UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

CENTRO DE TECNOLOGIA

ESCOLA POLITÉCNICA

EEL891 - Introdução ao Aprendizado de Maquina

Trabalho 01

2024.2

Abraão Carvalho Gomes - 121066101

1. Introdução

Este relatório descreve o processo de modelagem preditiva realizado com o objetivo de prever a inadimplência de solicitantes. O trabalho incluiu a importação e pré-processamento de dados, seleção de atributos, balanceamento de classes, treinamento de modelo com ajuste de hiperparâmetros e avaliação dos resultados.

---

2. Pré-Processamento dos Dados

2.1. Importação de Dados

Os conjuntos de dados de treinamento e teste foram importados a partir de arquivos CSV.

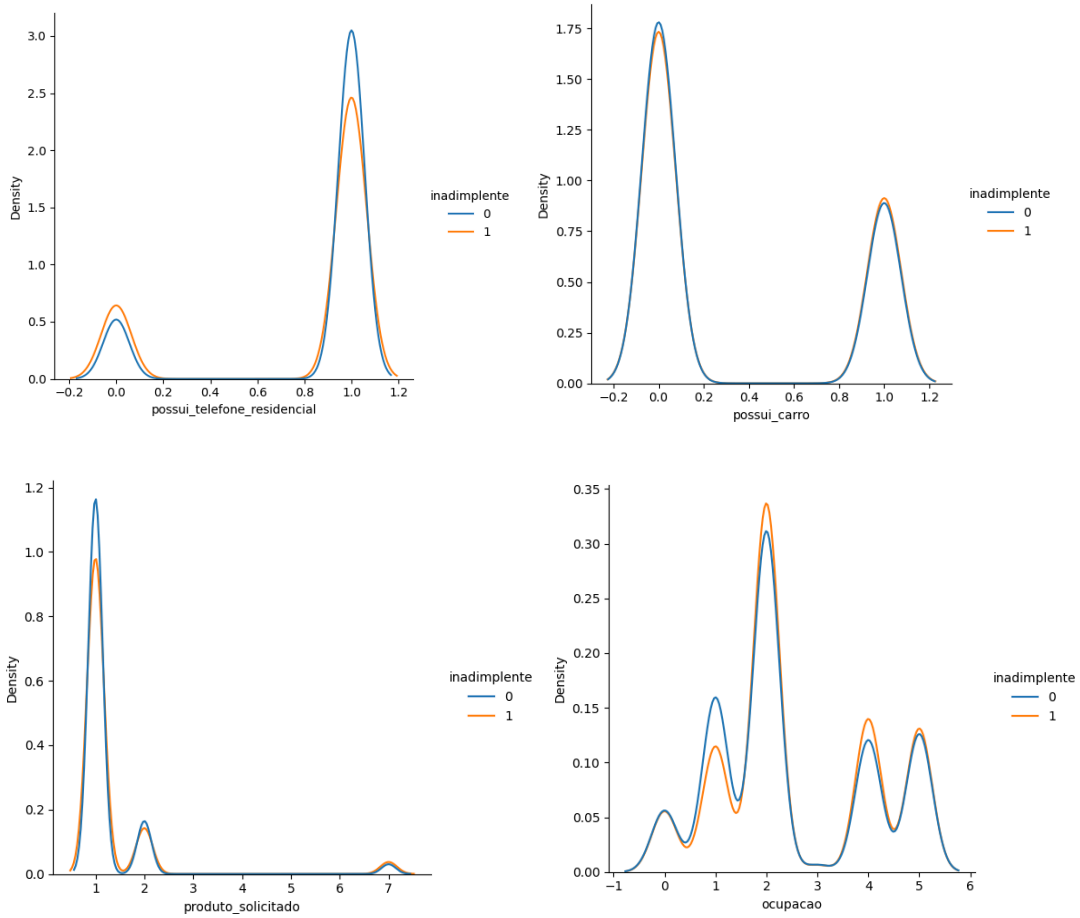
2.2. Seleção e Codificação de Atributos

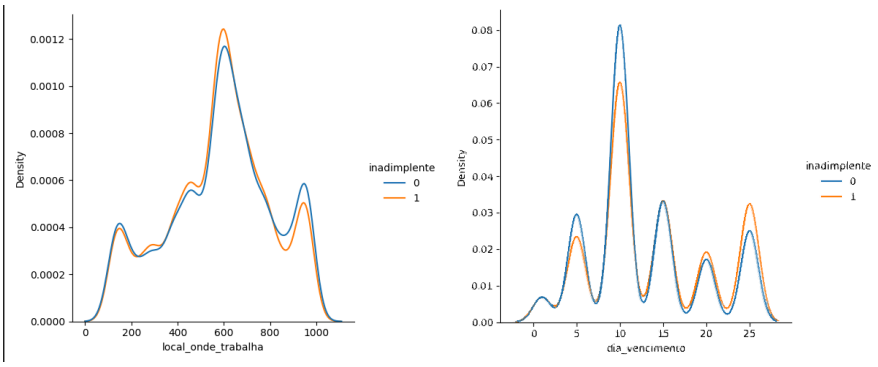
- A coluna `id_solicitante` foi excluída por não contribuir para a previsão do alvo.
- Colunas categóricas foram identificadas e codificadas utilizando `LabelEncoder`.

Foi feita uma análise de gráficos de densidade de inadimplentes para cada atributo na tentativa de identificar atributos com maior agrupamento de adimplentes e inadimplentes. Apesar de haver alguns atributos com uma distribuição levemente tendenciosa, nenhum deles era realmente desigual(facilitando a classificação). Assim, a filtragem com esses atributos piorou o score e o modelo final desconsidera essa seleção.

Além disso, esses gráficos revelaram uma característica importante: a distribuição de adimplentes e inadimplentes em cada atributo é bem iguaitária para cada faixa de valor. Desde que isso foi observado, soube-se que scores grandes seriam inalvejáveis.

Exemplo de gráficos:





2.3. Tratamento de Valores Ausentes

- Colunas com alta porcentagem de valores ausentes (`profissao_companheiro` e `grau_instrucao_companheiro`) foram excluídas. (Mais de 12 mil linhas com NaN)
- Valores nulos restantes nos dados de teste foram preenchidos utilizando a mediana dos valores das respectivas colunas.

2.4. Discretização de Variáveis Contínuas

Variáveis do tipo `float` foram convertidas para `int` para simplificar a análise e o treinamento do modelo.

2.5. Divisão e Normalização

- Os dados foram divididos em conjuntos de treino e validação utilizando `train_test_split` (80/20).
- A normalização foi realizada com o `StandardScaler` para padronizar os valores.

---

3. Treinamento do Modelo

3.1. Balanceamento de Classes

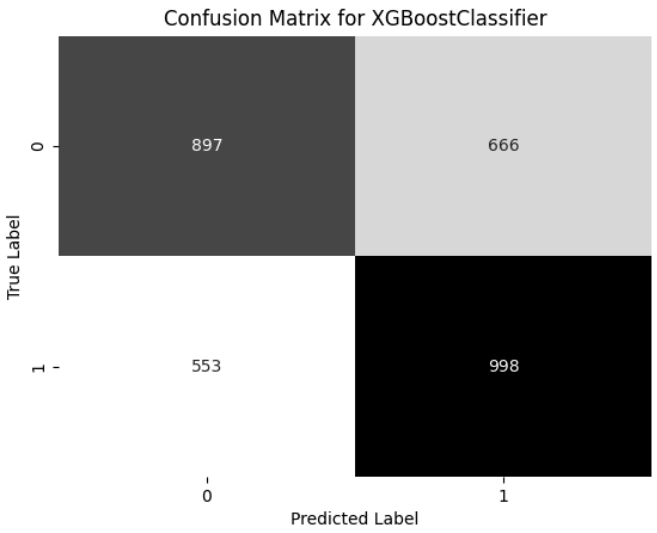
O conjunto de dados foi balanceado utilizando a técnica SMOTE para lidar com a desproporção de classes. Esse método de tratamento de dados mostrou resultado, aumentando a acurácia do modelo.

3.2. Modelo Selecionado

- O modelo `XGBClassifier` foi escolhido após testes o mesmo, além de RandomForestClassifier , GradientBoostingClassifier, AdaBoostClassifier, KNeighborsClassifier e GaussianNB.
- Hiperparâmetros ajustados com GridSearch**:**
    - **`learning_rate`: 0.01**
    - **`max_depth`: 4**
    - **`n_estimators`: 850**

3.3. Avaliação no Conjunto de Validação

- A matriz de confusão foi gerada para avaliar o desempenho do modelo.



- Relatório de classificação foi produzido para detalhar métricas como precisão, recall e F1-score.

```
Classification Report for XGBoostClassifier:
              precision    recall  f1-score   support

           0       0.62      0.57      0.60      1563
           1       0.60      0.64      0.62      1551

    accuracy                           0.61      3114
   macro avg       0.61      0.61      0.61      3114
weighted avg       0.61      0.61      0.61      3114

Cross-validation scores:  [0.57795591 0.58356713 0.60400802 0.59398798 0.59262229]
Mean cross-validation score:  0.5904282663161127
```

- Validação cruzada foi realizada com 5 folds, resultando em um score de 0.5904

---

4. Predição e Resultados Finais

4.1. Reajuste do Modelo

- O modelo foi re-treinado com todo o conjunto de dados balanceado.
- O conjunto de teste foi normalizado com os mesmos parâmetros do `StandardScaler` aplicado aos dados de treinamento.

4.2. Resultados Finais

- As predições foram realizadas no conjunto de teste.
- Os resultados foram salvos em um novo dataframe contendo as colunas `id_solicitante` e `inadimplente`.

5. Conclusão

- O dataSet fornecido para essa tarefa tem uma distribuição igualitária entre adimplentes e inadimplentes em todos os atributos. Isso dificulta a formação de grupos e consequentemente a classificação de uma pessoa em adimplente e inadimplente, mostrando-se ser um desafio com acertividade baixa (seria impossível almejar um score alto, mesmo com melhores técnicas).
- O uso de SMOTE e normalização mostraram-se eficazes para melhorar o resultado geral do modelo
- O modelo `XGBClassifier`, com hiperparâmetros ajustados, foi o melhor modelo dentre os testados.

Repositório dos trabalhos:

https://github.com/AbraaoCG/EEL891_ML_projects