

1ª Competição de Inteligência Computacional

Predição de Cobertura de um Plano de Saúde

**Discente. Humberto Pereira
Teixeira Silva**

**Discente. Rhuan Webster
de Lourenco e Silva**

1. humbertot123@discente.ufg.br
2. rhuanwebster@discente.ufg.br

Material elaborado em parceria com os
discentes Humberto Pereira Teixeira Silva e
Rhuan Webster de Lourenco e Silva

2022

INF

INSTITUTO DE
INFORMÁTICA



Sumário

1. Descrição do problema e o dataset
2. Dataset e problema
3. Descrição de atividades
4. Análise de resultados

Descrição do problema e o dataset

O principal objetivo da operadora é controlar os gastos gerados pelos beneficiários, garantindo que eles realizem apenas os procedimentos contratualizados e que sejam pertinentes para a manutenção da saúde. Para isso, a operadora exige que todos os atos assistenciais sejam previamente autorizados.

Então o objetivo do problema é construir uma ferramenta capaz de combinar todas as variáveis de uma requisição, e baseada no comportamento histórico dos auditores, consiga “prever” o desfecho da solicitação (autorizado ou não autorizado).

O dataset é composto por dados preenchidos por um colaborador da prestadora (hospital, clínica, laboratório ou consultório) requisitando cobertura das despesas de produtos e serviços prestados ao cliente (beneficiário do plano), os aqui disponibilizados foram anonimizados e fornecidos por uma operadora de plano de saúde (dados reais).

Descrição de atividades

Ler os datasets disponibilizados no Kaggle:

```
treino = pd.read_csv('/kaggle/input/competicao-um-ic/train.csv')  
teste = pd.read_csv('/kaggle/input/competicao-um-ic/test.csv')
```

Separando os tipos de dados:

```
treinamento = preTratamento(treino)  
numerico = treinamento[['QT_TEMPO_DOENCA', 'QT_SOLICITADA', 'QT_DIA_SOLICITADO', 'CD_GUIA_REFERENCIA']]  
categorico = treinamento[['DS_UNIDADE_TEMPO_DOENCA', 'DS_TIPO_INTERNACAO', 'DS_CARATER_ATENDIMENTO', 'DS_TIPO_ATENDIMENTO',  
classe = treinamento['DS_STATUS_ITEM']
```

Colocar algum valor dentro dos campos do datasets que são nulos:

```
def preTratamento(df): #define valores para campos nulos

    df.DS_UNIDADE_TEMPO_DOENCA = df.DS_UNIDADE_TEMPO_DOENCA.fillna('0')
    df.DS_INDICACAO_ACIDENTE = df.DS_INDICACAO_ACIDENTE.fillna('0')
    df.DS_TIPO_ATENDIMENTO = df.DS_TIPO_ATENDIMENTO.fillna('0')
    df.DS_TIPO_INTERNACAO = df.DS_TIPO_INTERNACAO.fillna('0')
    df.DS_CARATER_ATENDIMENTO = df.DS_CARATER_ATENDIMENTO.fillna('0')
    df.DS_TIPO_PREST_SOLICITANTE = df.DS_TIPO_PREST_SOLICITANTE.fillna('0')
    df.DS_TIPO_GUIA = df.DS_TIPO_GUIA.fillna('0')
    df.DS_GRUPO = df.DS_GRUPO.fillna('0')
    df.DS_CBO = df.DS_CBO.fillna('0')
    df.DS_SUBGRUPO = df.DS_SUBGRUPO.fillna('0')

    df.QT_TEMPO_DOENCA = df.QT_TEMPO_DOENCA.fillna(0)
    df.QT_DIA_SOLICITADO = df.QT_DIA_SOLICITADO.fillna(0)
    df.CD_GUIA_REFERENCIA = df.CD_GUIA_REFERENCIA.fillna(0)
    df.QT_SOLICITADA = df.QT_SOLICITADA.fillna(0)
    df.CD_ITEM = df.CD_ITEM.fillna(0)
    return df
```

Insere um tipo de valor numérico aos dados que não são numéricos:

```
def standardScalerFunc(data):  
    ss = StandardScaler()  
    sstransformed = ss.fit_transform(data)  
  
    return ss, pd.DataFrame(sstransformed)  
  
def oneHotEncoderFunc(data):  
    x2 = OneHotEncoder(dtype='float', sparse=False, handle_unknown='ignore')  
    ohetransformed = x2.fit_transform(data)  
  
    return x2, pd.DataFrame(ohetransformed)|
```

Classificadores utilizados:

```
parameters = DecisionTreeClassifier(max_depth=None, criterion='gini')  
#parameters = svm.LinearSVC(penalty='l2', loss='squared_hinge', dual=True, tol=0.0001, C=1.0, multi_class='crammer_singer', fit_i  
#parameters = svm.SVC(kernel='poly', max_iter=1000)
```



Análise de resultados

Resultados obtidos usando o classificador SVM

```
Report:

```

	precision	recall	f1-score	support
Autorizado	0.43	0.00	0.00	30832
Negado	0.32	1.00	0.49	14593
accuracy			0.32	45425
macro avg	0.37	0.50	0.24	45425
weighted avg	0.39	0.32	0.16	45425

Matriz de confusão:

```
[[ 3 30829]
 [ 4 14589]]
```

Acurácia:

```
0.3212328013208586
```

Resultados obtidos o classificador Árvore de Decisão

Report:

	precision	recall	f1-score	support
Autorizado	0.73	0.92	0.82	30832
Negado	0.64	0.29	0.40	14593
accuracy			0.72	45425
macro avg	0.69	0.61	0.61	45425
weighted avg	0.70	0.72	0.68	45425

Matriz de confusão:

```
[[28493 2339]
 [10399 4194]]
```

Acurácia:

0.7195817281232801

Obrigado!

Dúvidas ou sugestões:

rhuanwebster@discente.ufg.br

