

Primeira Competição de Inteligência Computacional

Lucas Duarte Sobreira¹

Tarek Campos Saleh²

1. lucasds123@discente.ufg.br
2. tarekcamposaleh@discente.ufg.br

Material elaborado em parceria com os
professores Nádia Felix Felipe da Silva e Fulano
da Silva
2022

INF

INSTITUTO DE
INFORMÁTICA



Sumário

1. Descrição do Problema
2. Descrição do Conjunto de Dados
3. Pré Processamento
4. Algoritmo
5. Modelo Final
6. Análise dos Resultados

Descrição do Problema

Descrição do Problema

O principal objetivo de uma Operadora de planos de saúde é controlar os gastos dos beneficiários, garantindo apenas o contratado. O fluxo atual, onde as prestadoras de serviço (Hospitais, clínicas, etc), realizam o pedido à operadora, essa valida o pedido e retorna foi ou não autorizado. O serviço será realizado apenas com a autorização da operadora. O processo de verificação de pedidos é realizado por auditores, que são caros de serem mantidos. A Operadora, para redução de custos, quer automatizar o sistema de validação dos pedidos, assim permitindo a redução de pessoal.

Descrição do Conjunto de Dados

Descrição do Conjunto de Dados

Tabela 1: Conjunto de dados de teste inicial e suas Features

Feature	Tipo	Quantidade	Faltando	Classes
Unnamed: 0	Numérico	227,122	0	227,122
NR_SEQ_REQUISICAO	Numérico	227,122	0	80,699
NR_SEQ_ITEM	Numérico	227,122	0	227,122
DT_REQUISICAO	Numérico	227,122	0	357
DS_TIPO_GUIA	Categórico	227,122	0	3
DT_NASCIMENTO	Numérico	227,112	10	16,557
NR_PRODUTO	Categórico	227,122	0	1
DS_TIPO_PREST_SOLICITANTE	Categórico	227,122	0	12
DS_CBO	Categórico	227,122	0	59
DS_TIPO_CONSULTA	Categórico	10,511	216,611	4
QT_TEMPO_DOENCA	Numérico	266	226,856	17
DS_UNIDADE_TEMPO_DOENCA	Categórico	266	226,856	3

DS_TIPO_DOENCA	Categórico	531	226,591	2
DS_INDICACAO_ACIDENTE	Categórico	209,539	17,583	4
DS_TIPO_SAIDA	Textual	0	227,122	0
DS_TIPO_INTERNACAO	Categórico	59,863	167,259	6
DS_REGIME_INTERNACAO	Categórico	59,863	167,259	3
DS_CARATER_ATENDIMENTO	Categórico	227,122	0	2
DS_TIPO_ACOMODACAO	Categórico	59,781	167,341	8
QT_DIA_SOLICITADO	Numérico	58,995	168,127	34
CD_GUIA_REFERENCIA	Numérico	37,463	189,659	4,610
DS_TIPO_ATENDIMENTO	Categórico	168,045	59,077	13
CD_CID	Textual	131,250	95,872	1,626
DS_INDICACAO_CLINICA	Textual	179,944	47,178	40,428
DS_TIPO_ITEM	Categórico	227,122	0	2
CD_ITEM	Numérico	227,122	0	6,220
DS_ITEM	Textual	227,122	0	6,146
DS_CLASSE	Textual	227,122	0	460
DS_SUBGRUPO	Categórico	227,122	0	72
DS_GRUPO	Categórico	227,122	0	9
QT_SOLICITADA	Categórico	227,122	0	270

Figura 2: Tabela de associações do SweetVIZ



Descrição do Conjunto de Dados

Features que destacam-se de maneira positiva

DS_INDICACAO_ACIDENTE, DS_GRUPO, DS_TIPO GUIA, DS_TIPO_ITEM

Alta amostra de dados com poucas classes

DT_NASCIMENTO

Alta amostra de dados

DS_ITEM, DS_CLASSE, DS_INDICACAO_CLINICA

Alta amostragem de dados e por definição desses itens, tendem a descrever o motivo da requisição

Descrição do Conjunto de Dados

Features que destacam-se de maneira negativa

NR_SEQ_REQUISICAO e NR_SEQ_ITEM

Quantidade de classes muito alta

DS_TIPO_SAIDA, QT_TEMPO_DOENCA e DS_UNIDADE_TEMPO_DOENCA

Amostra pequena de dados em relação ao total

NR_PRODUTO

Apenas uma classe

Pré Processamento

Limpeza de dados

```
def preproc(df):  
    df['DT_NASCIMENTO'] = df['DT_NASCIMENTO'].fillna(df['DT_NASCIMENTO'].median())  
    df['DS_TIPO_ACOMODACAO'] = df['DS_TIPO_ACOMODACAO'].fillna('NAO SE APLICA')  
    df['DS_TIPO_INTERNACAO'] = df['DS_TIPO_INTERNACAO'].fillna('NAO SE APLICA')  
    df['DS_TIPO_CONSULTA'] = df['DS_TIPO_CONSULTA'].fillna('NAO SE APLICA')  
    df['DS_UNIDADE_TEMPO_DOENCA'] = df['DS_UNIDADE_TEMPO_DOENCA'].fillna('dias')  
    df['QT_TEMPO_DOENCA'] = df['QT_TEMPO_DOENCA'].fillna(0)  
    df['QT_DIA_SOLICITADO'] = df['QT_DIA_SOLICITADO'].fillna(0)  
    df['DS_TIPO_ATENDIMENTO'] = df['DS_TIPO_ATENDIMENTO'].fillna('DESCONHECIDO')  
    df['DS_INDICACAO_ACIDENTE'] = df['DS_INDICACAO_ACIDENTE'].fillna('DESCONHECIDO')  
    df['CD_GUIA_REFERENCIA'] = df['CD_GUIA_REFERENCIA'].fillna(0)  
    return df
```

Pré Processamento

Transformações em *features* de texto

```
def textcolumns(df):  
    df['TAMANHO_DS_ITEM'] = df['DS_ITEM'].map(lambda s: len(str(s)))  
    df['TAMANHO_DS_CLASSE'] = df['DS_CLASSE'].map(lambda s: len(str(s)))  
    df['TAMANHO_DS_INDICACAO_CLINICA'] = df['DS_INDICACAO_CLINICA'].fillna('').map(lambda  
s: len(str(s)))  
  
    return df
```

Algoritmos

Algoritmos

GaussianNB foi testado, visto que este implementa um algoritmo de **Naive Bayes**. Devido à característica dos dados, vista pelo **SweetVIZ**, a maioria dos dados possuem baixa correlação com a feature *DS_STATUS_ITEM*, o que indica quase ou completa independência entre as features. Por o algoritmo de **Naive Bayes** realizar a suposição de independência entre as features, era de bom senso testar este algoritmo para a classificação

O algoritmo **RandomForestClassifier** também foi testado por ser um algoritmo de árvores de decisões que são comumente utilizados em problemas de classificação e por sua implementação melhorar a acurácia e prevenir o *over-fitting* com a utilização de uma floresta de árvores de decisões.

Modelo Final

Modelo Final

O modelo final foi obtido com o ***RandomForestClassifier*** e essas features:

Features

DS_INDICACAO_ACIDENTE, DS_TIPO_ITEM, DS_TIPO_INTERNACAO, DS_TIPO_GUIA, DS_TIPO_ACOMODACAO, DS_TIPO_ATENDIMENTO, DS_TIPO_CONSULTA, DS_TIPO_PREST_SOLICITANTE, DS_GRUPO, DS_REGIME_INTERNACAO, TAMANHO_DS_INDICACAO_CLINICA, TAMANHO_DS_ITEM, TAMANHO_DS_CLASSE, DS_CBO, QT_SOLICITADA, DT_REQUISICAO e DT_NASCIMENTO

Análise dos Resultados

Análise dos Resultados

Figura 3: Resultado no conjunto de testes

	precision	recall	f1-score	support
Autorizado	0.84	0.91	0.87	30832
Negado	0.77	0.63	0.69	14593
accuracy			0.82	45425
macro avg	0.81	0.77	0.78	45425
weighted avg	0.82	0.82	0.82	45425

- Modelo atingiu resultados decentes para a classificação desejada de um pedido autorizado, considerando casos teóricos.
- Precisão e Recall com bons scores para classificação de Autorizados.
- Muitos Falsos Negativos.
- Resultado no conjunto de validação: 0,68634 (Mean F1-Score)

Obrigado!

Dúvidas ou sugestões:

lucasds123@ufg.br ou
tarekcamposaleh@inf.ufg.br

