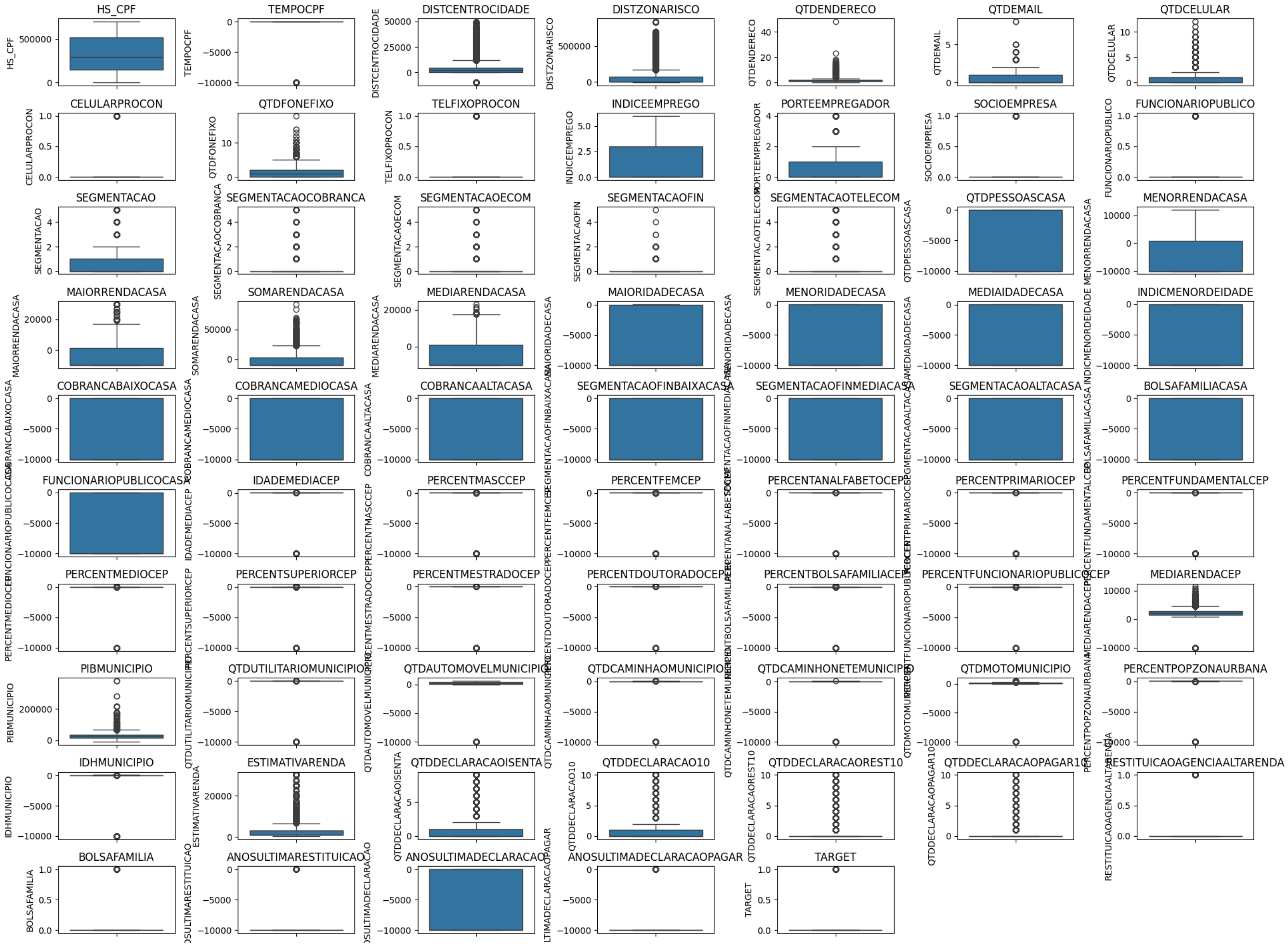
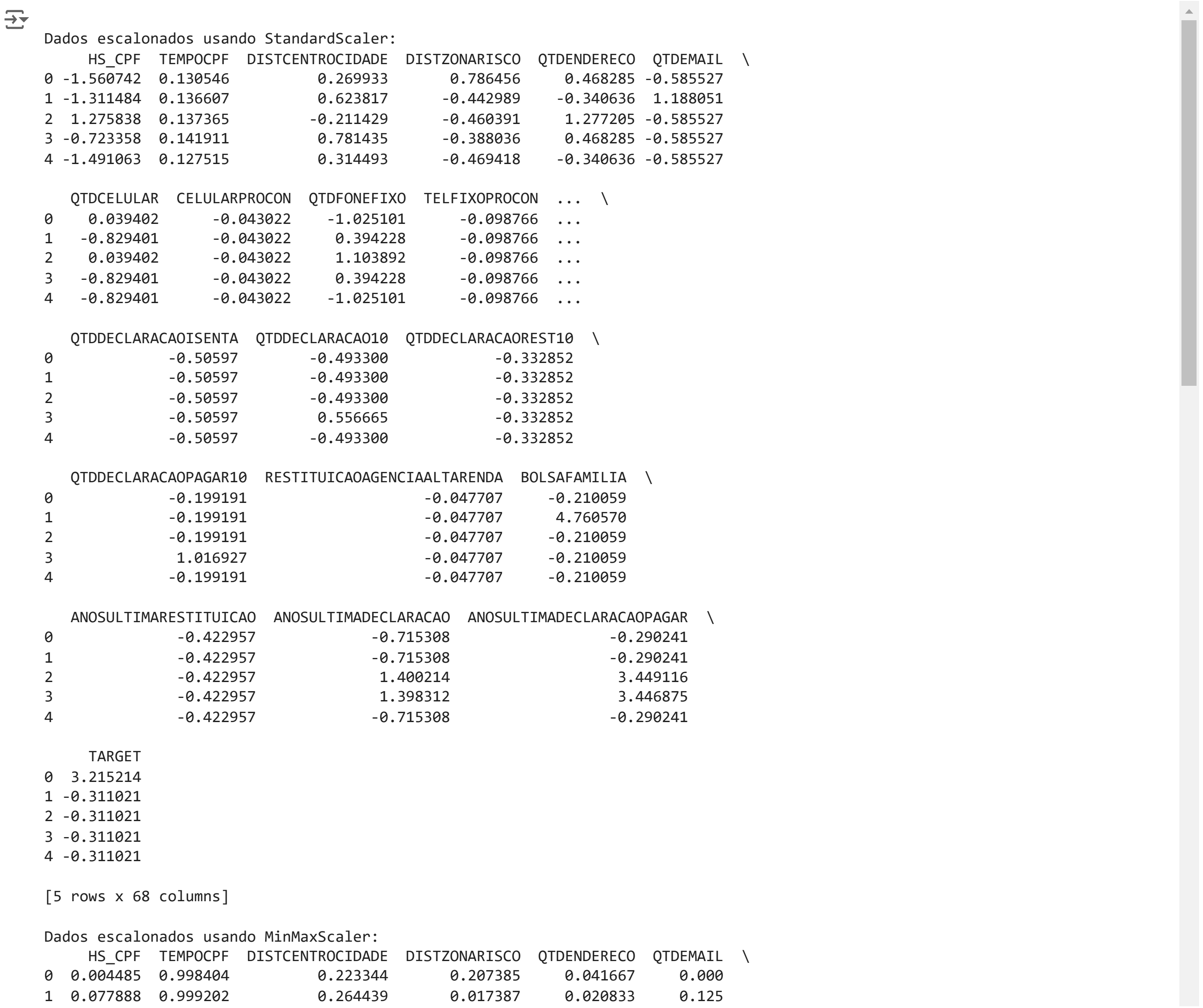
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
from sklearn.ensemble import IsolationForest
from sklearn.preprocessing import StandardScaler, MinMaxScaler, RobustScaler, PowerTransformer
import warnings
warnings.filterwarnings('ignore')


# Carregar dados
file_path = 'train.csv'
data = pd.read_csv(file_path)


# Separar colunas numéricas e categóricas
numeric_cols = data.select_dtypes(include=['int64', 'float64']).columns
categorical_cols = data.select_dtypes(include=['object']).columns


# Detecção de outliers com IsolationForest
isolation_forest = IsolationForest(contamination='auto', random_state=42)
outliers = isolation_forest.fit_predict(data[numeric_cols])


# Adicionar coluna de outliers ao dataframe
data['outlier'] = outliers


# Contagem de outliers detectados
outliers_count = Counter(outliers)
print("Contagem de outliers:", outliers_count)
```

```
Contagem de outliers: Counter({1: 77946, -1: 14160})
```

```
# Remover outliers
data = data[data['outlier'] == 1]
data = data.drop(columns=['outlier'])


# Visualização de outliers após o tratamento com IsolationForest
plt.figure(figsize=(20, 15))
for i, col in enumerate(numeric_cols):
    plt.subplot(10, 7, i + 1)
    sns.boxplot(data[col])
    plt.title(col)
plt.tight_layout()
plt.show()
```



```
# Tratamento de dados faltantes (verificar novamente após remoção de outliers)
data[numeric_cols] = data[numeric_cols].apply(lambda x: x.fillna(x.mean()))
for col in categorical_cols:
    mode_value = data[col].mode()[0]
    data[col] = data[col].fillna(mode_value)


# Escalonamento dos dados
scalers = {
    'StandardScaler': StandardScaler(),
    'MinMaxScaler': MinMaxScaler(),
    'RobustScaler': RobustScaler(),
    'PowerTransformer': PowerTransformer(method='yeo-johnson')
}


# Aplicar cada escalonador e mostrar as primeiras linhas dos dados escalonados
scaled_data = {}
for name, scaler in scalers.items():
    scaled_data[name] = pd.DataFrame(scaler.fit_transform(data[numeric_cols]), columns=numeric_cols)
    print(f"\nDados escalonados usando {name}:")
    print(scaled_data[name].head())
```



```
Dados escalonados usando StandardScaler:
     HS_CPF  TEMPOCPF  DISTCENTROCIDADE  DISTZONARISCO  QTDENDERECO  QTDEMAIL  \
0 -1.560742  0.130546          0.269933       0.786456     0.468285 -0.585527
1 -1.311484  0.136607          0.623817      -0.442989    -0.340636  1.188051
2  1.275838  0.137365         -0.211429      -0.460391     1.277205 -0.585527
3 -0.723358  0.141911          0.781435      -0.388036     0.468285 -0.585527
4 -1.491063  0.127515          0.314493      -0.469418    -0.340636 -0.585527

   QTDCELULAR  CELULARPROCON  QTDFONEFIXO  TELFIXOPROCON  ...  \
0    0.039402      -0.043022    -1.025101      -0.098766  ...
1   -0.829401      -0.043022     0.394228      -0.098766  ...
2    0.039402      -0.043022     1.103892      -0.098766  ...
3   -0.829401      -0.043022     0.394228      -0.098766  ...
4   -0.829401      -0.043022    -1.025101      -0.098766  ...

   QTDDECLARACAOISENTA  QTDDECLARACAO10  QTDDECLARACAOREST10  \
0             -0.50597        -0.493300            -0.332852
1             -0.50597        -0.493300            -0.332852
2             -0.50597        -0.493300            -0.332852
3             -0.50597         0.556665            -0.332852
4             -0.50597        -0.493300            -0.332852

   QTDDECLARACAOPAGAR10  RESTITUICAOAGENCIAALTARENDA  BOLSAFAMILIA  \
0             -0.199191                    -0.047707     -0.210059
1             -0.199191                    -0.047707      4.760570
2             -0.199191                    -0.047707     -0.210059
3              1.016927                    -0.047707     -0.210059
4             -0.199191                    -0.047707     -0.210059

   ANOSULTIMARESTITUICAO  ANOSULTIMADECLARACAO  ANOSULTIMADECLARACAOPAGAR  \
0              -0.422957             -0.715308                  -0.290241
1              -0.422957             -0.715308                  -0.290241
2              -0.422957              1.400214                   3.449116
3              -0.422957              1.398312                   3.446875
4              -0.422957             -0.715308                  -0.290241

     TARGET
0  3.215214
1 -0.311021
2 -0.311021
3 -0.311021
4 -0.311021

[5 rows x 68 columns]

Dados escalonados usando MinMaxScaler:
     HS_CPF  TEMPOCPF  DISTCENTROCIDADE  DISTZONARISCO  QTDENDERECO  QTDEMAIL  \
0  0.004485  0.998404          0.223344       0.207385     0.041667     0.000
1  0.077888  0.999202          0.264439       0.017387     0.020833     0.125
```