SAPIENZA
UNIVERSITÀ DI ROMA

...

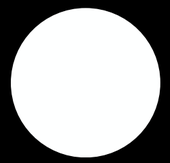
TELCO CUSTOMER CHURN DATASET

GIULIANO BISINELLA 2139735
BERNARDO CASAROSA 2152655
STELLA SOFIA CILLIS 2141304

IL PROBLEMA

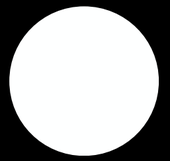
Classificazione

Input



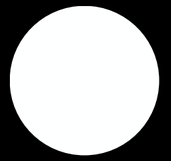
DATI ETEROGENEI (FINANZIARI, TECNICI, DEMOGRAFICI).

Task



CLASSIFICAZIONE BINARIA (YES/NO).

Vincolo



CLASSI SBILANCIATE (DATA IMBALANCE)

IL DATASET

Telco Customer Churn

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Gruppo demografico

Gruppo tecnico

Gruppo finanziario

PRE PROCESSING E FEATURE ENGINEERING

Parte 1

PULIZIA DEL DATASET

```
# 1. La feature "customerID"
if "customerID" in data.columns: data.drop(columns=["customerID"], inplace=True)

# 2. La feature "TotalCharges" è il prodotto di "MonthlyCharges" e "tenure"
if "TotalCharges" in data.columns: data.drop(columns=["TotalCharges"], inplace=True)
```

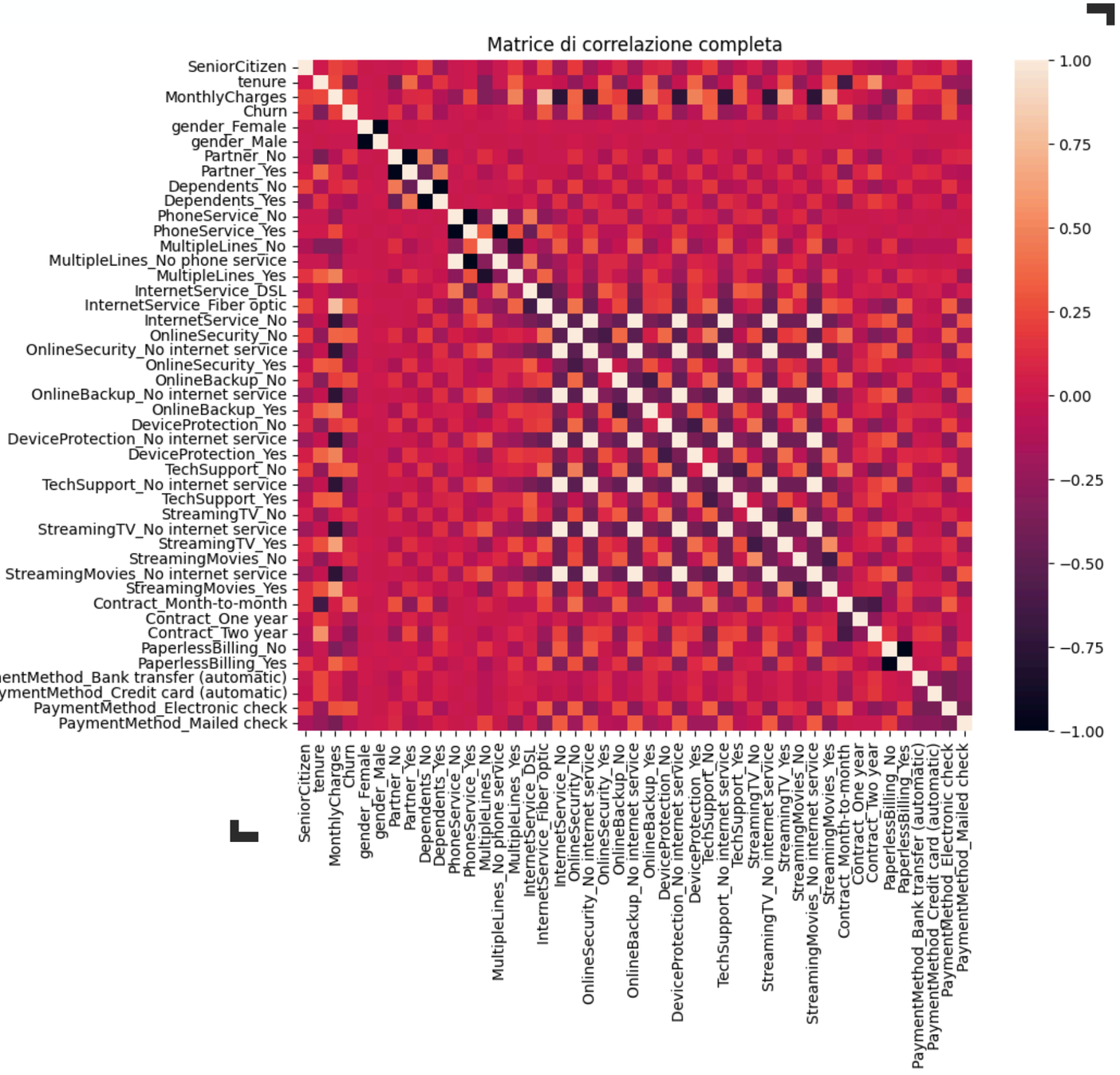
- Eliminazione delle variabili superflue
- Verifica dei valori nulli
- Operazione di One-hot encoding

```
valori_nulli = ["", "none", "n/a", "na", "null", "?"]
print("RICERCA VALORI NULLI - dtype object:")
print(data.select_dtypes(include="object").apply(lambda x:
x.astype(str).str.strip().str.lower().isin(valori_nulli).sum()))
print("\n")
print("RICERCA VALORI NULLI - dtype numerico:")
print(data.isna().sum())
```

```
data_dummy_provv = pd.get_dummies(data, drop_first=False, dtype=int)
```

ANALISI ESPLORATIVA DEI DATI

EDA

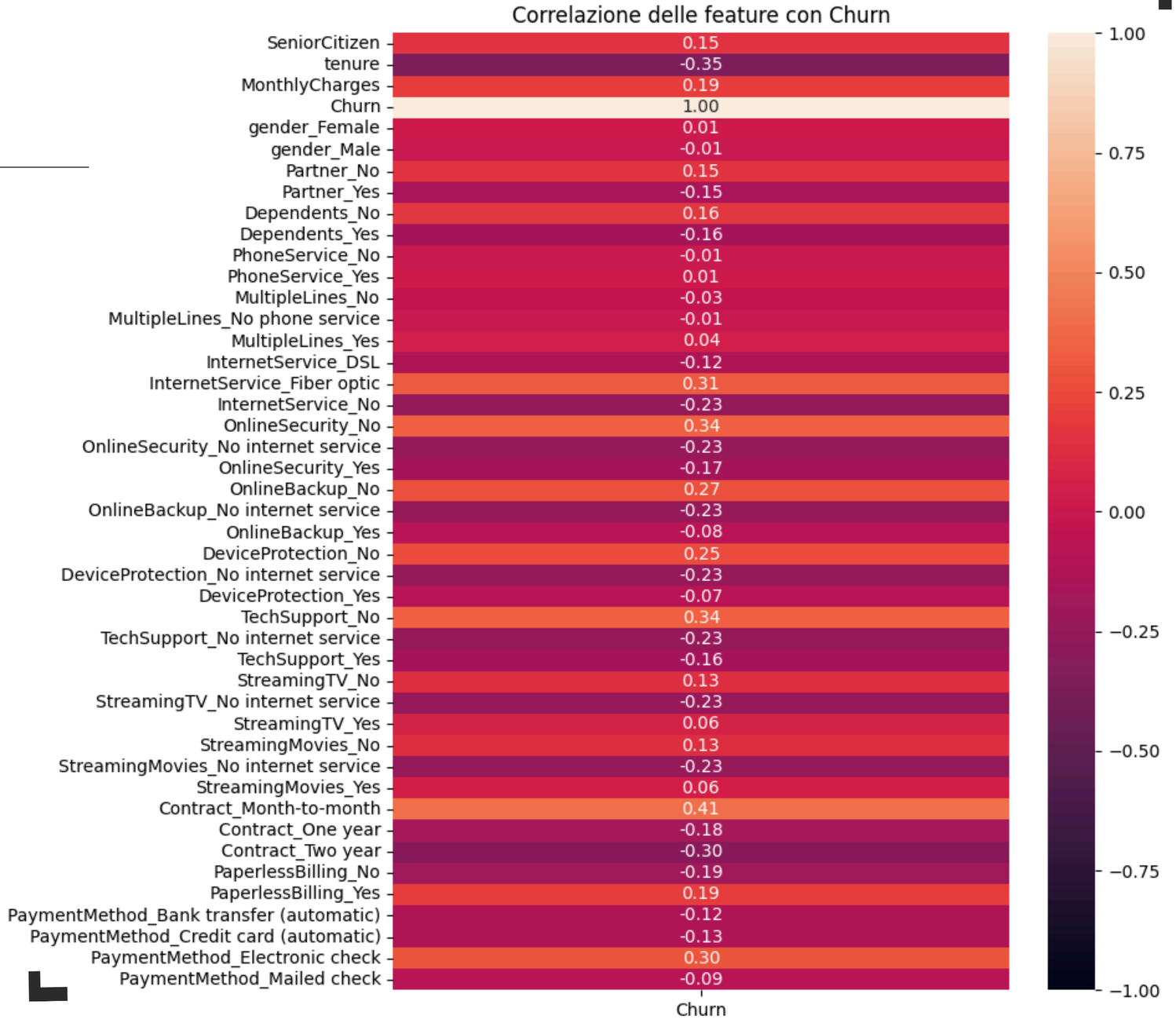


Heatmap completa

Osservazioni

- Mantenere “PhoneService_Yes” e “MultipleLines_No phone service” è ridondante.
- Mantenere tutti i valori che si concludono con "[...]_No internet service" è ridondante.

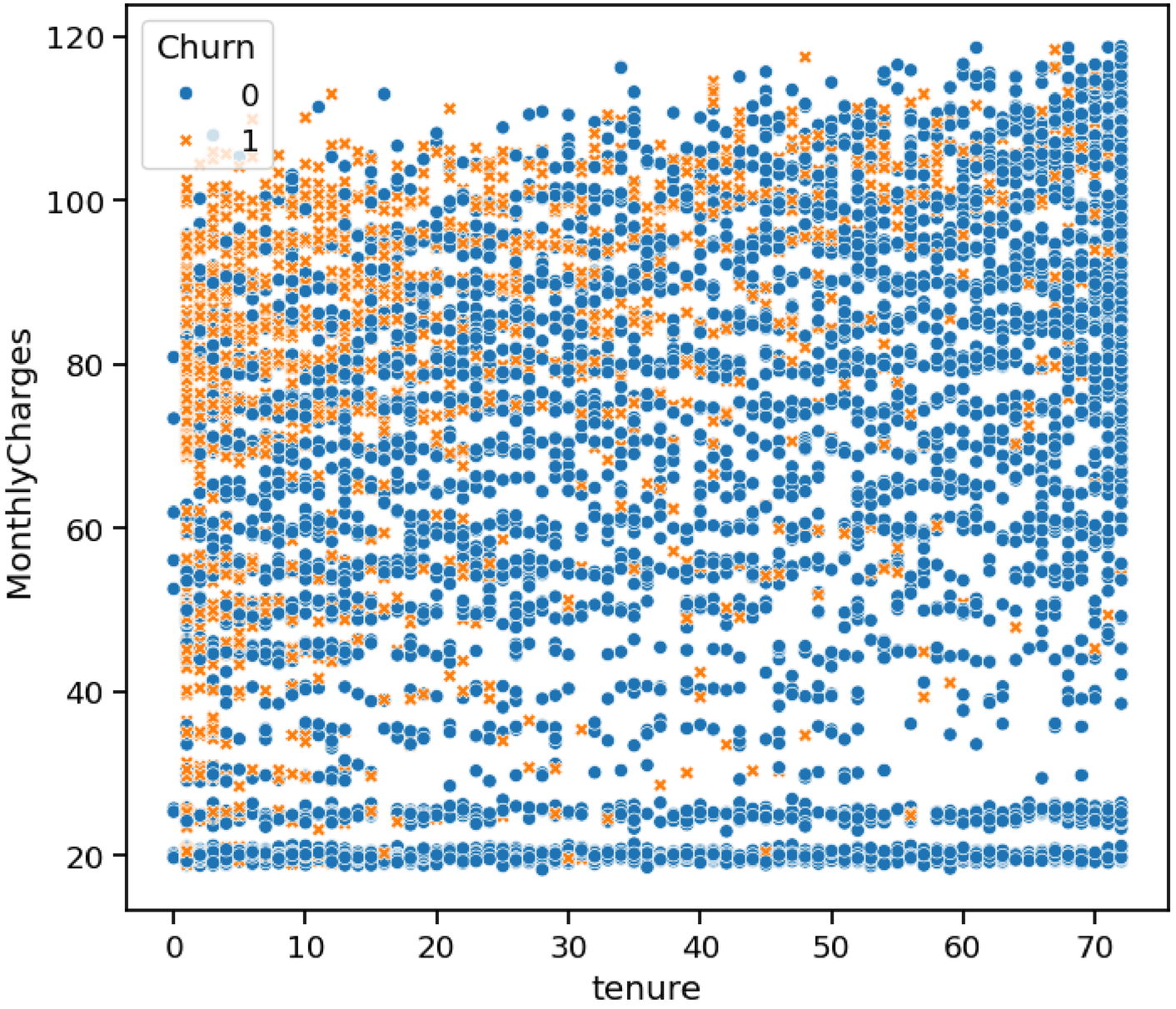
Heatmap della variabile target



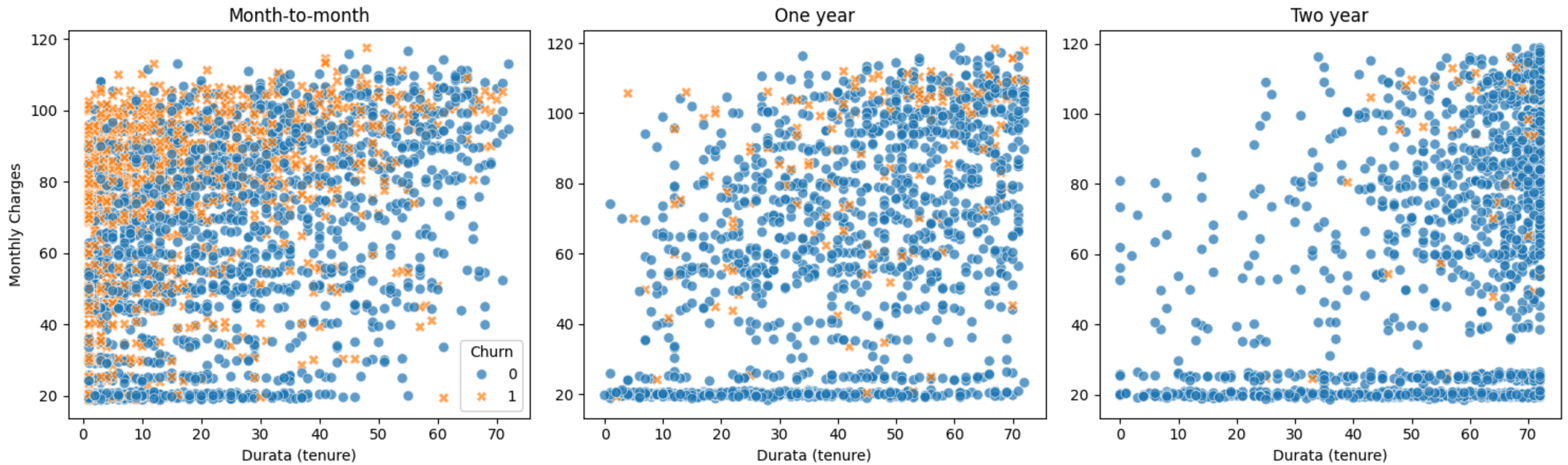
Scatter plot

Relazione tra tenure e Monthly charges rispetto alla variabile target

Relazione tra tenure e Monthly charges rispetto alla variabile target in correlazione con il tipo di contratto

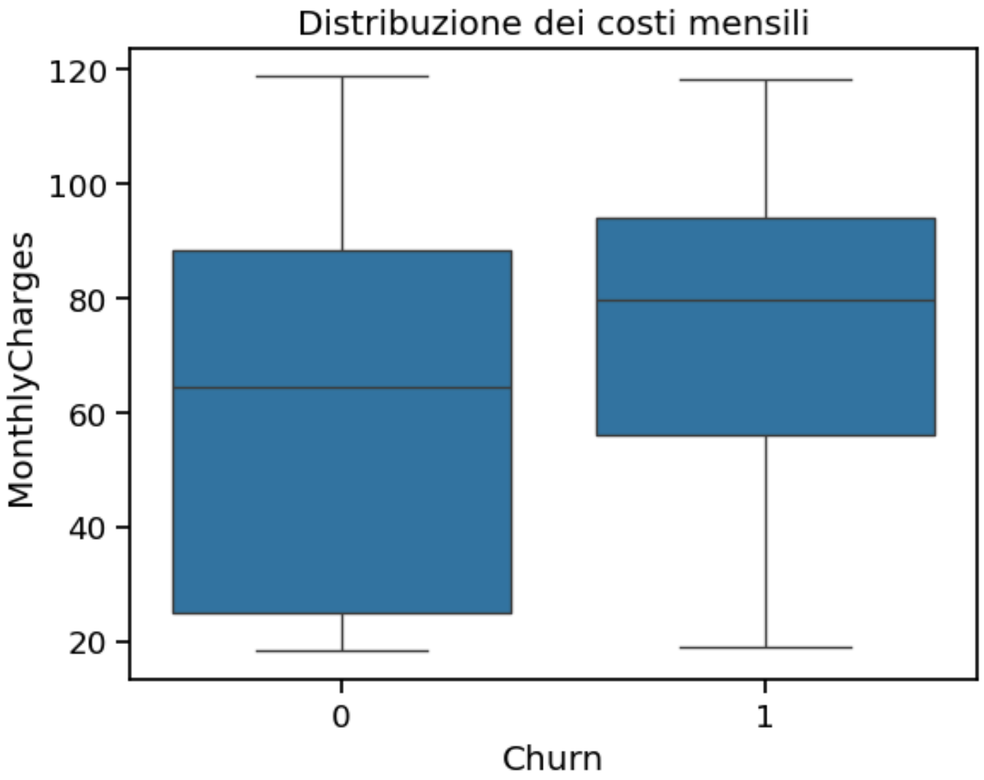

Churn
0
1
MonthlyCharges
120
100
80
60
40
20
0
10
20
30
40
50
60
70
tenure

```python
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

for i, contract_type in enumerate(data["Contract"].unique()):
    contract_data = data[data["Contract"] == contract_type]
    sns.scatterplot(data=contract_data, x="tenure", y="MonthlyCharges", hue="Churn", style="Churn",
    axes[i].set_title(contract_type)
    axes[i].set_xlabel("Durata (tenure)")

    if i == 0:
        axes[i].set_ylabel("Monthly Charges")
    else:
        axes[i].set_ylabel("")

    if i > 0:
        axes[i].get_legend().remove()

plt.suptitle("La durata del contratto impatta sulla permanenza del cliente?", y=1.05, fontsize=14)
plt.tight_layout()
plt.subplots_adjust(top=0.85)
plt.show()
```

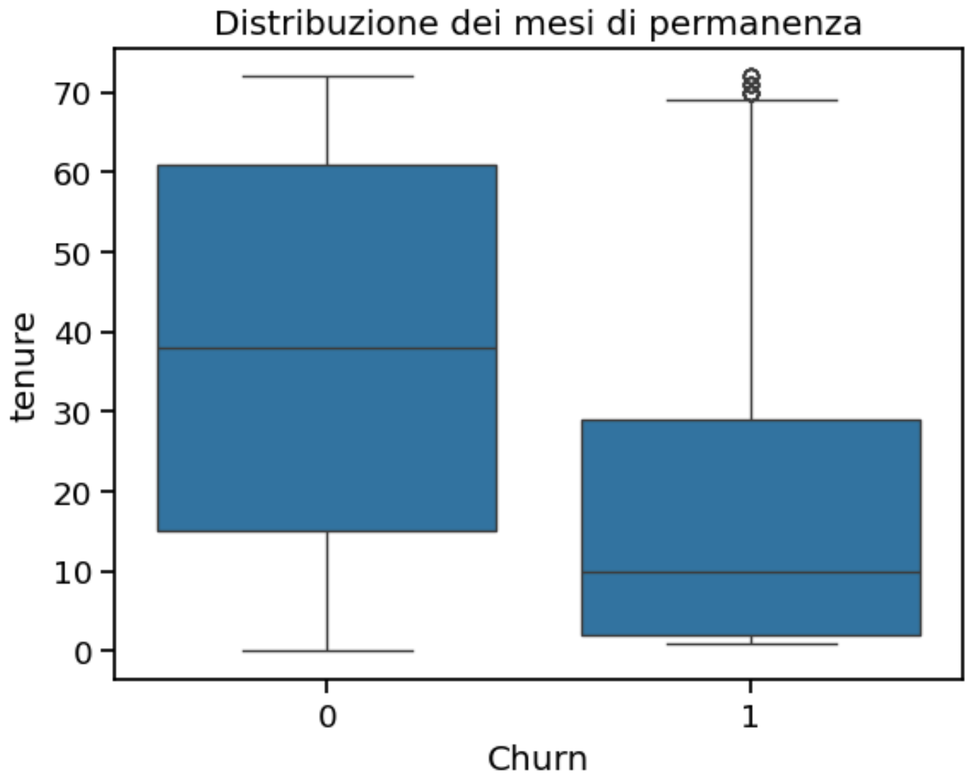


Boxplot

```
#### 1. MonthlyCharges
sns.boxplot(x='Churn', y='MonthlyCharges', data=data_dummy)
plt.title('Distribuzione dei costi mensili')
plt.show()
print("\n\n")
```




```
#### 2. tenure
sns.boxplot(x='Churn', y='tenure', data=data_dummy)
plt.title('Distribuzione dei mesi di permanenza')
plt.show()
```



PRE PROCESSING E FEATURE ENGINEERING

Parte 2

Verifica del bilanciamento delle classi

```
data.describe()[["Churn"]]
```

	Churn
count	7043.000000
mean	0.265370
std	0.441561
min	0.000000
25%	0.000000
50%	0.000000
75%	1.000000
max	1.000000

```
print("Distribuzione Variabile Target [Churn]:")
print((data["Churn"].value_counts()/len(data_dummy)) * 100)
```

```
Distribuzione Variabile Target [Churn]:
Churn
0    73.463013
1    26.536987
Name: count, dtype: float64
```

```
# Divisione del dataset training set e test set
X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size = 0.2,
                 random_state = 42, stratify = y)
```

Normalizzazione

STATISTICHE

	tenure	MonthlyCharges	SeniorCitizen	Churn
count	7043.000000	7043.000000	7043.000000	7043.000000
mean	32.371149	64.761692	0.162147	0.265370
std	24.559481	30.090047	0.368612	0.441561
min	0.000000	18.250000	0.000000	0.000000
25%	9.000000	35.500000	0.000000	0.000000
50%	29.000000	70.350000	0.000000	0.000000
75%	55.000000	89.850000	0.000000	1.000000
max	72.000000	118.750000	1.000000	1.000000

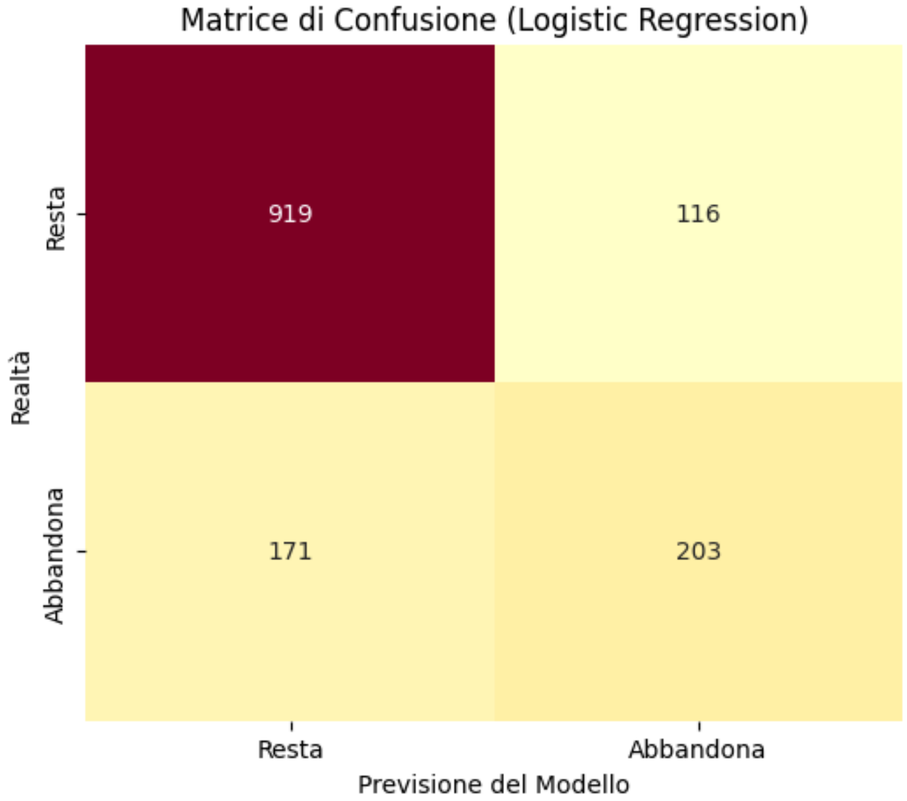
```
features_numeriche = ["tenure", "MonthlyCharges"]
scaler = MinMaxScaler()
X_train[features_numeriche] = scaler.fit_transform(X_train[features_numeriche])
X_test[features_numeriche] = scaler.transform(X_test[features_numeriche])
```

I MODELLI

Modello base

Modello base

```
RLogistica = LogisticRegression()
```

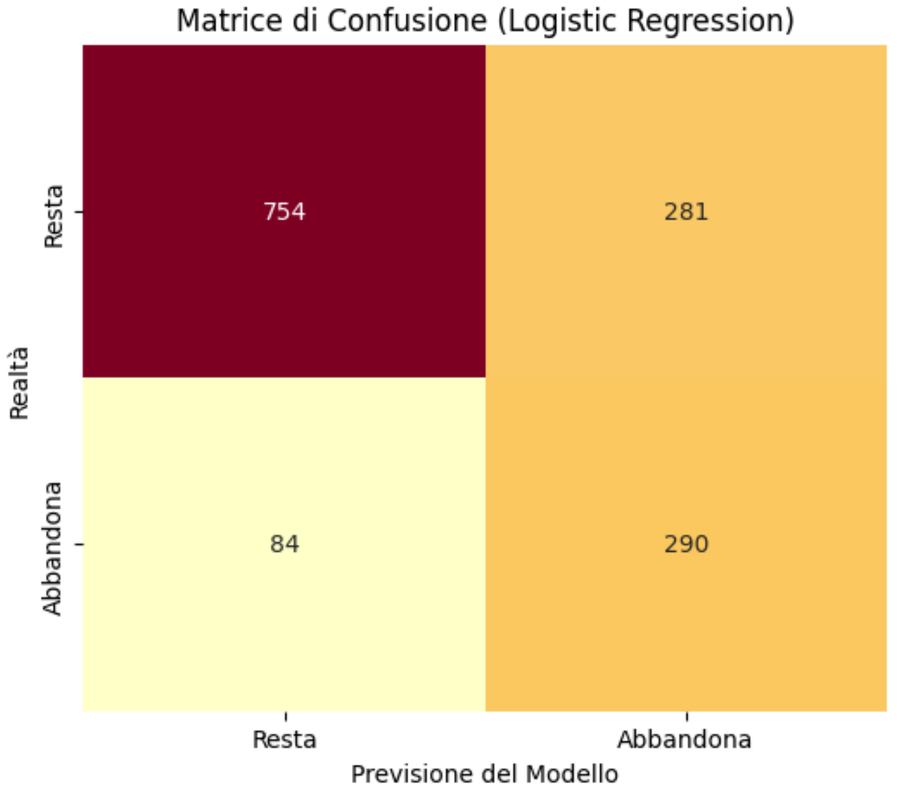


RIEPILOGO

	precision	recall	f1-score	support
0	0.84	0.89	0.86	1035
1	0.64	0.54	0.59	374
accuracy			0.80	1409
macro avg	0.74	0.72	0.73	1409
weighted avg	0.79	0.80	0.79	1409

Modello base bilanciato

```
RLogistica2 = LogisticRegression(class_weight='balanced', random_state=42)
```



RIEPILOGO

	precision	recall	f1-score	support
0	0.90	0.73	0.81	1035
1	0.51	0.78	0.61	374
accuracy			0.74	1409
macro avg	0.70	0.75	0.71	1409
weighted avg	0.80	0.74	0.75	1409

MODELLO CON SOGLIA OTTIMALE

1. CREIAMO IL TERZO MODELLO

```
RLOGISTICA3 = LOGISTICREGRESSION(CLASS_WEIGHT="BALANCED",
RANDOM_STATE=42)
```

2. RICORRIAMO ALLA CROSS-VALIDATION PER NON CALCOLARE LA SOGLIA INCLUDENDO IL TEST SET E CAUSANDO UNA DATA LEAKAGE

```
Y_PROB_CV = CROSS_VAL_PREDICT(RLOGISTICA3, X_TRAIN,
Y_TRAIN, CV=5, METHOD='PREDICT_PROBA')[:, 1]
```

3. CALCOLO SOGLIA OTTIMA

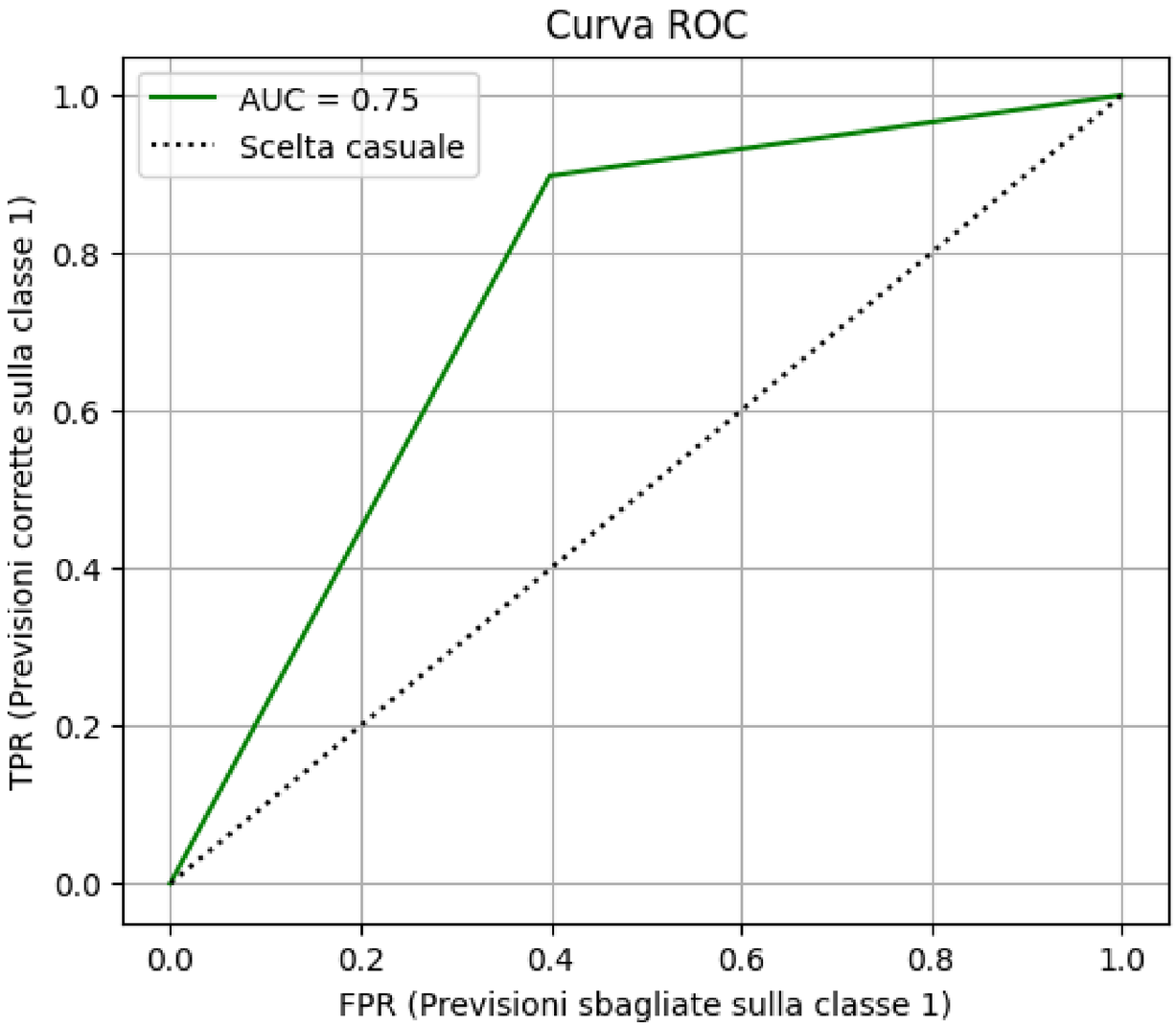
```
SOGLIE = NP.LINSPACE(0, 1, 100)
F2_SCORES = [FBETA_SCORE(Y_TRAIN, (Y_PROB_CV >= S).ASTYPE(INT),
BETA=2) FOR S IN SOGLIE]
SOGLIA_OTTIMA_1 = SOGLIE[NP.ARGMAX(F2_SCORES)]
```

4. FIT MODELLO

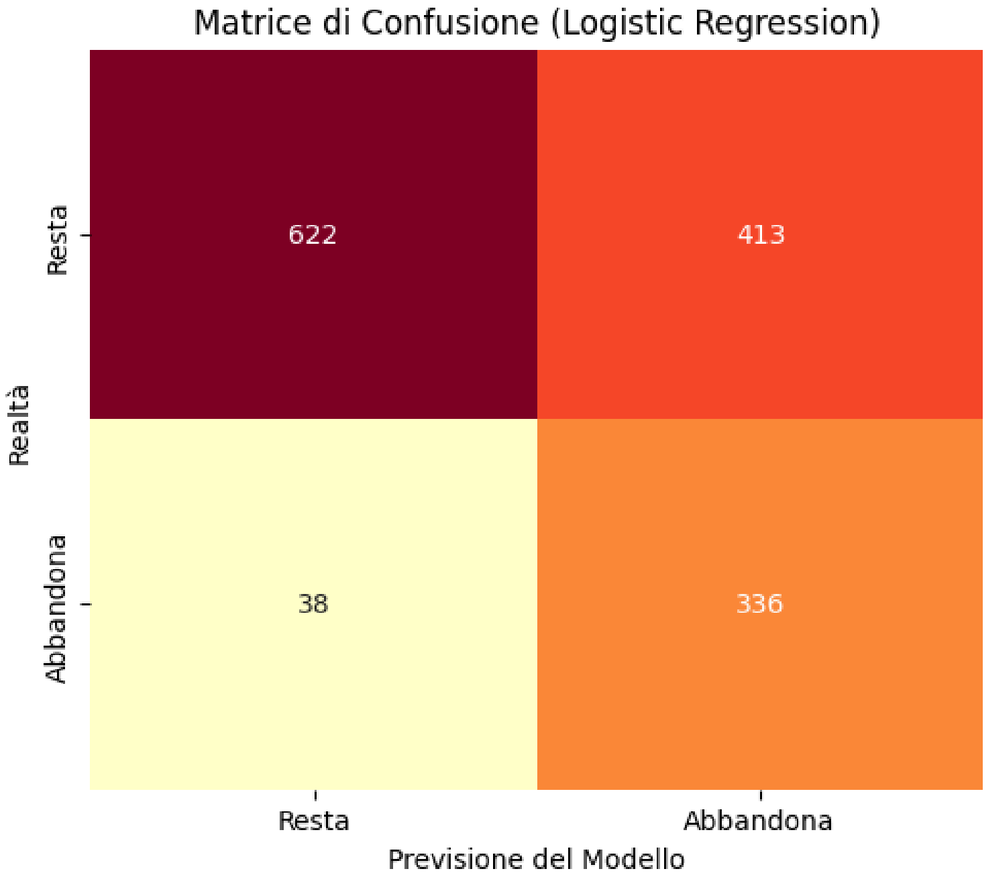
```
RLOGISTICA3.FIT(X_TRAIN, Y_TRAIN)
```

5. PREDIZIONE SUL TEST SET

```
Y_PROB = RLOGISTICA3.PREDICT_PROBA(X_TEST)[:, 1]
Y_SOGLIA = (Y_PROB >= SOGLIA_OTTIMA_1).ASTYPE(INT)
```



AUC SCORE
| 0.7497 |
SOGLIA OTTIMALE
| 0.3535 |



MODELLO CON SOGLIA OTTIMALE

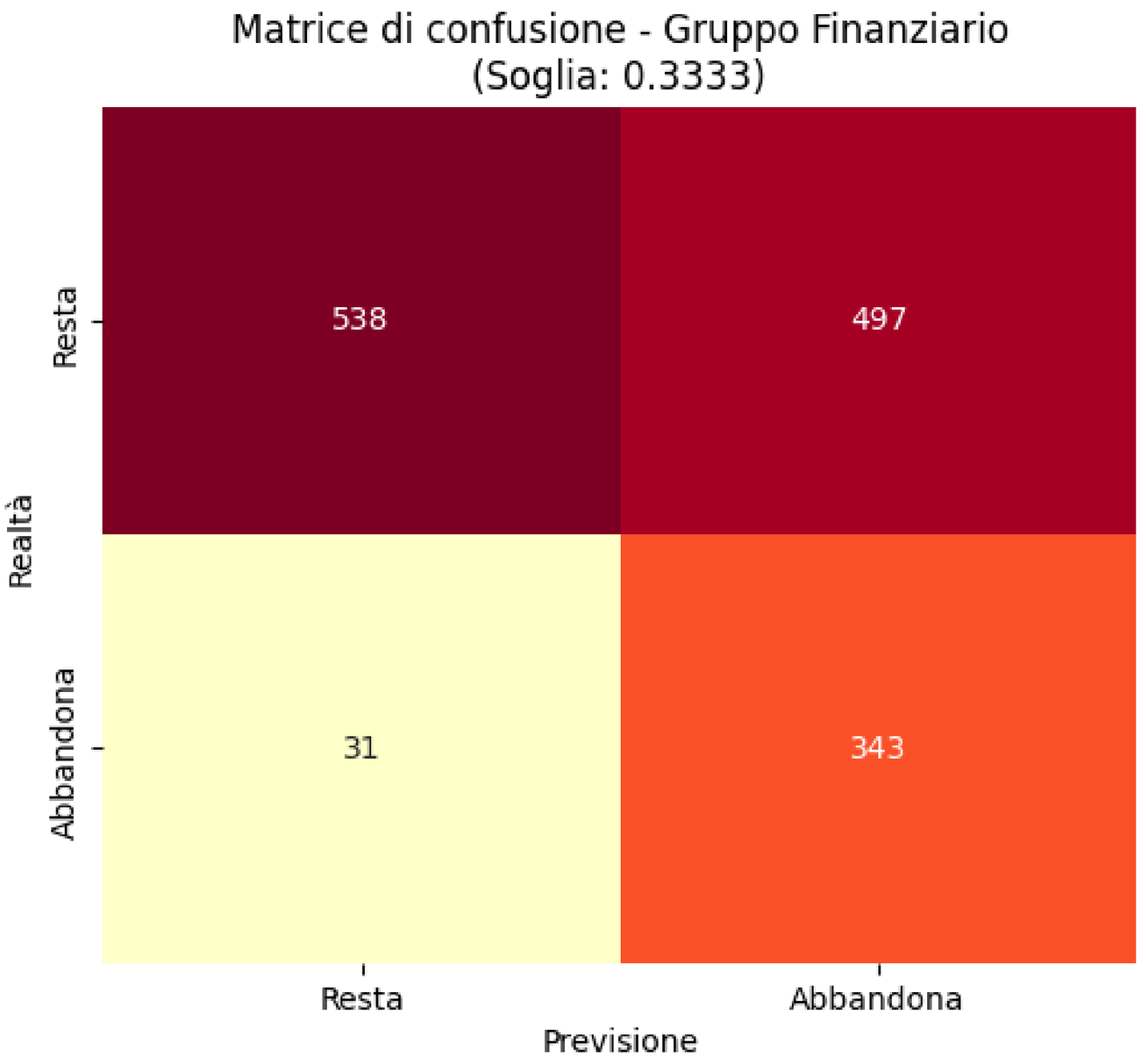
RIEPILOGO

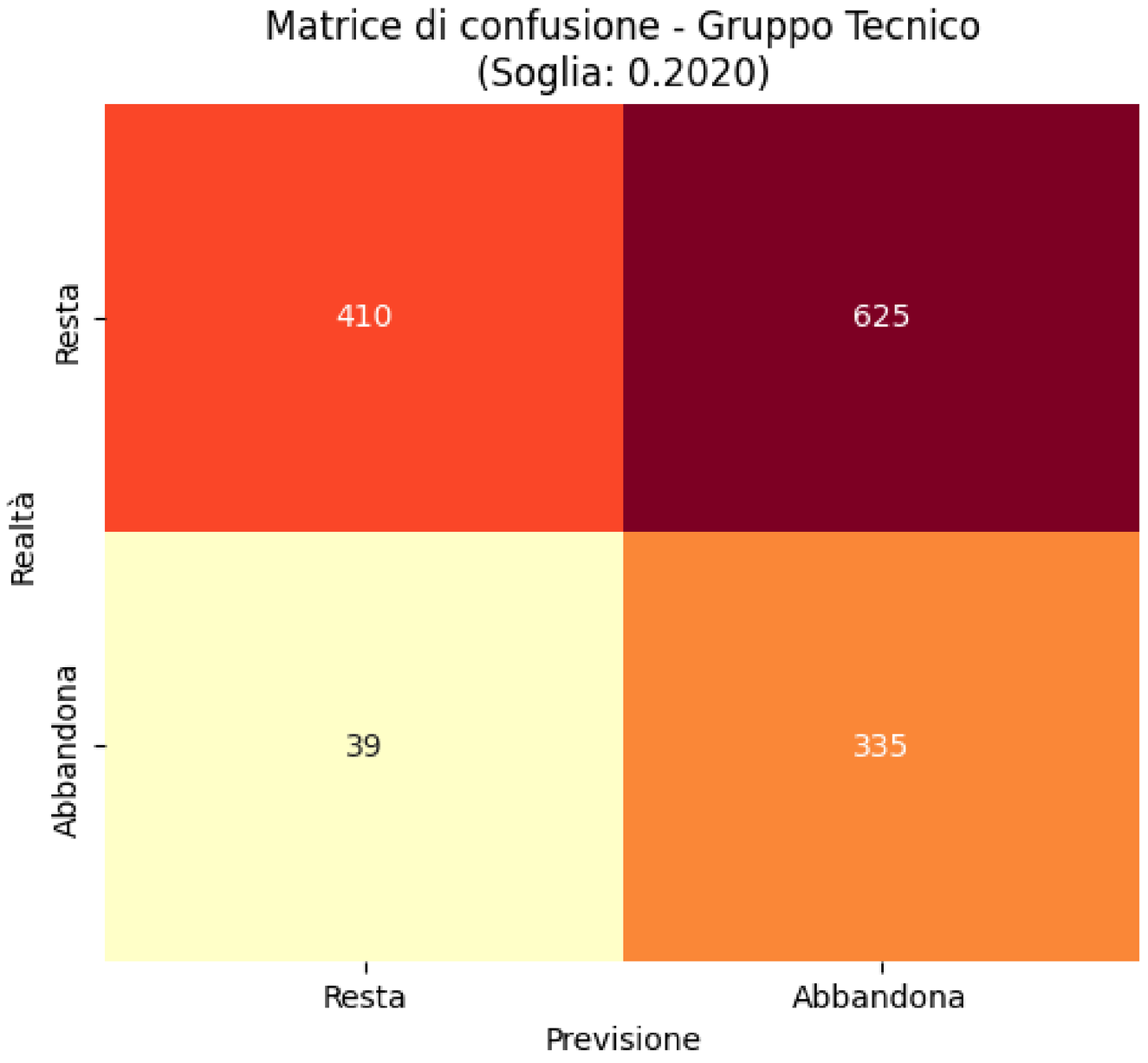
	precision	recall	f1-score	support
0	0.94	0.60	0.73	1035
1	0.45	0.90	0.60	374
accuracy			0.68	1409
macro avg	0.70	0.75	0.67	1409
weighted avg	0.81	0.68	0.70	1409

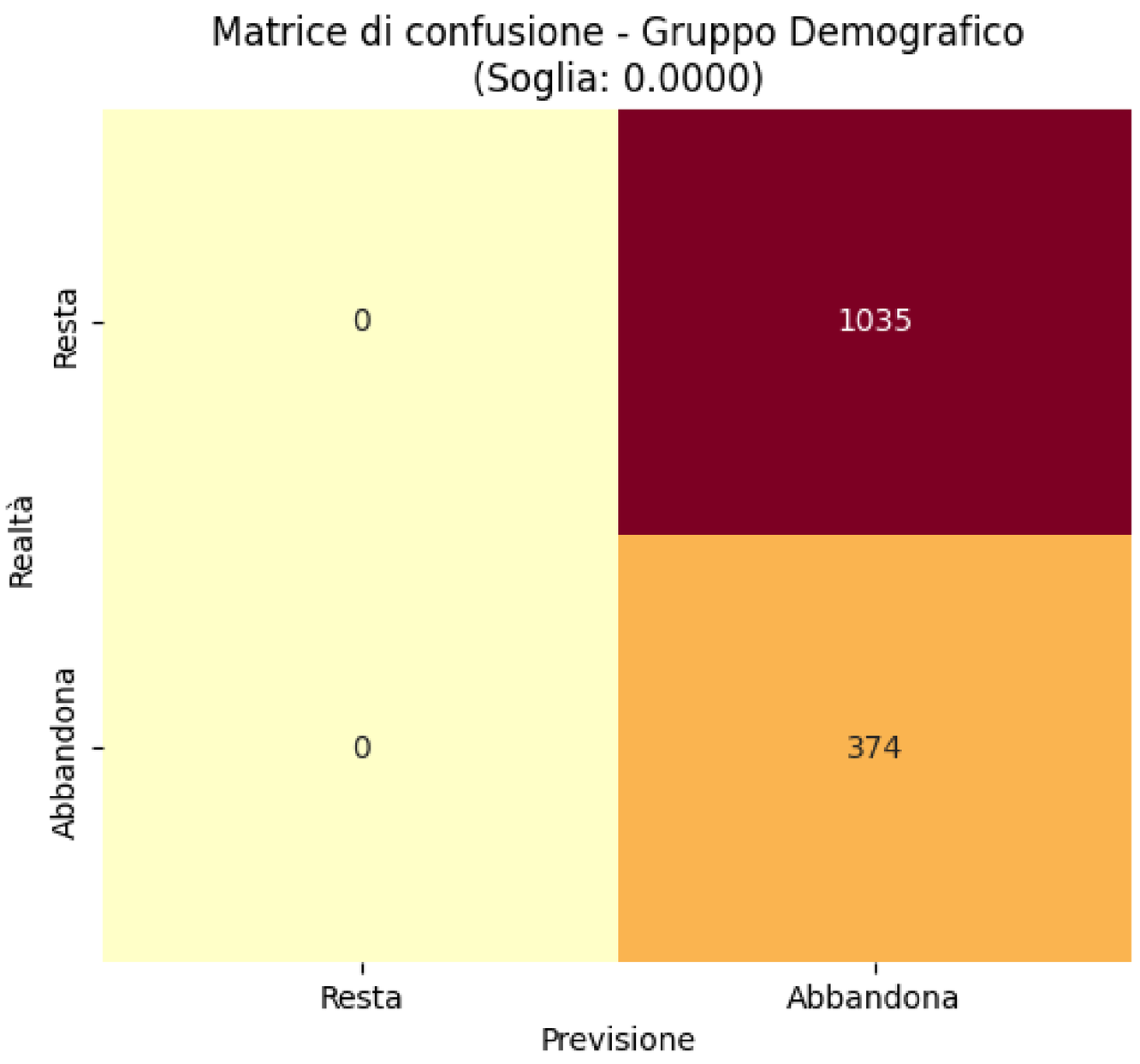
I MODELLI

Modelli tematici

GRUPPO TECNICO	["InternetService_Fiber optic", "InternetService_No", "OnlineSecurity_Yes", "TechSupport_Yes"]
GRUPPO FINANZIARIO	["MonthlyCharges", "Contract_One year", "Contract_Two year", "PaymentMethod_Electronic check", "PaperlessBilling_Yes"]
GRUPPO DEMOGRAFICO	["SeniorCitizen", "Partner_Yes", "Dependents_Yes", "gender_Male"]


Matrice di confusione - Gruppo Finanziario
(Soglia: 0.3333)
Realtà
Resta
Abbandona
538
497
31
343
Resta
Abbandona
Previsione


Matrice di confusione - Gruppo Tecnico
(Soglia: 0.2020)
Realtà
Resta
Abbandona
410
625
39
335
Resta
Abbandona
Previsione


Matrice di confusione - Gruppo Demografico
(Soglia: 0.0000)
Realtà
Resta
Abbandona
0
1035
0
374
Resta
Abbandona
Previsione

MODELLI	AUC	Recall	Precision	Soglia
Modello base non bilanciato	0.72	0.54	0.64	-
Modello base bilanciato	0.75	0.78	0.51	-
Modello base SO	0.75	0.90	0.45	0.35
Modello finanziario SO	0.79	0.92	0.41	0.33
Modello tecnico SO	0.75	0.90	0.35	0.20
Modello demografico SO	0.65	1.00	0.27	0.00

RISULTATI