda-cusparse-dev-9-0 9.0.176-1 [25.3 MB]
Get:20 file:/var/cuda-repo-9-0-local  cuda-npp-9-0 9.0.176-1 [46.6 MB]
Get:21 file:/var/cuda-repo-9-0-local  cuda-npp-dev-9-0 9.0.176-1 [46.6 MB]
Get:22 file:/var/cuda-repo-9-0-local  cuda-nvgraph-9-0 9.0.176-1 [6,081 kB]
Get:23 file:/var/cuda-repo-9-0-local  cuda-nvgraph-dev-9-0 9.0.176-1 [5,658 kB]
Get:24 file:/var/cuda-repo-9-0-local  cuda-samples-9-0 9.0.176-1 [75.9 MB]
Get:25 file:/var/cuda-repo-9-0-local  cuda-documentation-9-0 9.0.176-1 [53.1 MB]
Get:26 file:/var/cuda-repo-9-0-local  cuda-libraries-dev-9-0 9.0.176-1 [2,596 B]
Get:27 file:/var/cuda-repo-9-0-local  cuda-nvml-dev-9-0 9.0.176-1 [47.6 kB]
Get:28 file:/var/cuda-repo-9-0-local  cuda-visual-tools-9-0 9.0.176-1 [398 MB]
Get:29 file:/var/cuda-repo-9-0-local  cuda-toolkit-9-0 9.0.176-1 [2,836 B]
Get:30 file:/var/cuda-repo-9-0-local  cuda-libraries-9-0 9.0.176-1 [2,566 B]
Get:31 file:/var/cuda-repo-9-0-local  cuda-runtime-9-0 9.0.176-1 [2,526 B]
Get:32 file:/var/cuda-repo-9-0-local  cuda-demo-suite-9-0 9.0.176-1 [3,880 kB]
Get:33 file:/var/cuda-repo-9-0-local  cuda-9-0 9.0.176-1 [2,552 B]
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 76,
<> line 33.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package cuda-license-9-0.
(Reading database … 160831 files and directories currently installed.)
Preparing to unpack …/00-cuda-license-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-license-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-misc-headers-9-0.
Preparing to unpack …/01-cuda-misc-headers-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-misc-headers-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-core-9-0.
Preparing to unpack …/02-cuda-core-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-core-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cudart-9-0.
Preparing to unpack …/03-cuda-cudart-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cudart-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-driver-dev-9-0.
Preparing to unpack …/04-cuda-driver-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-driver-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cudart-dev-9-0.
Preparing to unpack …/05-cuda-cudart-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cudart-dev-9-0 (9.0.176-1) …
```

```
Selecting previously unselected package cuda-command-line-tools-9-0.
Preparing to unpack …/06-cuda-command-line-tools-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-command-line-tools-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-nvrtc-9-0.
Preparing to unpack …/07-cuda-nvrtc-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-nvrtc-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-nvrtc-dev-9-0.
Preparing to unpack …/08-cuda-nvrtc-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-nvrtc-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cusolver-9-0.
Preparing to unpack …/09-cuda-cusolver-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cusolver-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cusolver-dev-9-0.
Preparing to unpack …/10-cuda-cusolver-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cusolver-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cublas-9-0.
Preparing to unpack …/11-cuda-cublas-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cublas-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cublas-dev-9-0.
Preparing to unpack …/12-cuda-cublas-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cublas-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cufft-9-0.
Preparing to unpack …/13-cuda-cufft-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cufft-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cufft-dev-9-0.
Preparing to unpack …/14-cuda-cufft-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cufft-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-curand-9-0.
Preparing to unpack …/15-cuda-curand-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-curand-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-curand-dev-9-0.
Preparing to unpack …/16-cuda-curand-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-curand-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cusparse-9-0.
Preparing to unpack …/17-cuda-cusparse-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cusparse-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-cusparse-dev-9-0.
Preparing to unpack …/18-cuda-cusparse-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-cusparse-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-npp-9-0.
Preparing to unpack …/19-cuda-npp-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-npp-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-npp-dev-9-0.
Preparing to unpack …/20-cuda-npp-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-npp-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-nvgraph-9-0.
Preparing to unpack …/21-cuda-nvgraph-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-nvgraph-9-0 (9.0.176-1) …
```

```
Selecting previously unselected package cuda-nvgraph-dev-9-0.
Preparing to unpack …/22-cuda-nvgraph-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-nvgraph-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-samples-9-0.
Preparing to unpack …/23-cuda-samples-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-samples-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-documentation-9-0.
Preparing to unpack …/24-cuda-documentation-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-documentation-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-libraries-dev-9-0.
Preparing to unpack …/25-cuda-libraries-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-libraries-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-nvml-dev-9-0.
Preparing to unpack …/26-cuda-nvml-dev-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-nvml-dev-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-visual-tools-9-0.
Preparing to unpack …/27-cuda-visual-tools-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-visual-tools-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-toolkit-9-0.
Preparing to unpack …/28-cuda-toolkit-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-toolkit-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-libraries-9-0.
Preparing to unpack …/29-cuda-libraries-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-libraries-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-runtime-9-0.
Preparing to unpack …/30-cuda-runtime-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-runtime-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-demo-suite-9-0.
Preparing to unpack …/31-cuda-demo-suite-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-demo-suite-9-0 (9.0.176-1) …
Selecting previously unselected package cuda-9-0.
Preparing to unpack …/32-cuda-9-0_9.0.176-1_amd64.deb …
Unpacking cuda-9-0 (9.0.176-1) …
Setting up cuda-license-9-0 (9.0.176-1) …
*** LICENSE AGREEMENT ***
By using this software you agree to fully comply with the terms and
conditions of the EULA (End User License Agreement). The EULA is located
at /usr/local/cuda-9.0/doc/EULA.txt. The EULA can also be found at
http://docs.nvidia.com/cuda/eula/index.html. If you do not agree to the
terms and conditions of the EULA, do not use the software.

Setting up cuda-cusparse-9-0 (9.0.176-1) …
Setting up cuda-cudart-9-0 (9.0.176-1) …
Setting up cuda-nvrtc-9-0 (9.0.176-1) …
Setting up cuda-cusparse-dev-9-0 (9.0.176-1) …
Setting up cuda-cufft-9-0 (9.0.176-1) …
Setting up cuda-cusolver-9-0 (9.0.176-1) …
Setting up cuda-nvml-dev-9-0 (9.0.176-1) …
```

```
Setting up cuda-npp-9-0 (9.0.176-1) …
Setting up cuda-cusolver-dev-9-0 (9.0.176-1) …
Setting up cuda-misc-headers-9-0 (9.0.176-1) …
Setting up cuda-cublas-9-0 (9.0.176-1) …
Setting up cuda-nvrtc-dev-9-0 (9.0.176-1) …
Setting up cuda-driver-dev-9-0 (9.0.176-1) …
Setting up cuda-curand-9-0 (9.0.176-1) …
Setting up cuda-nvgraph-9-0 (9.0.176-1) …
Setting up cuda-core-9-0 (9.0.176-1) …
Setting up cuda-libraries-9-0 (9.0.176-1) …
Setting up cuda-runtime-9-0 (9.0.176-1) …
Setting up cuda-cudart-dev-9-0 (9.0.176-1) …
Setting up cuda-cufft-dev-9-0 (9.0.176-1) …
Setting up cuda-npp-dev-9-0 (9.0.176-1) …
Setting up cuda-curand-dev-9-0 (9.0.176-1) …
Setting up cuda-cublas-dev-9-0 (9.0.176-1) …
Setting up cuda-nvgraph-dev-9-0 (9.0.176-1) …
Setting up cuda-command-line-tools-9-0 (9.0.176-1) …
Setting up cuda-demo-suite-9-0 (9.0.176-1) …
Setting up cuda-visual-tools-9-0 (9.0.176-1) …
Setting up cuda-samples-9-0 (9.0.176-1) …
Setting up cuda-libraries-dev-9-0 (9.0.176-1) …
Setting up cuda-documentation-9-0 (9.0.176-1) …
Setting up cuda-toolkit-9-0 (9.0.176-1) …
Setting up cuda-9-0 (9.0.176-1) …
Processing triggers for libc-bin (2.27-3ubuntu1.2) …
/sbin/ldconfig.real: /usr/local/lib/python3.7/dist-
packages/ideep4py/lib/libmkldnn.so.0 is not a symbolic link
```

2 Descrizione del problema

Il dataset scelto ha come scopo la classificazione binaria (solvente o non solvente) della carta di credito da parte dei clienti di istituti finanziari a Taiwan.

3 Data exploration

3.1 Data upload

Carico il dataset e imposto come indice la prima colonna (ID) e come header la prima riga.

[5]:
```
dataset = pd.read_csv('https://raw.githubusercontent.com/lorenzobalzani/
↪Credit-cards-default-ML/master/dataset.csv', sep=";", header=0, index_col=0)
dataset = dataset.rename(columns={'default payment next month' : 'default pay'})
```

[6]:
```
dataset.head()
```

[6]:
```
    LIMIT_BAL  SEX  EDUCATION  …  PAY_AMT5  PAY_AMT6  default pay
ID                             …
1       20000    2          2  …         0         0            1
2      120000    2          2  …         0      2000            1
3       90000    2          2  …      1000      5000            0
4       50000    2          2  …      1069      1000            0
5       50000    1          2  …       689       679            0

[5 rows x 24 columns]
```

[7]:
```
dataset.describe()
```

[7]:
```
            LIMIT_BAL           SEX  …       PAY_AMT6   default pay
count    30000.000000  30000.000000  …   30000.000000  30000.000000
mean    167484.322667      1.603733  …    5215.502567      0.221200
std     129747.661567      0.489129  …   17777.465775      0.415062
min      10000.000000      1.000000  …       0.000000      0.000000
25%      50000.000000      1.000000  …     117.750000      0.000000
50%     140000.000000      2.000000  …    1500.000000      0.000000
75%     240000.000000      2.000000  …    4000.000000      0.000000
max    1000000.000000      2.000000  …  528666.000000      1.000000

[8 rows x 24 columns]
```

[8]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30000 entries, 1 to 30000
Data columns (total 24 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   LIMIT_BAL    30000 non-null  int64
 1   SEX          30000 non-null  int64
 2   EDUCATION    30000 non-null  int64
 3   MARRIAGE     30000 non-null  int64
 4   AGE          30000 non-null  int64
 5   PAY_1        30000 non-null  int64
 6   PAY_2        30000 non-null  int64
 7   PAY_3        30000 non-null  int64
 8   PAY_4        30000 non-null  int64
 9   PAY_5        30000 non-null  int64
 10  PAY_6        30000 non-null  int64
 11  BILL_AMT1    30000 non-null  int64
 12  BILL_AMT2    30000 non-null  int64
 13  BILL_AMT3    30000 non-null  int64
 14  BILL_AMT4    30000 non-null  int64
 15  BILL_AMT5    30000 non-null  int64
 16  BILL_AMT6    30000 non-null  int64
```

```
 17  PAY_AMT1     30000 non-null  int64
 18  PAY_AMT2     30000 non-null  int64
 19  PAY_AMT3     30000 non-null  int64
 20  PAY_AMT4     30000 non-null  int64
 21  PAY_AMT5     30000 non-null  int64
 22  PAY_AMT6     30000 non-null  int64
 23  default pay  30000 non-null  int64
dtypes: int64(24)
memory usage: 5.7 MB
```

3.2 Rilevazione di valori nulli

Si rilevano, se presenti, valori nulli nel dataset.
In questo dataset, non sono presenti valori nulli, quindi non risulta necessario trattarli.

[9]:
```
dataset.isna().sum()
```

[9]:
```
LIMIT_BAL      0
SEX            0
EDUCATION      0
MARRIAGE       0
AGE            0
PAY_1          0
PAY_2          0
PAY_3          0
PAY_4          0
PAY_5          0
PAY_6          0
BILL_AMT1      0
BILL_AMT2      0
BILL_AMT3      0
BILL_AMT4      0
BILL_AMT5      0
BILL_AMT6      0
PAY_AMT1       0
PAY_AMT2       0
PAY_AMT3       0
PAY_AMT4       0
PAY_AMT5       0
PAY_AMT6       0
default pay    0
dtype: int64
```

3.3 Comprensione delle features

Nello specifico, le **features** disponibili (come si può osservare nella cella soprastante) sono:

1. LIMIT_BAL | Credito totale (indivudale+familiare) disponibile | int (New Taiwan dollar)
2. SEX | 1: male, 2: female |

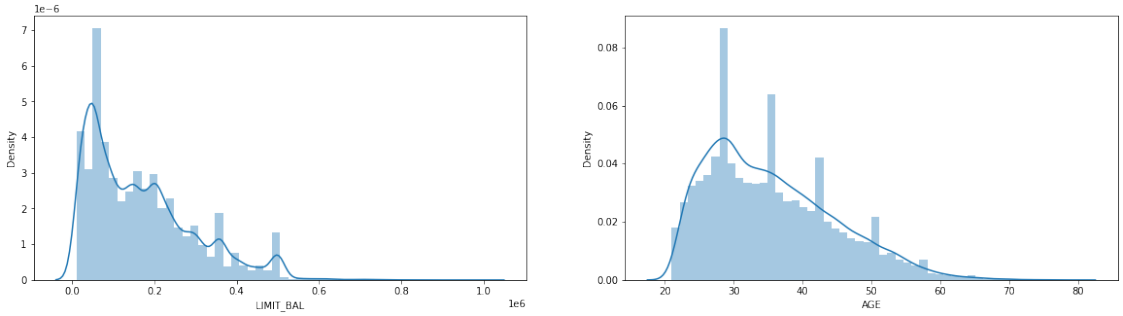
3. EDUCATION | 1: graduate school, 2: university, 3: high school, 4: others |
4. MARRIAGE | 1: married, 2: single, 3: others |
5. AGE | age | int (years)
6. PAY_1 | Repayment status in September, 2005 | view description below
7. PAY_2 | Repayment status in August, 2005 | view description below
8. PAY_3 | Repayment status in July, 2005 | view description below
9. PAY_4 | Repayment status in June, 2005 | view description below
10. PAY_5 | Repayment status in May, 2005 | view description below
11. PAY_6 | Repayment status in April, 2005 | view description below
12. BILL_AMT1 | Amout of bill statement in September, 2005 | int (New Taiwan dollar)
13. BILL_AMT2 | Amout of bill statement in August, 2005 | int (New Taiwan dollar)
14. BILL_AMT3 | Amout of bill statement in July, 2005 | int (New Taiwan dollar)
15. BILL_AMT4 | Amout of bill statement in June, 2005 | int (New Taiwan dollar)
16. BILL_AMT5 | Amout of bill statement in May, 2005 | int (New Taiwan dollar)
17. BILL_AMT6 | Amout of bill statement in April, 2005 | int (New Taiwan dollar)
18. PAY_AMT1 | Amount paid in September, 2005 | int (New Taiwan dollar)
19. PAY_AMT2 | Amount paid in August, 2005 | int (New Taiwan dollar)
20. PAY_AMT3 | Amount paid in July, 2005 | int (New Taiwan dollar)
21. PAY_AMT4 | Amount paid in June, 2005 | int (New Taiwan dollar)
22. PAY_AMT5 | Amount paid in May, 2005 | int (New Taiwan dollar)
23. PAY_AMT6 | Amount paid in April, 2005 | int (New Taiwan dollar)
24. Default payment next month | Target Variable | Yes: 1, No: 0 | binary

Le variabili da PAY0 a PAY6 sono categoriche (saranno trasformate in seguito). Sono codificate nel seguente modo: -1: pagamento effettuato in tempo; 2: pagamento ritardato di 2 mesi; … 8: pagamento ritardato di 8 mesi; 9: pagamento ritardato di 9 mesi o più.

Le features esaminate sono sia categoriche che continue. Quelle categoriche sono: `SEX`, `EDUCATION`, `MARRIAGE`, `PAY_1`, `PAY_2`, `PAY_3`, `PAY_4`, `PAY_5` and `PAY_6`. Quelle continue sono: `LIMIT_BAL`, `AGE`, `BILL_AMT1`, `BILL_AMT2`, `BILL_AMT3`, `BILL_AMT4`, `BILL_AMT5`, `BILL_AMT6`, `PAY_AMT1`, `PAY_AMT2`, `PAY_AMT3`, `PAY_AMT4`, `PAY_AMT5` e `PAY_AMT6`.

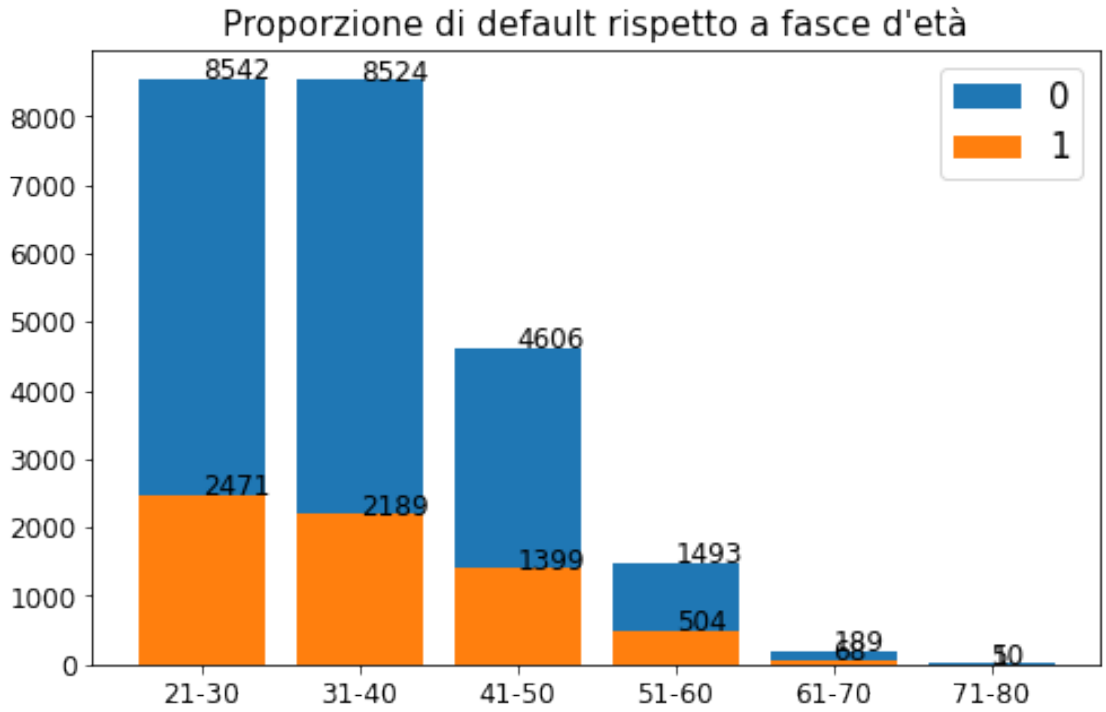
3.4 Distribuzioni

[10]:
```
plt.subplots(figsize=(20,5))
plt.subplot(121)
sns.distplot(dataset.LIMIT_BAL)

plt.subplot(122)
sns.distplot(dataset.AGE)

plt.show()
```



Dai grafici si evince che le variabili continue consistono nella limitazione del credito e l'età dei clienti. La maggior parte dei clienti ha credito nella fascia tra zero e 200k. Inoltre, per quanto concerne l'età, abbiamo la maggior parte dei clienti nella fascia 20-40. Di seguito osserveremo come le distribuzioni continue impattano sulla variabile target 'default pay.

[11]:
```
bins = [20,30,40,50,60,70,80]
names = ['21-30','31-40','41-50','51-60','61-70','71-80']
dataset['AGE_BIN'] = pd.cut(x=dataset.AGE, bins=bins, labels=names, right=True)

age_cnt = dataset.AGE_BIN.value_counts()
age_0 = (dataset.AGE_BIN[dataset['default pay'] == 0].value_counts())
age_1 = (dataset.AGE_BIN[dataset['default pay'] == 1].value_counts())

plt.subplots(figsize=(8,5))
plt.bar(age_0.index, age_0.values, label='0')
plt.bar(age_1.index, age_1.values, label='1')
for x,y in zip(names,age_0):
    plt.text(x,y,y,fontsize=12)
for x,y in zip(names,age_1):
    plt.text(x,y,y,fontsize=12)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.title("Proporzione di default rispetto a fasce d'età", fontsize=15)
plt.legend(loc='upper right', fontsize=15)
plt.show()
dataset.drop('AGE_BIN', axis=1, inplace=True)
```

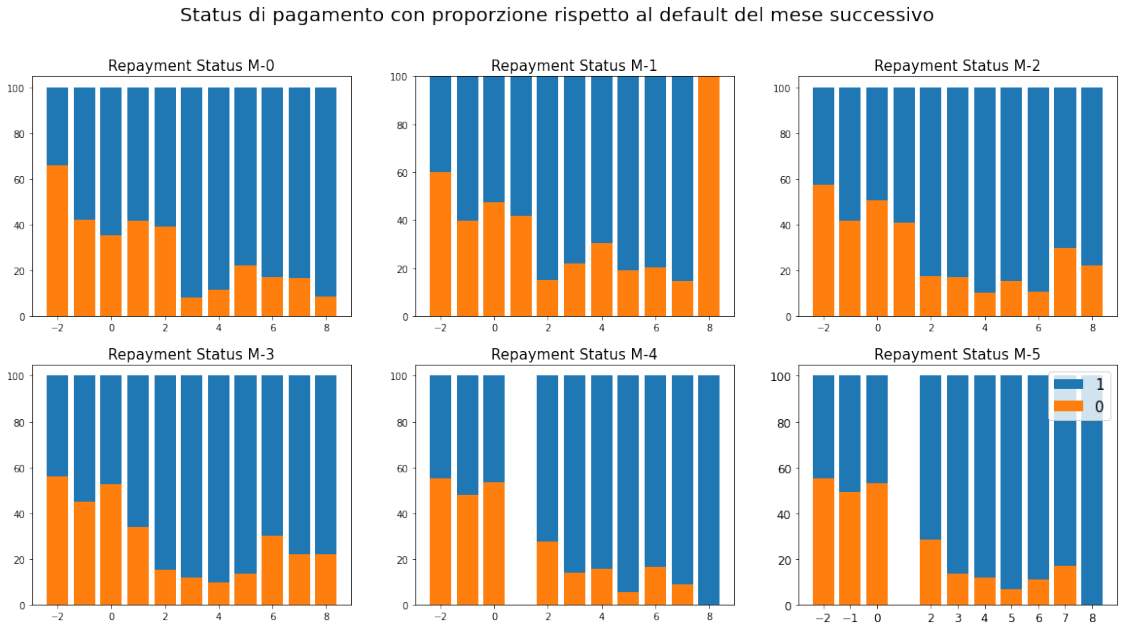


La maggior parte dei clienti appartiene alla fascia tra 21 e 30 anni. Si nota come con l'aumentare dell'età del cliente decresce la probabilità di default nel mese successivo. Possiamo concludere che si tratti di una feature importante per la predizione.

[12]:
```
## Inserire link
plt.subplots(figsize=(20,10))

ind = sorted(dataset.PAY_1.unique())
pay_0 = (dataset.PAY_1[dataset['default pay'] == 0].
 ↪value_counts(normalize=True))
pay_1 = (dataset.PAY_1[dataset['default pay'] == 1].
 ↪value_counts(normalize=True))
total = pay_0.values+pay_1.values
pay_0_prop = np.true_divide(pay_0, total)*100
pay_1_prop = np.true_divide(pay_1, total)*100
plt.subplot(231)
plt.bar(ind, pay_1_prop, bottom=pay_0_prop, label='1')
plt.bar(ind, pay_0_prop, label='0')
plt.title("Repayment Status M-0", fontsize=15)

ind = sorted(dataset.PAY_2.unique())
pay_0 = (dataset.PAY_2[dataset['default pay'] == 0].
 ↪value_counts(normalize=True))
```

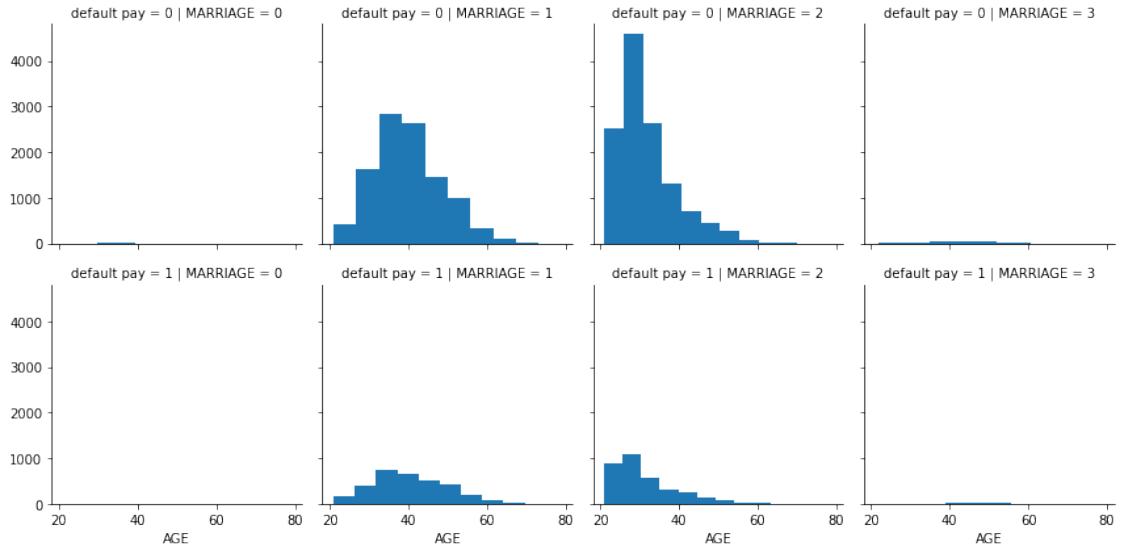
```
pay_1 = (dataset.PAY_2[dataset['default pay'] == 1].
 ↪value_counts(normalize=True))
for i in pay_0.index:
    if i not in pay_1.index:
        pay_1[i]=0
total = pay_0.values+pay_1.values
pay_0_prop = np.true_divide(pay_0, total)*100
pay_1_prop = np.true_divide(pay_1, total)*100
plt.subplot(232)
plt.bar(ind, pay_1_prop, bottom=pay_0_prop, label='1')
plt.bar(ind, pay_0_prop, label='0')
plt.title("Repayment Status M-1", fontsize=15)

ind = sorted(dataset.PAY_3.unique())
pay_0 = (dataset.PAY_3[dataset['default pay'] == 0].
 ↪value_counts(normalize=True))
pay_1 = (dataset.PAY_3[dataset['default pay'] == 1].
 ↪value_counts(normalize=True))
for i in pay_0.index:
    if i not in pay_1.index:
        pay_1[i]=0
total = pay_0.values+pay_1.values
pay_0_prop = np.true_divide(pay_0, total)*100
pay_1_prop = np.true_divide(pay_1, total)*100
plt.subplot(233)
plt.bar(ind, pay_1_prop, bottom=pay_0_prop, label='1')
plt.bar(ind, pay_0_prop, label='0')
plt.title("Repayment Status M-2", fontsize=15)

ind = sorted(dataset.PAY_4.unique())
pay_0 = (dataset.PAY_4[dataset['default pay'] == 0].
 ↪value_counts(normalize=True))
pay_1 = (dataset.PAY_4[dataset['default pay'] == 1].
 ↪value_counts(normalize=True))
for i in pay_0.index:
    if i not in pay_1.index:
        pay_1[i]=0
total = pay_0.values+pay_1.values
pay_0_prop = np.true_divide(pay_0, total)*100
pay_1_prop = np.true_divide(pay_1, total)*100
plt.subplot(234)
plt.bar(ind, pay_1_prop, bottom=pay_0_prop, label='1')
plt.bar(ind, pay_0_prop, label='0')
plt.title("Repayment Status M-3", fontsize=15)

ind = sorted(dataset.PAY_5.unique())
```

```
pay_0 = (dataset.PAY_5[dataset['default pay'] == 0].
 ↪value_counts(normalize=True))
pay_1 = (dataset.PAY_5[dataset['default pay'] == 1].
 ↪value_counts(normalize=True))
for i in pay_0.index:
    if i not in pay_1.index:
        pay_1[i]=0
for i in pay_1.index:
    if i not in pay_0.index:
        pay_0[i]=0
total = pay_0.values+pay_1.values
pay_0_prop = np.true_divide(pay_0, total)*100
pay_1_prop = np.true_divide(pay_1, total)*100
plt.subplot(235)
plt.bar(ind, pay_1_prop, bottom=pay_0_prop, label='1')
plt.bar(ind, pay_0_prop, label='0')
plt.title("Repayment Status M-4", fontsize=15)

ind = sorted(dataset.PAY_6.unique())
pay_0 = (dataset.PAY_6[dataset['default pay'] == 0].
 ↪value_counts(normalize=True))
pay_1 = (dataset.PAY_6[dataset['default pay'] == 1].
 ↪value_counts(normalize=True))
for i in pay_0.index:
    if i not in pay_1.index:
        pay_1[i]=0
for i in pay_1.index:
    if i not in pay_0.index:
        pay_0[i]=0
total = pay_0.values+pay_1.values
pay_0_prop = np.true_divide(pay_0, total)*100
pay_1_prop = np.true_divide(pay_1, total)*100
plt.subplot(236)
plt.bar(ind, pay_1_prop, bottom=pay_0_prop, label='1')
plt.bar(ind, pay_0_prop, label='0')
plt.title("Repayment Status M-5", fontsize=15)

plt.xticks(ind, fontsize=12)
plt.yticks(fontsize=12)
plt.legend(loc="upper right", fontsize=15)
plt.suptitle("Status di pagamento con proporzione rispetto al default del mese␣
 ↪successivo", fontsize=20)

plt.show()
```



I grafici sopra mostrano la proporzione dei clienti che andranno in default il mese successivo rispetto allo status di un pagamento. Dato un generico mese e il suo status, si evidenzia che più tardi un pagamento è completato più è probabile il deault.

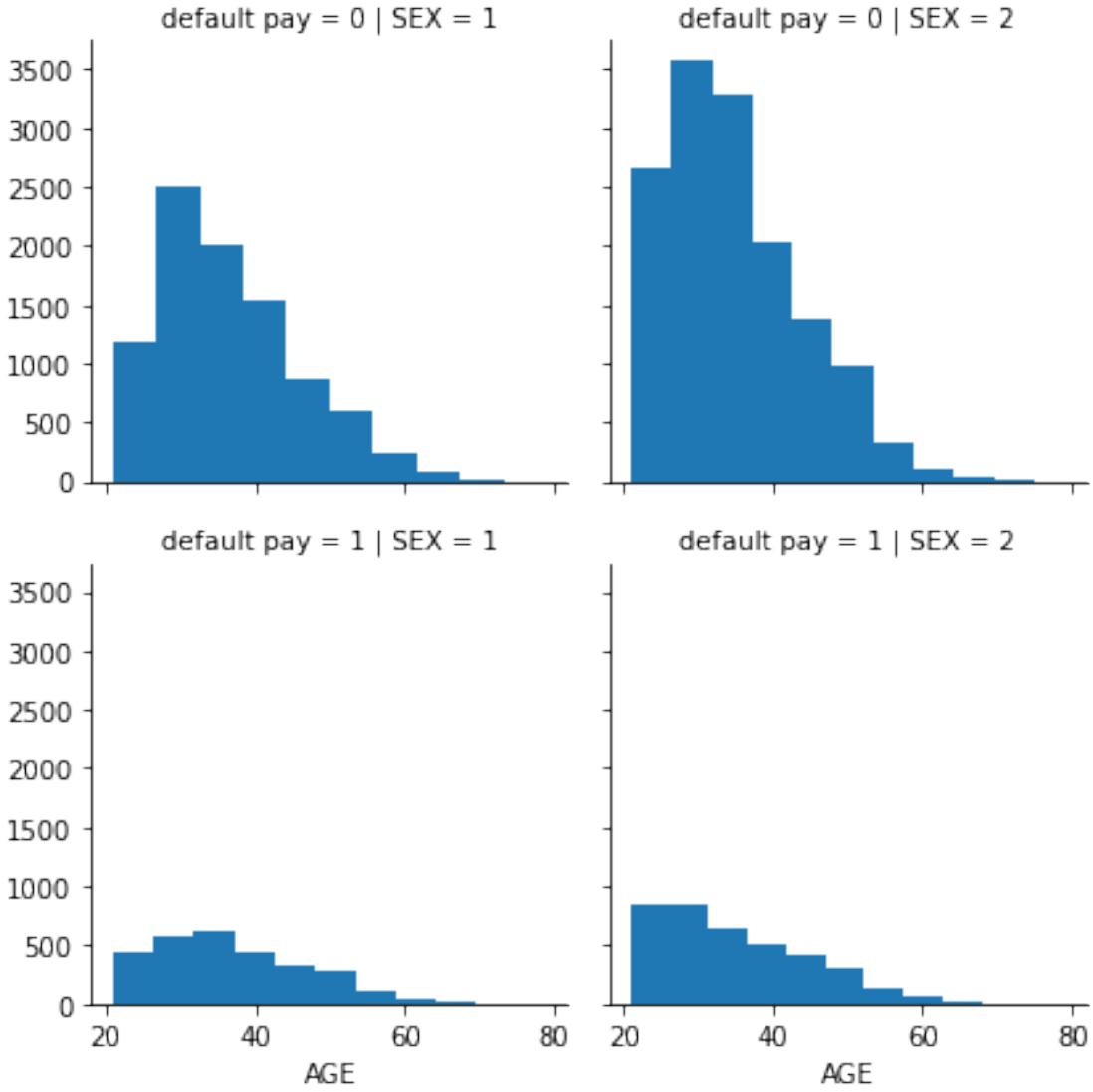
[13]:
```
g = sns.FacetGrid(dataset, row='default pay', col='MARRIAGE')
g = g.map(plt.hist, 'AGE')
plt.show()
```



Dal grafico soprastante si può apprendere che le persone che sono andate in default con molta prob-

abilità sono sposate nella fascia di età compresa tra 30 e 50 anni `[2, 2]` e non sposate nella fascia di età tra 20 e 30 `[2, 3]`.Possiamo concludere che la feature MARRIAGE può essere importante nel predirre il default.

[14]:
```
g = sns.FacetGrid(dataset, row='default pay', col='SEX')
g = g.map(plt.hist, 'AGE')
```



al grafico soprastante si può apprendere che le persone che sono andate in default con molta probabilità, paragonandole con persone di sesso maschile in tutte le fasce d'età, sono di sesso femminile nella fascia di età compresa tra 20 e 30 anni `[2, 2]`. Possiamo concludere che la feature SEX può essere importante nel predirre il default.

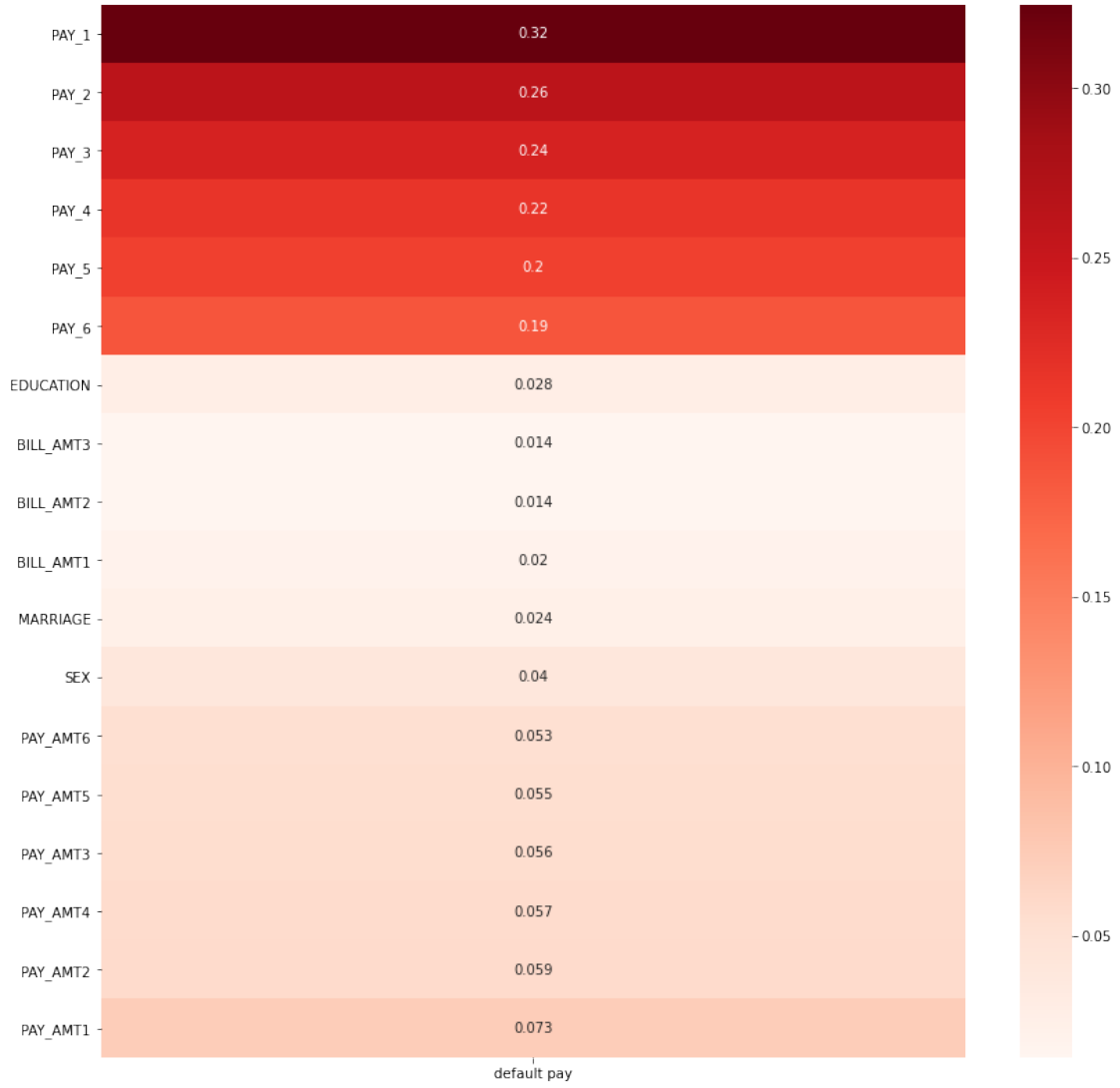
3.5 Correlazione tra features

Uso la correlazione di Pearson

[15]:
```
cor = (dataset.iloc[:,1:].corr()).drop('default pay')
```

[16]:
```
minCorrelation = 0.014
highCorrelation = [i
                   for i, correlation in enumerate(abs(cor['default pay']))
                   if correlation>minCorrelation]
```

[17]:
```
plt.figure(figsize=(14,14))
sns.heatmap(abs(cor [['default pay']].iloc[highCorrelation].
 ↪sort_values(by='default pay', ascending=False)), annot=True, cmap=plt.cm.
 ↪Reds)
plt.show()
```



3.6 Data preprocessing

3.6.1 One-Hot encoding

Eseguo pre-processing sulle variabili categoriche attraverso One-Hot-Encoding. Questo perchè non si tratta di valori ordinali, ma di semplici codifiche per rappresentare classi diverse.

[18]:
```
categorical_cols = ['SEX', 'EDUCATION', 'MARRIAGE', 'PAY_1', 'PAY_2', 'PAY_3',␣
↪'PAY_4', 'PAY_5', 'PAY_6']
```

[19]:
```
dataset = pd.get_dummies(dataset, columns = categorical_cols)
```

Ottengo i dataset

[20]:
```
y = dataset['default pay']
x = dataset.drop(['default pay'], axis=1)
all_features = [str(index) for index in x.columns]
```

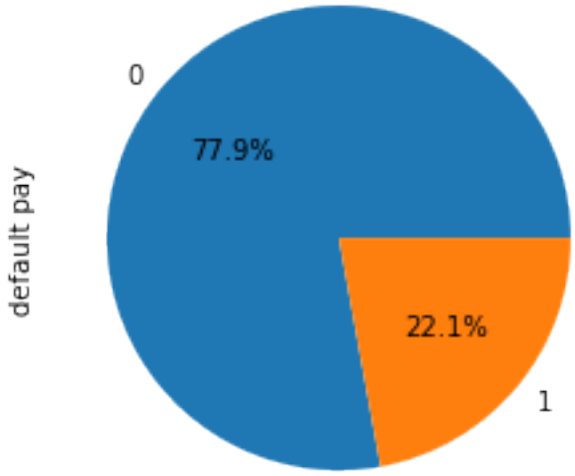
3.6.2 Sbilanciamento classi

Dalle seguente cella si può verificare lo sbilanciamento delle classi. Si può risolvere la questione attraverso l'oversampling, l'undersampling oppure dando più peso agli errori sulla classe meno rappresentativa.

[21]:
```
print(f'Percentuale della classe meno rappresentativa: {y[y == 1].shape[0] / y.
↪shape[0]:.1%}')
dataset['default pay'].value_counts().plot.pie(autopct='%1.1f%%')
```

```
Percentuale della classe meno rappresentativa: 22.1%
```

[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00c1439d50>
```



Per risolvere il problema ho deciso di adottare una tecnica di over sampling usando la libreria SMOTE.

[22]:
```
#conda install -c conda-forge imbalanced-learn
from imblearn.over_sampling import SMOTE
```

[23]:
```
sm = SMOTE(random_state=42)
x, y = sm.fit_resample(x, y)
print(f'Verifico che le classi siano bilanciate: {y[y == 0].shape == y[y == 1].
↪shape}')
```

```
Verifico che le classi siano bilanciate: True
```

3.6.3 Data splitting and scaling

Divido il dataset nello sottoparti di training e validation set (30%).

[24]:
```
X_train, X_val, y_train, y_val = train_test_split(
    x, y, test_size=0.3, random_state=randSeed)
```

[25]:
```
X_train_relevant, X_val_relevant = X_train[:, highCorrelation], X_val[:,␣
↪highCorrelation]
```

Applico la standardizzazione delle features.

[26]:
```
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_val = sc.transform(X_val)
```

```
X_train_relevant = sc.fit_transform(X_train_relevant)
X_val_relevant = sc.transform(X_val_relevant)
```

4 Modelli

Definisco una funzione per il calcolo dell'intervallo di confidenza.

[27]:
```
def diff_interval(a1, a2, N1, N2, Z):
    d = abs(a1 - a2)
    sd = np.sqrt(a1 * (1-a1) / N1 + a2 * (1-a2) / N2)
    return d - Z * sd, d + Z * sd

def different_model(score1, score2, X, level=0.95):
    N = len(X)
    Z = norm.ppf((1 + level) / 2)
    return diff_interval(score1, score2, N, N, Z)
```

[28]:
```
from scipy.stats import norm

def conf_interval(model, X, y, level=0.95):
    score = f1_score(y, model.predict(X))
    N = len(X)
    Z = norm.ppf((1 + level) / 2)
    c = (2 * N * score + Z**2) / (2 * (N + Z**2))
    d = Z * np.sqrt(Z**2 + 4*N*score - 4*N*score**2) / (2 * (N + Z**2))
    return c - d, c + d
```

4.1 Perceptron

[29]:
```
from sklearn.linear_model import Perceptron
perceptron = Perceptron(random_state=randSeed)
```

[30]:
```
perceptron.fit(X_train, y_train)
```

[30]:
```
Perceptron(alpha=0.0001, class_weight=None, early_stopping=False, eta0=1.0,
           fit_intercept=True, max_iter=1000, n_iter_no_change=5, n_jobs=None,
           penalty=None, random_state=42, shuffle=True, tol=0.001,
           validation_fraction=0.1, verbose=0, warm_start=False)
```

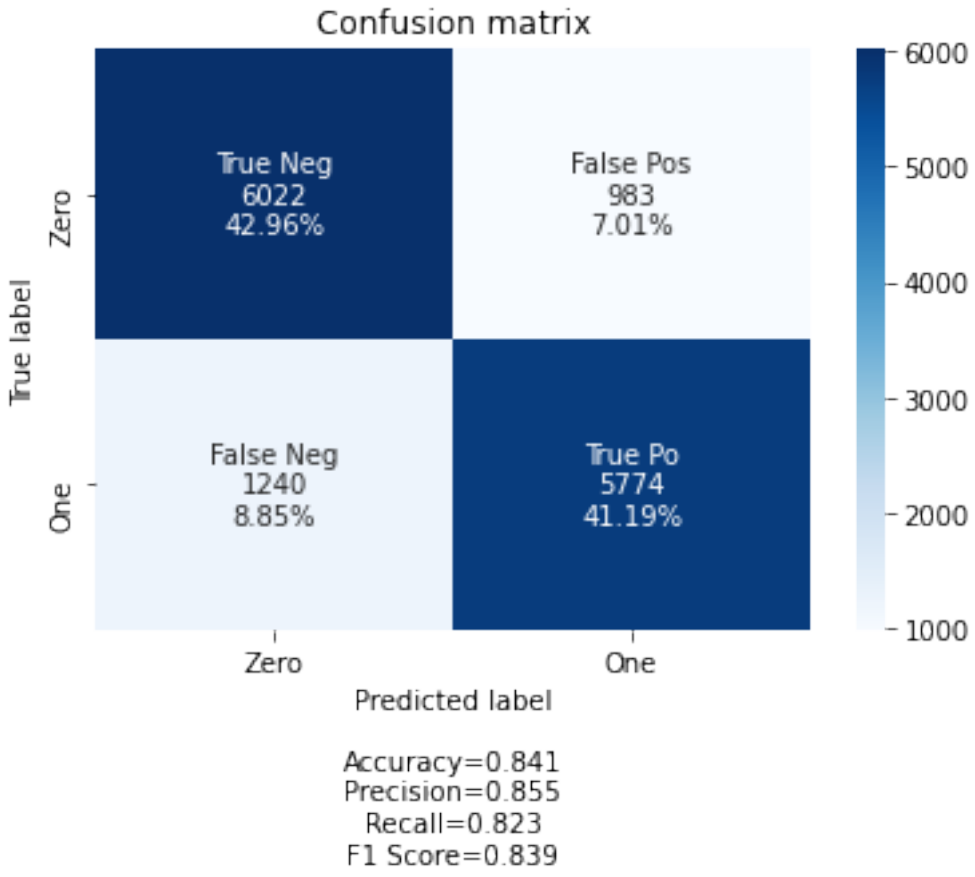
4.1.1 Valutazione modello

[31]:
```
models["Perceptron"] = {
    'conf_interval': conf_interval(perceptron, X_val, y_val,␣
 ↪level=confidenceLevel),
    'f1-score': f1_score(y_val, perceptron.predict(X_val))
}
print(f'F1-score del modello: {models["Perceptron"]["f1-score"]:.2%}')
```

```
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["Perceptron"]["conf_interval"]}')
```

```
F1-score del modello: 83.86%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.8323907730095819,
0.8445713562341648)
```

[32]:
```
y_pred = perceptron.predict(X_val)
y_pred = (y_pred > 0.5)
conf_matr = confusion_matrix(y_val, y_pred)
labels = ['True Neg','False Pos','False Neg','True Po']
categories = ['Zero', 'One']
make_confusion_matrix(conf_matr,
                      group_names=labels,
                      categories=categories,
                      cmap='Blues',
                      title='Confusion matrix')
```



4.2 Logistic regression

4.2.1 Definizione modello

[33]:
```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import f1_score, confusion_matrix
```

[34]:
```
model = Pipeline([
    ('logRegr', LogisticRegression(solver='saga'))
])
grid = [{
    'logRegr__penalty': ['l2', 'l1'],
    'logRegr__C': np.logspace(-2, 2, 5),
},
{
    'logRegr__penalty': ['elasticnet'],
    'logRegr__C': np.logspace(-2, 2, 5),
    'logRegr__l1_ratio': [0.2, 0.5]
}]
gs_all = GridSearchCV(model, grid, cv=skf, n_jobs=-1)
gs_important_features = GridSearchCV(model, grid, cv=skf, n_jobs=-1)
```

4.2.2 Solo features importanti

[35]:
```
gs_important_features.fit(X_train_relevant, y_train);
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_sag.py:330:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  "the coef_ did not converge", ConvergenceWarning)
```

[36]:
```
pd.DataFrame(gs_important_features.cv_results_).sort_values("rank_test_score").
↪head(5)
```

[36]:

	mean_fit_time	std_fit_time	…	std_test_score	rank_test_score
14	1.484388	0.008879	…	0.002248	1
17	1.492534	0.001864	…	0.002209	2
2	1.145743	0.005577	…	0.002097	2
4	1.149328	0.003335	…	0.002172	2
19	1.385795	0.140325	…	0.002210	5

[5 rows x 14 columns]

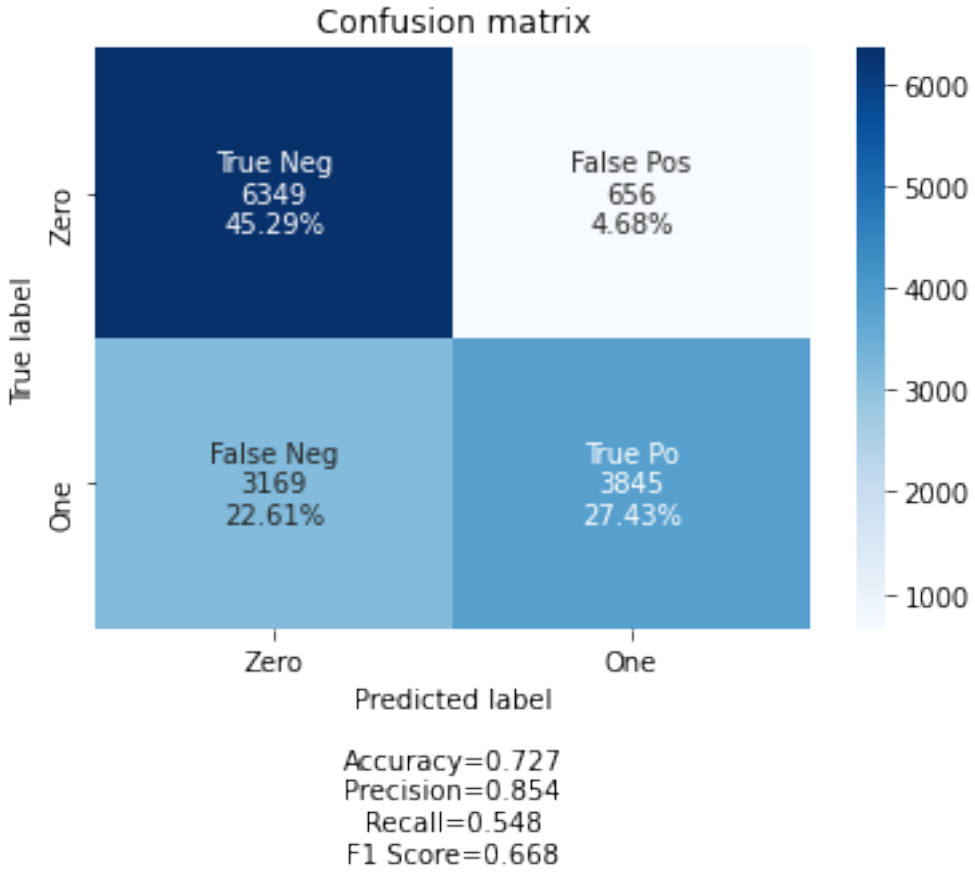
Valutazione modello

[37]:
```
models["Logistic regression important features"] = {
    'conf_interval': conf_interval(gs_important_features, X_val_relevant,␣
↪y_val, level=confidenceLevel),
    'f1-score': f1_score(y_val, gs_important_features.predict(X_val_relevant))
```

```
}
print(f'F1-score del modello: {models["Logistic regression important␣
 ↪features"]["f1-score"]:.2%}')
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["Logistic regression important features"]["conf_interval"]}')
```

```
F1-score del modello: 66.78%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.6599829478534277,
0.6755742566993698)
```

[38]:
```
y_pred = gs_important_features.predict(X_val_relevant)
y_pred = (y_pred > 0.5)
conf_matr = confusion_matrix(y_val, y_pred)
labels = ['True Neg','False Pos','False Neg','True Po']
categories = ['Zero', 'One']
make_confusion_matrix(conf_matr,
                      group_names=labels,
                      categories=categories,
                      cmap='Blues',
                      title='Confusion matrix')
```



4.2.3 Tutte le features

[39]:
```
gs_all.fit(X_train, y_train);
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_sag.py:330:
ConvergenceWarning: The max_iter was reached which means the coef_ did not
converge
  "the coef_ did not converge", ConvergenceWarning)
```

Applico una tecnica di Grid Search per ottimizare gli iperparametri. Poi alleno il miglior modello ottenuto

[40]:
```
pd.DataFrame(gs_all.cv_results_).sort_values("rank_test_score").head(5)
```

[40]:

	mean_fit_time	std_fit_time	…	std_test_score	rank_test_score
6	4.266871	0.030248	…	0.000903	1
18	5.881339	0.029593	…	0.000938	2
19	5.190942	1.018296	…	0.000977	3
16	5.883450	0.038273	…	0.000977	3
7	5.710814	0.022872	…	0.000977	3

[5 rows x 14 columns]

Valutazione modello

[41]:
```
pd.Series(gs_all.best_estimator_['logRegr'].coef_[0], index=all_features).
↪sort_values(ascending=False)
```

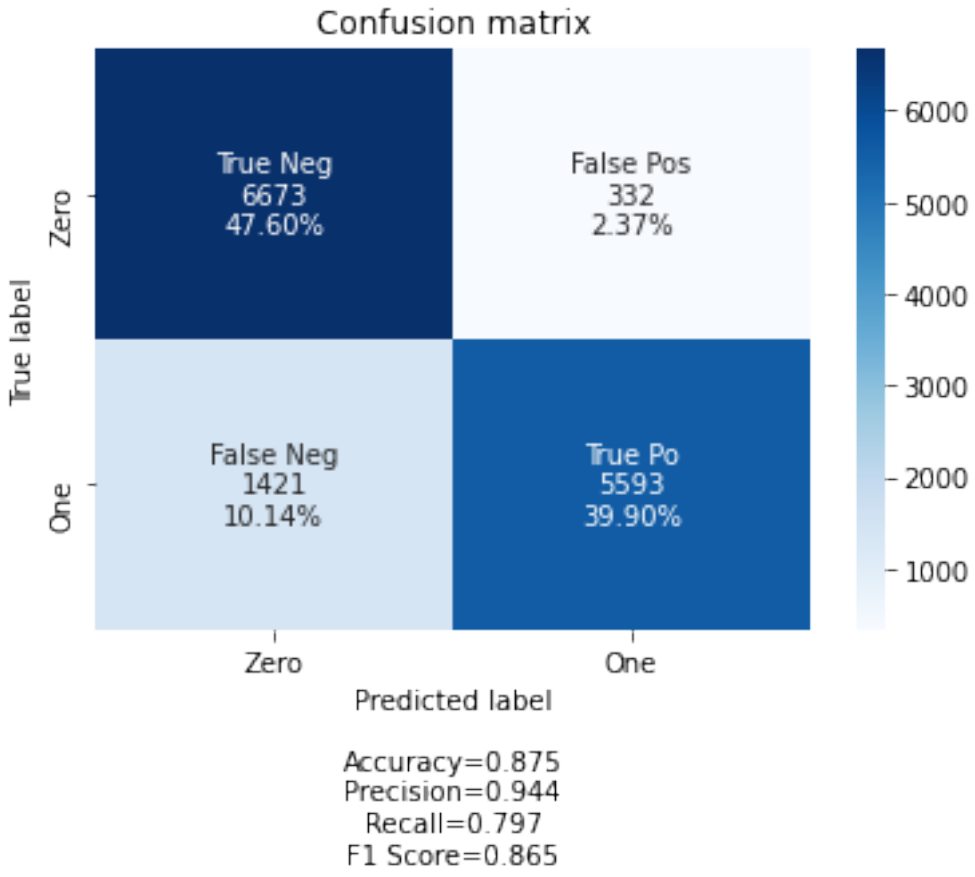
[41]:
```
BILL_AMT3      0.105174
BILL_AMT2      0.100193
PAY_6_8        0.055801
BILL_AMT4      0.032761
PAY_5_6        0.028050
                  …
EDUCATION_1   -0.861241
EDUCATION_2   -0.906989
SEX_2         -0.912564
MARRIAGE_2    -0.931572
PAY_1_0       -1.062516
Length: 91, dtype: float64
```

[42]:
```
models["Logistic regression all features"] = {
    'conf_interval': conf_interval(gs_all, X_val, y_val, level=confidenceLevel),
    'f1-score': f1_score(y_val, gs_all.predict(X_val))
}
print(f'F1-score del modello: {models["Logistic regression all␣
↪features"]["f1-score"]:.2%}')
```

```
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["Logistic regression all features"]["conf_interval"]}')
```

```
F1-score del modello: 86.45%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.8587529364230313,
0.8700835962282556)
```

[43]:
```
y_pred = gs_all.predict(X_val)
y_pred = (y_pred > 0.5)
conf_matr = confusion_matrix(y_val, y_pred)
labels = ['True Neg','False Pos','False Neg','True Po']
categories = ['Zero', 'One']
make_confusion_matrix(conf_matr,
                      group_names=labels,
                      categories=categories,
                      cmap='Blues',
                      title='Confusion matrix')
```



Come si evince, il modello di regressione logistica associa più importanza allo stato di pagamento

dell'utimo mese.

4.2.4 Valutazioni finali logistic regression

Il modello di regressione logistica supera il modello perceptron. In particolare, si nota che si raggiunge un'accuratezza maggiore se si usano tutte le features nel training del modello. Sono sicuro di questa accuratezza più elevata con una confidenza del 95%.

[44]:
```
score_important_features = f1_score(y_val, gs_important_features.
 ↪predict(X_val_relevant))
score_all = f1_score(y_val, gs_all.predict(X_val))
print(f"Secondo modello migliore del primo modello?: Si␣
 ↪{different_model(score_important_features, score_all, X_val, level=0.95)}")
```

```
Secondo modello migliore del primo modello?: Si (0.18705603940881904,
0.20633105431802629)
```

4.3 SVM

[45]:
```
from thundersvm import SVC
svm = SVC()
grid = {
    'kernel': ['rbf'],
    'C': [0.01, 0.1, 1]
}
gs = GridSearchCV(svm, grid, cv=skf)
gs.fit(X_train, y_train)
```

[45]:
```
GridSearchCV(cv=StratifiedKFold(n_splits=3, random_state=42, shuffle=True),
             error_score=nan,
             estimator=SVC(C=1.0, cache_size=None, class_weight=None, coef0=0.0,
                           decision_function_shape='ovo', degree=3,
                           gamma='auto', gpu_id=0, kernel='rbf', max_iter=-1,
                           max_mem_size=-1, n_jobs=-1, probability=False,
                           random_state=None, shrinking=False, tol=0.001,
                           verbose=False),
             iid='deprecated', n_jobs=None,
             param_grid={'C': [0.01, 0.1, 1], 'kernel': ['rbf']},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

4.3.1 Valutazione modello

[46]:
```
svm = gs.best_estimator_
models["SVM"] = {
    'conf_interval': conf_interval(svm, X_val, y_val, level=confidenceLevel),
    'f1-score': f1_score(y_val, svm.predict(X_val))
}
print(f'F1-score del modello: {models["SVM"]["f1-score"]:.2%}')
```

```
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["SVM"]["conf_interval"]}')
```

```
F1-score del modello: 86.24%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.8566398399526858,
0.868043265861745)
```

4.4 Decision tree

[48]:
```
treeClassifier = tree.DecisionTreeClassifier(random_state=randSeed)
grid = {
    "max_depth": [5, 10, 15, 20, 25, 30, None],
    "min_samples_split": [2, 50, 100]
}
gs = GridSearchCV(treeClassifier, grid, cv=skf)
gs.fit(X_train, y_train)
```

[48]:
```
GridSearchCV(cv=StratifiedKFold(n_splits=3, random_state=42, shuffle=True),
             error_score=nan,
             estimator=DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
                                              criterion='gini', max_depth=None,
                                              max_features=None,
                                              max_leaf_nodes=None,
                                              min_impurity_decrease=0.0,
                                              min_impurity_split=None,
                                              min_samples_leaf=1,
                                              min_samples_split=2,
                                              min_weight_fraction_leaf=0.0,
                                              presort='deprecated',
                                              random_state=42,
                                              splitter='best'),
             iid='deprecated', n_jobs=None,
             param_grid={'max_depth': [5, 10, 15, 20, 25, 30, None],
                         'min_samples_split': [2, 50, 100]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

[49]:
```
treeClassifier = gs.best_estimator_
print(f"Numero di foglie dell'albero: {treeClassifier.get_n_leaves()}")
```

```
Numero di foglie dell'albero: 257
```

4.4.1 Visualizzazione modello

[50]:
```
dot_data = tree.export_graphviz(treeClassifier, out_file=None,
                                feature_names=all_features,
                                class_names="Default",
```

```
                                    filled=True)
graph = graphviz.Source(dot_data, format="png")
```

4.4.2 Valutazione modello

[51]:
```
models["Decision tree"] = {
    'conf_interval': conf_interval(treeClassifier, X_val, y_val,␣
 ↪level=confidenceLevel),
    'f1-score': f1_score(y_val, treeClassifier.predict(X_val))
}
print(f'F1-score del modello: {models["Decision tree"]["f1-score"]:.2%}')
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["Decision tree"]["conf_interval"]}')
```

```
F1-score del modello: 83.62%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.829962923808452,
0.8422158805949433)
```

4.5 XGBoost

[52]:
```
xgboost = XGBClassifier()
xgboost.fit(X_train, y_train)
```

[52]:
```
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0,
              learning_rate=0.1, max_delta_step=0, max_depth=3,
              min_child_weight=1, missing=None, n_estimators=100, n_jobs=1,
              nthread=None, objective='binary:logistic', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, seed=None,
              silent=None, subsample=1, verbosity=1)
```

Valutazione modello

[53]:
```
models["XGBoost"] = {
    'conf_interval': conf_interval(xgboost, X_val, y_val,␣
 ↪level=confidenceLevel),
    'f1-score': f1_score(y_val, xgboost.predict(X_val))
}
print(f'F1-score del modello: {models["XGBoost"]["f1-score"]:.2%}')
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["XGBoost"]["conf_interval"]}')
```

```
F1-score del modello: 86.17%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.8559308097784665,
0.8673584585360085)
```

4.6 Artificial Neural Network

Per prima cosa verifichiamo che si dispone di una GPU per il training.

[54]:
```
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
```

```
Num GPUs Available:  1
```

4.6.1 Definizione modello

[57]:
```
def build_nn():
  numberOfHiddenLayers = 5
  numberOfNeuronsForEachHL = [50, 40, 30, 20, 10]
  nn = Sequential()
  for i in range(numberOfHiddenLayers):
    if i == 0:
        nn.add(Dense(numberOfNeuronsForEachHL[i], input_dim=X_train.shape[1],␣
 ↪kernel_initializer='uniform', activation='relu'))
    else:
        nn.add(Dense(numberOfNeuronsForEachHL[i],␣
 ↪kernel_initializer=glorot_uniform(seed=0), activation='relu'))
  nn.add(Dense(1, kernel_initializer=glorot_uniform(seed=0),␣
 ↪activation='sigmoid'))
  nn.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
  nn.summary()
  return nn


mode_path = '../models/ann.h5'
callbacks = [ModelCheckpoint(filepath=mode_path, save_best_only=True)]

model = KerasClassifier(build_fn=build_nn, verbose=1, epochs=100,␣
 ↪batch_size=256,
                        validation_data=(X_val, y_val), callbacks=callbacks)
```

[58]:
```
history = model.fit(X_train, y_train)
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_6 (Dense)              (None, 50)                4600
_________________________________________________________________
dense_7 (Dense)              (None, 40)                2040
_________________________________________________________________
dense_8 (Dense)              (None, 30)                1230
_________________________________________________________________
dense_9 (Dense)              (None, 20)                620
_________________________________________________________________
```

```
dense_10 (Dense)             (None, 10)                210
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 11
=================================================================
Total params: 8,711
Trainable params: 8,711
Non-trainable params: 0
_________________________________________________________________
Epoch 1/100
128/128 [==============================] - 2s 8ms/step - loss: 0.5259 -
accuracy: 0.7430 - val_loss: 0.3073 - val_accuracy: 0.8750
Epoch 2/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2931 -
accuracy: 0.8786 - val_loss: 0.3012 - val_accuracy: 0.8722
Epoch 3/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2821 -
accuracy: 0.8835 - val_loss: 0.3065 - val_accuracy: 0.8719
Epoch 4/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2823 -
accuracy: 0.8816 - val_loss: 0.2954 - val_accuracy: 0.8740
Epoch 5/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2788 -
accuracy: 0.8850 - val_loss: 0.2978 - val_accuracy: 0.8757
Epoch 6/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2772 -
accuracy: 0.8848 - val_loss: 0.2960 - val_accuracy: 0.8743
Epoch 7/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2747 -
accuracy: 0.8859 - val_loss: 0.2942 - val_accuracy: 0.8745
Epoch 8/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2751 -
accuracy: 0.8862 - val_loss: 0.2976 - val_accuracy: 0.8743
Epoch 9/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2715 -
accuracy: 0.8884 - val_loss: 0.2984 - val_accuracy: 0.8723
Epoch 10/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2676 -
accuracy: 0.8900 - val_loss: 0.3014 - val_accuracy: 0.8735
Epoch 11/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2685 -
accuracy: 0.8867 - val_loss: 0.3011 - val_accuracy: 0.8731
Epoch 12/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2659 -
accuracy: 0.8910 - val_loss: 0.3083 - val_accuracy: 0.8648
Epoch 13/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2704 -
accuracy: 0.8892 - val_loss: 0.3009 - val_accuracy: 0.8736
Epoch 14/100
```

```
128/128 [==============================] - 1s 5ms/step - loss: 0.2698 -
accuracy: 0.8861 - val_loss: 0.2991 - val_accuracy: 0.8735
Epoch 15/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2671 -
accuracy: 0.8871 - val_loss: 0.2975 - val_accuracy: 0.8740
Epoch 16/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2632 -
accuracy: 0.8911 - val_loss: 0.3011 - val_accuracy: 0.8726
Epoch 17/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2683 -
accuracy: 0.8874 - val_loss: 0.3034 - val_accuracy: 0.8734
Epoch 18/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2626 -
accuracy: 0.8900 - val_loss: 0.3012 - val_accuracy: 0.8730
Epoch 19/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2625 -
accuracy: 0.8891 - val_loss: 0.3017 - val_accuracy: 0.8747
Epoch 20/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2693 -
accuracy: 0.8883 - val_loss: 0.3047 - val_accuracy: 0.8725
Epoch 21/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2571 -
accuracy: 0.8937 - val_loss: 0.3073 - val_accuracy: 0.8729
Epoch 22/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2599 -
accuracy: 0.8910 - val_loss: 0.3030 - val_accuracy: 0.8723
Epoch 23/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2609 -
accuracy: 0.8918 - val_loss: 0.3097 - val_accuracy: 0.8720
Epoch 24/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2645 -
accuracy: 0.8902 - val_loss: 0.3103 - val_accuracy: 0.8724
Epoch 25/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2595 -
accuracy: 0.8939 - val_loss: 0.3128 - val_accuracy: 0.8712
Epoch 26/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2592 -
accuracy: 0.8922 - val_loss: 0.3084 - val_accuracy: 0.8726
Epoch 27/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2559 -
accuracy: 0.8928 - val_loss: 0.3104 - val_accuracy: 0.8722
Epoch 28/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2596 -
accuracy: 0.8925 - val_loss: 0.3172 - val_accuracy: 0.8693
Epoch 29/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2539 -
accuracy: 0.8954 - val_loss: 0.3139 - val_accuracy: 0.8689
Epoch 30/100
```

```
128/128 [==============================] - 1s 5ms/step - loss: 0.2536 -
accuracy: 0.8948 - val_loss: 0.3217 - val_accuracy: 0.8705
Epoch 31/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2485 -
accuracy: 0.8975 - val_loss: 0.3166 - val_accuracy: 0.8715
Epoch 32/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2504 -
accuracy: 0.8952 - val_loss: 0.3213 - val_accuracy: 0.8704
Epoch 33/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2491 -
accuracy: 0.8968 - val_loss: 0.3243 - val_accuracy: 0.8673
Epoch 34/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2487 -
accuracy: 0.8976 - val_loss: 0.3309 - val_accuracy: 0.8701
Epoch 35/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2480 -
accuracy: 0.8953 - val_loss: 0.3250 - val_accuracy: 0.8690
Epoch 36/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2523 -
accuracy: 0.8941 - val_loss: 0.3289 - val_accuracy: 0.8684
Epoch 37/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2485 -
accuracy: 0.8967 - val_loss: 0.3330 - val_accuracy: 0.8694
Epoch 38/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2471 -
accuracy: 0.8995 - val_loss: 0.3289 - val_accuracy: 0.8658
Epoch 39/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2522 -
accuracy: 0.8948 - val_loss: 0.3305 - val_accuracy: 0.8694
Epoch 40/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2384 -
accuracy: 0.9022 - val_loss: 0.3289 - val_accuracy: 0.8687
Epoch 41/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2422 -
accuracy: 0.8999 - val_loss: 0.3467 - val_accuracy: 0.8667
Epoch 42/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2469 -
accuracy: 0.8986 - val_loss: 0.3382 - val_accuracy: 0.8683
Epoch 43/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2413 -
accuracy: 0.8997 - val_loss: 0.3417 - val_accuracy: 0.8663
Epoch 44/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2398 -
accuracy: 0.8993 - val_loss: 0.3389 - val_accuracy: 0.8678
Epoch 45/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2415 -
accuracy: 0.8999 - val_loss: 0.3469 - val_accuracy: 0.8687
Epoch 46/100
```

```
128/128 [==============================] - 1s 5ms/step - loss: 0.2341 -
accuracy: 0.9038 - val_loss: 0.3479 - val_accuracy: 0.8680
Epoch 47/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2375 -
accuracy: 0.9007 - val_loss: 0.3528 - val_accuracy: 0.8672
Epoch 48/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2313 -
accuracy: 0.9054 - val_loss: 0.3525 - val_accuracy: 0.8653
Epoch 49/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2375 -
accuracy: 0.9018 - val_loss: 0.3510 - val_accuracy: 0.8654
Epoch 50/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2352 -
accuracy: 0.9013 - val_loss: 0.3606 - val_accuracy: 0.8658
Epoch 51/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2339 -
accuracy: 0.9032 - val_loss: 0.3596 - val_accuracy: 0.8671
Epoch 52/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2328 -
accuracy: 0.9020 - val_loss: 0.3610 - val_accuracy: 0.8673
Epoch 53/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2319 -
accuracy: 0.9048 - val_loss: 0.3673 - val_accuracy: 0.8649
Epoch 54/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2276 -
accuracy: 0.9046 - val_loss: 0.3642 - val_accuracy: 0.8663
Epoch 55/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2328 -
accuracy: 0.9027 - val_loss: 0.3670 - val_accuracy: 0.8647
Epoch 56/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2306 -
accuracy: 0.9040 - val_loss: 0.3649 - val_accuracy: 0.8670
Epoch 57/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2279 -
accuracy: 0.9036 - val_loss: 0.3904 - val_accuracy: 0.8600
Epoch 58/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2284 -
accuracy: 0.9032 - val_loss: 0.3814 - val_accuracy: 0.8683
Epoch 59/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2316 -
accuracy: 0.9007 - val_loss: 0.3922 - val_accuracy: 0.8638
Epoch 60/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2256 -
accuracy: 0.9067 - val_loss: 0.3864 - val_accuracy: 0.8640
Epoch 61/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2250 -
accuracy: 0.9055 - val_loss: 0.3948 - val_accuracy: 0.8659
Epoch 62/100
```

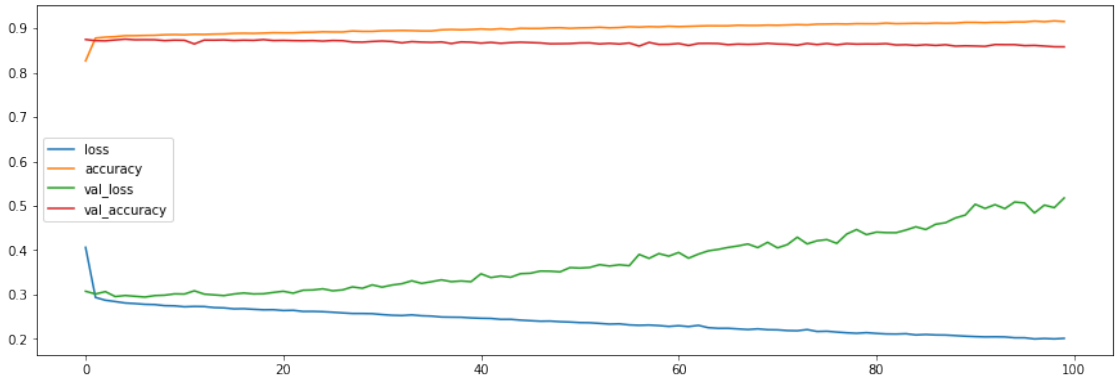
```
128/128 [==============================] - 1s 5ms/step - loss: 0.2252 -
accuracy: 0.9047 - val_loss: 0.3818 - val_accuracy: 0.8614
Epoch 63/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2253 -
accuracy: 0.9074 - val_loss: 0.3911 - val_accuracy: 0.8660
Epoch 64/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2223 -
accuracy: 0.9063 - val_loss: 0.3984 - val_accuracy: 0.8662
Epoch 65/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2210 -
accuracy: 0.9073 - val_loss: 0.4017 - val_accuracy: 0.8658
Epoch 66/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2211 -
accuracy: 0.9062 - val_loss: 0.4061 - val_accuracy: 0.8630
Epoch 67/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2205 -
accuracy: 0.9075 - val_loss: 0.4095 - val_accuracy: 0.8645
Epoch 68/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2193 -
accuracy: 0.9089 - val_loss: 0.4139 - val_accuracy: 0.8638
Epoch 69/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2176 -
accuracy: 0.9095 - val_loss: 0.4058 - val_accuracy: 0.8645
Epoch 70/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2176 -
accuracy: 0.9097 - val_loss: 0.4177 - val_accuracy: 0.8663
Epoch 71/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2160 -
accuracy: 0.9098 - val_loss: 0.4051 - val_accuracy: 0.8648
Epoch 72/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2170 -
accuracy: 0.9084 - val_loss: 0.4125 - val_accuracy: 0.8641
Epoch 73/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2187 -
accuracy: 0.9070 - val_loss: 0.4292 - val_accuracy: 0.8620
Epoch 74/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2194 -
accuracy: 0.9090 - val_loss: 0.4141 - val_accuracy: 0.8660
Epoch 75/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2091 -
accuracy: 0.9140 - val_loss: 0.4212 - val_accuracy: 0.8634
Epoch 76/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2176 -
accuracy: 0.9111 - val_loss: 0.4238 - val_accuracy: 0.8658
Epoch 77/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2095 -
accuracy: 0.9128 - val_loss: 0.4153 - val_accuracy: 0.8629
Epoch 78/100
```

```
128/128 [==============================] - 1s 5ms/step - loss: 0.2122 -
accuracy: 0.9106 - val_loss: 0.4362 - val_accuracy: 0.8655
Epoch 79/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2130 -
accuracy: 0.9106 - val_loss: 0.4464 - val_accuracy: 0.8645
Epoch 80/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2093 -
accuracy: 0.9116 - val_loss: 0.4349 - val_accuracy: 0.8648
Epoch 81/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2078 -
accuracy: 0.9126 - val_loss: 0.4406 - val_accuracy: 0.8647
Epoch 82/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2080 -
accuracy: 0.9135 - val_loss: 0.4395 - val_accuracy: 0.8655
Epoch 83/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2091 -
accuracy: 0.9116 - val_loss: 0.4393 - val_accuracy: 0.8625
Epoch 84/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2127 -
accuracy: 0.9100 - val_loss: 0.4454 - val_accuracy: 0.8631
Epoch 85/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2040 -
accuracy: 0.9150 - val_loss: 0.4526 - val_accuracy: 0.8615
Epoch 86/100
128/128 [==============================] - 1s 5ms/step - loss: 0.1993 -
accuracy: 0.9154 - val_loss: 0.4465 - val_accuracy: 0.8631
Epoch 87/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2068 -
accuracy: 0.9139 - val_loss: 0.4584 - val_accuracy: 0.8615
Epoch 88/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2043 -
accuracy: 0.9129 - val_loss: 0.4618 - val_accuracy: 0.8631
Epoch 89/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2040 -
accuracy: 0.9141 - val_loss: 0.4726 - val_accuracy: 0.8599
Epoch 90/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2016 -
accuracy: 0.9153 - val_loss: 0.4792 - val_accuracy: 0.8606
Epoch 91/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2067 -
accuracy: 0.9115 - val_loss: 0.5034 - val_accuracy: 0.8601
Epoch 92/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2034 -
accuracy: 0.9128 - val_loss: 0.4940 - val_accuracy: 0.8594
Epoch 93/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2012 -
accuracy: 0.9152 - val_loss: 0.5026 - val_accuracy: 0.8635
Epoch 94/100
```

```
128/128 [==============================] - 1s 5ms/step - loss: 0.2020 -
accuracy: 0.9151 - val_loss: 0.4936 - val_accuracy: 0.8633
Epoch 95/100
128/128 [==============================] - 1s 5ms/step - loss: 0.1997 -
accuracy: 0.9165 - val_loss: 0.5086 - val_accuracy: 0.8633
Epoch 96/100
128/128 [==============================] - 1s 5ms/step - loss: 0.2020 -
accuracy: 0.9151 - val_loss: 0.5062 - val_accuracy: 0.8610
Epoch 97/100
128/128 [==============================] - 1s 5ms/step - loss: 0.1945 -
accuracy: 0.9188 - val_loss: 0.4838 - val_accuracy: 0.8615
Epoch 98/100
128/128 [==============================] - 1s 5ms/step - loss: 0.1973 -
accuracy: 0.9167 - val_loss: 0.5016 - val_accuracy: 0.8600
Epoch 99/100
128/128 [==============================] - 1s 5ms/step - loss: 0.1959 -
accuracy: 0.9188 - val_loss: 0.4957 - val_accuracy: 0.8585
Epoch 100/100
128/128 [==============================] - 1s 5ms/step - loss: 0.1984 -
accuracy: 0.9168 - val_loss: 0.5175 - val_accuracy: 0.8584
```

[59]:
```
plt.figure(figsize=(15,5))
sns.lineplot(data=history.history, dashes=False)
```

[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f00b008f5d0>
```



4.6.2 Valutazione modello

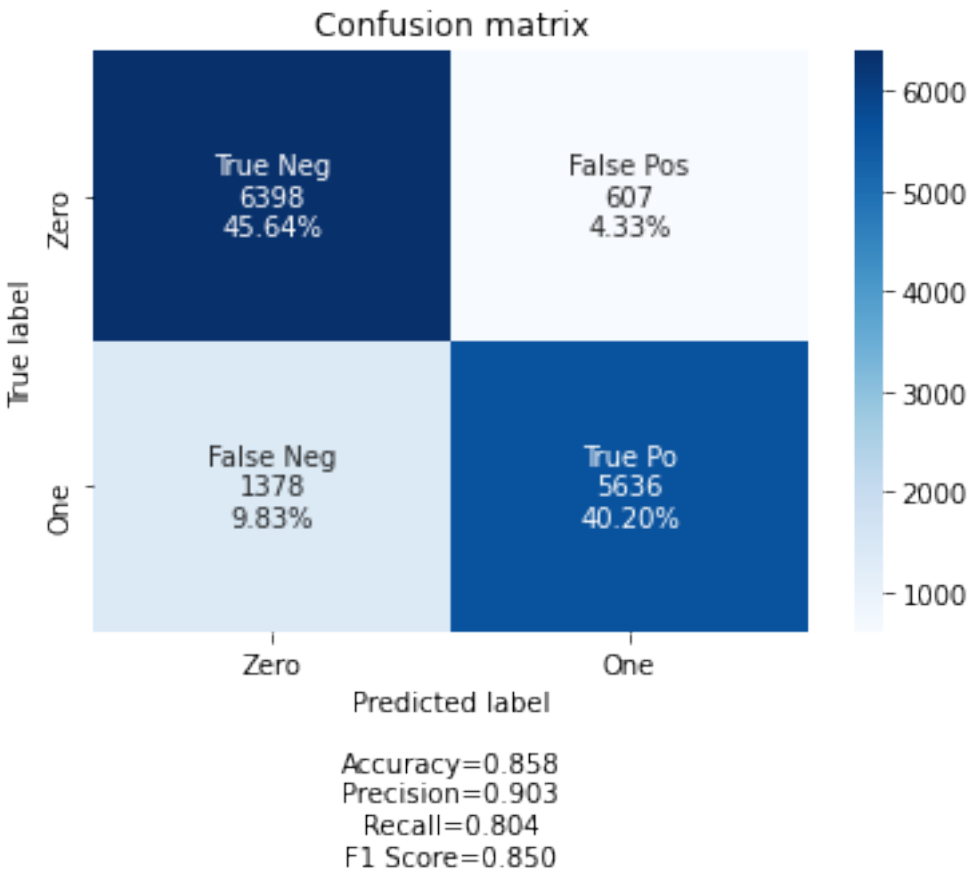
[60]:
```
models["Artificial Neural Network"] = {
    'conf_interval': conf_interval(model, X_val, y_val, level=confidenceLevel),
    'f1-score': f1_score(y_val, model.predict(X_val))
}
print(f'F1-score del modello: {models["Artificial Neural Network"]["f1-score"]:.
 ↪2%}')
```

```
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["Artificial Neural Network"]["conf_interval"]}')
```

```
/usr/local/lib/python3.7/dist-packages/keras/engine/sequential.py:450:
UserWarning: `model.predict_classes()` is deprecated and will be removed after
2021-01-01. Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if
your model does multi-class classification   (e.g. if it uses a `softmax` last-
layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model
does binary classification   (e.g. if it uses a `sigmoid` last-layer
activation).
  warnings.warn('`model.predict_classes()` is deprecated and '

55/55 [==============================] - 0s 2ms/step
55/55 [==============================] - 0s 1ms/step
F1-score del modello: 85.03%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.8442654292488218,
0.8560782300744377)
```

[61]:
```
y_pred = model.predict(X_val)
y_pred = (y_pred > 0.5)
conf_matr = confusion_matrix(y_val, y_pred)
labels = ['True Neg','False Pos','False Neg','True Po']
categories = ['Zero', 'One']
make_confusion_matrix(conf_matr,
                      group_names=labels,
                      categories=categories,
                      cmap='Blues',
                      title='Confusion matrix')
```

```
55/55 [==============================] - 0s 2ms/step

/usr/local/lib/python3.7/dist-packages/keras/engine/sequential.py:450:
UserWarning: `model.predict_classes()` is deprecated and will be removed after
2021-01-01. Please use instead:* `np.argmax(model.predict(x), axis=-1)`,   if
your model does multi-class classification   (e.g. if it uses a `softmax` last-
layer activation).* `(model.predict(x) > 0.5).astype("int32")`,   if your model
does binary classification   (e.g. if it uses a `sigmoid` last-layer
activation).
  warnings.warn('`model.predict_classes()` is deprecated and '
```



4.6.3 Ottimizzazione modello (TPU richiesta)

Per prevenire il riavvio del runtime, a causa della selezione della TPU, non eseguire queste celle: andrebbero perse le metriche di confronto tra modelli. Se si vuole comunque procedere con l'esecuzione delle seguenti celle ricordarsi di cambiare runtime da `Runtime` -> `Cambia tipo di runtime` e selezionare `TPU` sotto la voce `Accelerazione hardware`.

Setup TPU

[22]:
```
# Source: https://blog.tensorflow.org/2019/01/keras-on-tpus-in-colab.html

use_tpu = True #@param {type:"boolean"}

if use_tpu:
    assert 'COLAB_TPU_ADDR' in os.environ, 'Missing TPU; did you request a TPU␣
 ↪in Notebook Settings?'

if 'COLAB_TPU_ADDR' in os.environ:
  TF_MASTER = 'grpc://{}'.format(os.environ['COLAB_TPU_ADDR'])
else:
```

```
  TF_MASTER=''

tpu_address = TF_MASTER
```

[23]:
```
resolver = tf.distribute.cluster_resolver.TPUClusterResolver(TF_MASTER)
tf.config.experimental_connect_to_cluster(resolver)
tf.tpu.experimental.initialize_tpu_system(resolver)
strategy = tf.distribute.experimental.TPUStrategy(resolver)
```

```
INFO:tensorflow:Initializing the TPU system: grpc://10.120.233.242:8470

INFO:tensorflow:Initializing the TPU system: grpc://10.120.233.242:8470

INFO:tensorflow:Clearing out eager caches

INFO:tensorflow:Clearing out eager caches

INFO:tensorflow:Finished initializing TPU system.

INFO:tensorflow:Finished initializing TPU system.
WARNING:absl:`tf.distribute.experimental.TPUStrategy` is deprecated, please use
the non experimental symbol `tf.distribute.TPUStrategy` instead.

INFO:tensorflow:Found TPU system:

INFO:tensorflow:Found TPU system:

INFO:tensorflow:*** Num TPU Cores: 8

INFO:tensorflow:*** Num TPU Cores: 8

INFO:tensorflow:*** Num TPU Workers: 1

INFO:tensorflow:*** Num TPU Workers: 1

INFO:tensorflow:*** Num TPU Cores Per Worker: 8

INFO:tensorflow:*** Num TPU Cores Per Worker: 8

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:localhost/replica:0/task:0/device:CPU:0, CPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:localhost/replica:0/task:0/device:CPU:0, CPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:CPU:0, CPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:CPU:0, CPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:0, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:0, TPU, 0, 0)
```

```
INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:1, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:1, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:2, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:2, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:3, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:3, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:4, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:4, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:5, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:5, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:6, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:6, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:7, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU:7, TPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU_SYSTEM:0, TPU_SYSTEM,
0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:TPU_SYSTEM:0, TPU_SYSTEM,
0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:XLA_CPU:0, XLA_CPU, 0, 0)

INFO:tensorflow:*** Available Device:
_DeviceAttributes(/job:worker/replica:0/task:0/device:XLA_CPU:0, XLA_CPU, 0, 0)
```

Grid search e training modello

[24]:
```
def create_model(dense_layer_sizes, optimizer='adam', dropout=0.1,␣
↪init='uniform', dense_nparams=256):
    with strategy.scope():
      model = Sequential()
      model.add(Dense(dense_nparams, activation=tf.nn.relu, input_dim=X_train.
↪shape[1], kernel_initializer=init,))
      model.add(Dropout(dropout), )
      for layer_size in dense_layer_sizes:
          model.add(Dense(layer_size, activation=tf.nn.relu))
          model.add(Dropout(dropout), )
      model.add(Dense(1, activation=tf.nn.softmax))
      model.compile(optimizer=tf.keras.optimizers.Adagrad(learning_rate=0.1),
                  loss=['binary_crossentropy'],
                  metrics=['accuracy'])
    return model
```

[36]:
```
small_param_grid = {
    'kc__epochs': [5],
    'kc__dense_nparams': [32, 256, 512],
    'kc__init': [ 'zeros'],
    'kc__batch_size':[16],
    'kc__optimizer':['RMSprop', 'Adam', 'sgd'],
    'kc__dropout': [0.5, 0.4, 0.3, 0.2, 0.1, 0],
    'kc__dense_layer_sizes': [(50, 40, 30, 20, 10)]
}

param_grid = {
    'kc__epochs': [2, 3],
    'kc__dense_nparams': [32, 64, 128, 256, 512, 1024, 2048],
    'kc__init': [ 'uniform', 'zeros', 'normal'],
    'kc__batch_size':[2, 16, 32, 64, 128, 256],
    'kc__optimizer':['RMSprop', 'Adam', 'Adamax', 'sgd'],
    'kc__dropout': [0.6, 0.7, 0.8, 0.9, 0.5, 0.4, 0.3, 0.2, 0.1, 0],
    'kc__dense_layer_sizes': [(32, 64, 128, 256, 512, 1024, 2048)],
}

mode_path = '../models/best_ann.h5'
callbacks = [ModelCheckpoint(filepath=mode_path, save_best_only=True)]

estimator = Pipeline([
                    ('kc', KerasClassifier(build_fn=create_model, verbose=1,
                                           validation_data=(X_val, y_val),␣
↪callbacks=callbacks))
                    ])

grid = GridSearchCV(estimator=estimator,
```

```
                    n_jobs=-1,
                    verbose=5,
                    return_train_score=True,
                    cv=5,
                    param_grid=param_grid)
```

[ ]:
```
grid_result = grid.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 54 candidates, totalling 270 fits

[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
[Parallel(n_jobs=-1)]: Done  14 tasks      | elapsed:    4.9s
[Parallel(n_jobs=-1)]: Done 180 tasks      | elapsed:   10.9s
[Parallel(n_jobs=-1)]: Done 270 out of 270 | elapsed:   13.5s finished
```

[ ]:
```
grid.best_params_
```

[ ]:
```
score_best_ann = f1_score(y_val, y_pred.ravel())
print(f'F1-score della migliore ANN: {score_best_ann:.1%}')
print("Best score: %f using params %s" % (grid_result.best_score_, grid_result.
 ↪best_params_))
```

4.7 Valutazione finale modelli

Si paragonano gli intervalli di confidenza (al 95%) relativi agli F1-score di tutti i modelli provati nel notebook. Il modello migliore si è rivelato essere una regressione logistica usando tutte le features a disposizione. Si evidenza come tutti i modelli del notebook abbiano differenza statisticamente significativa rispetto ad un modello random.

[62]:
```
random = DummyClassifier(strategy="uniform", random_state=42)
random.fit(X_train, y_train)

models["Dummy classifier"] = {
    'conf_interval': conf_interval(random, X_val, y_val, level=confidenceLevel),
    'f1-score': f1_score(y_val, random.predict(X_val))
}
print(f'F1-score del modello: {models["Dummy classifier"]["f1-score"]:.2%}')
print(f'Intervallo di confidenza al {confidenceLevel:.1%} raggiunto dal modello:
 ↪ \
  {models["Dummy classifier"]["conf_interval"]}')
```

```
F1-score del modello: 50.16%
Intervallo di confidenza al 95.0% raggiunto dal modello:   (0.49332758401437193,
0.5098787214924633)
```

[63]:
```
print(f"{'Modello':<40}{'Intervallo di confidenza F1-score':<15}")
for k in models.keys():
  print(f'{k:<40}{str(models[k]["conf_interval"]):<15}')
```

```
Modello                                  Intervallo di confidenza F1-score
Perceptron                               (0.8323907730095819, 0.8445713562341648)
Logistic regression important features   (0.6599829478534277, 0.6755742566993698)
Logistic regression all features         (0.8587529364230313, 0.8700835962282556)
SVM                                      (0.8566398399526858, 0.868043265861745)
Decision tree                            (0.829962923808452, 0.8422158805949433)
XGBoost                                  (0.8559308097784665, 0.8673584585360085)
Artificial Neural Network                (0.8442654292488218, 0.8560782300744377)
Dummy classifier                         (0.49332758401437193,
0.5098787214924633)
```

[64]:
```
print(f"{'Dummy classifier vs modello':<45}{'Significativamente diverso?':<35}")
for k in models.keys():
  print(f'Dummy Classifier vs {k:<45}{str(different_model(\
      models[k]["f1-score"], models["Dummy classifier"]["f1-score"], X_val,␣
 ↪level=0.95)):<35}')
```

```
Dummy classifier vs modello                  Significativamente diverso?
Dummy Classifier vs Perceptron
(0.32669418168314796, 0.3472462633000783)
Dummy Classifier vs Logistic regression important features
(0.15485038345475338, 0.1775915857312121)
Dummy Classifier vs Logistic regression all features
(0.3528846467368758, 0.372944416175935)
Dummy Classifier vs SVM
(0.35078675735732495, 0.370887740604773)
Dummy Classifier vs Decision tree
(0.3242803641040246, 0.3448754453215812)
Dummy Classifier vs XGBoost
(0.3500827680319159, 0.3701975104936143)
Dummy Classifier vs Artificial Neural Network
(0.3384961174759701, 0.3588322645476768)
Dummy Classifier vs Dummy classifier
(-0.011705024895070297, 0.011705024895070297)
```