Universidade Federal de Goiás
Instituto de informática
Profa Nádia Félix Felipe da Silva

# Relatório: Predição de Cobertura de um Plano de Saúde

Aluno: Humberto Pereira Teixeira Silva e Rhuan Webster de Lourenco e Silva
**Disciplina:** Inteligência Computacional

Mês: Dezembro
Ano: 2022

Universidade Federal de Goiás
Instituto de Informática
Disciplina: Inteligência Computacional

# Relatório

Primeiro Relatório da participação dos Alunos Humberto Pereira Teixeira Silva e Rhuan Webster de Lourenco e Silva do Curso Engenharia de Computação da Universidade Federal de Goiás, como requisito parcial para Aprovação da Disciplina Inteligência Computacional.

Alunos: Humberto Pereira Teixeira Silva e Rhuan Webster de Lourenco e Silva

Professora: Nádia Félix Felipe da Silva

Mês: Dezembro
Ano: 2022

# Conteúdo

# 1 Resumo

O objetivo deste relatório é mostrar e explicar todas as atividades e recursos computacionais usados para solucionar o problema que nos foi proposto na 1ª Competição da disciplina de Inteligência Computacional.

O problema em questão é a "Predição de Cobertura de um Plano de Saúde", acredito que todos nós já tivemos que esperar algum procedimento médico ser liberado. Uma prestadora de serviço de saúde visando otimizar o atendimento e calcular em média quanto um paciente usou em determinado procedimento médico, resolveu colocar um funcionário para preencher um formulário com tudo que os pacientes utilizavam.

Basicamente com o banco de dados fornecido na competição, precisamos utilizar algumas bibliotecas (que serão melhor explicadas no decorrer do relatório) para a organização e tratamento desses dados e por fim utilizamos vários classificadores onde nosso objetivo era o melhor desempenho.

# 2 Descrição do Conjunto de dados

O conjunto de dados é composto por dados preenchidos por um funcionário da prestadora (hospital, clínica, laboratório ou atendente) solicitando a cobertura das despesas de produtos e serviços prestados ao cliente (beneficiário do plano), os dados que foram disponibilizados foram anonimizados e fornecidos pela operadora de plano de saúde (dados reais). No final dos dados são classificados se o serviço é autorizado ou não pela operadora. O conjunto de dados de treino estão na Figura 1

Figura 1: Conjunto de dados de treino

| | Unnamed: 0 | NR_SEQ_REQUISICAO | NR_SEQ_ITEM | DT_REQUISICAO | DS_TIPO_GUIA | DT_NASCIMENTO | NR_PRODUTO | DS_TIPO_PREST_SOLICITANTE |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1120527 | 2905946 | 2459367 | Guia de solicitac?o SP/SADT | 2439348.0 | 1 | PRESTADOR DE SERVICOS |
| 1 | 2 | 615210 | 1333736 | 2459091 | Guia de solicitac?o internac?o | 2443536.0 | 1 | HOSPITAL |
| 2 | 4 | 1135757 | 897098 | 2459375 | Guia de solicitac?o SP/SADT | 2439815.0 | 1 | CLINICA |
| 3 | 6 | 1088493 | 2800174 | 2459352 | Guia de solicitac?o SP/SADT | 2439139.0 | 1 | CLINICA DE IMAGEM |
| 4 | 7 | 936746 | 2311078 | 2459268 | Guia de solicitac?o SP/SADT | 2435595.0 | 1 | CLINICA |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 227117 | 413260 | 1069787 | 889198 | 2459342 | Guia de solicitac?o de prorrogac?o de internac?o | 2438592.0 | 1 | HOSPITAL |
| 227118 | 413261 | 921592 | 2265626 | 2459259 | Guia de solicitac?o SP/SADT | 2437804.0 | 1 | MEDICO |
| 227119 | 413262 | 608870 | 1313415 | 2459089 | Guia de solicitac?o SP/SADT | 2440041.0 | 1 | CLINICA |
| 227120 | 413264 | 690055 | 1570248 | 2459136 | Guia de solicitac?o SP/SADT | 2440704.0 | 1 | CLINICA |
| 227121 | 413265 | 1131307 | 2942784 | 2459374 | Guia de solicitac?o SP/SADT | 2441921.0 | 1 | CLINICA DE IMAGEM |

# 3 Descrição de atividades

Começamos analisando o conjunto de dados fornecidos na competição, que são os dados preenchidos por uma prestador de serviço de saúde requisitande despesas de saúde aos seus beneficiários. No primeiro momento, foi escolhido os atributos que seriam importantes no treinamento, já que não seria viável utilizar todos os atributos presentes no dataset devido o estouro de memória. Na análise dos dados destes atributos foi possível observar que existem campos vazios, então era preciso atribuir um valor númerico nesses campos. Após fazer o tratamento desses dados vazios, como mostrado na Figura 2, chegamos no problema de trarar os campos de textos no conjunto de dados. Para tratar os dados foi necessário utilizar bibliotecas do *Scikit Learn* como *OneHotEncoder*, *StandarScalerFunc* que fazem a conversão dos tipos de dados para dados numéricos. O mesmo foi feito nos dados de testes disponibilizados na plataforma da competição.

Figura 2: Usando o método fillna para atribuir valores aos campos vazios

```
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
```

Após o pré-tratamento dos dados, pulamos para a parte de escolher o melhor classificador para treinar os dados. Tentamos três opções de classificadores: classificador SVC (kernel poly), classificador LinearSVC e a árvore de decisão. Após treinarmos os dados, foi gerado um arquivo .csv com os dados já classificados, que era submetido na plataforma Kaggle. O arquivo gerado foi uma tabela de duas colunas: ID e DS_STATUS_ITEM. As bibliotecas utilizadas durante a resolução do problema são motradas na Figura 3.

Figura 3: Bibliotecas utilizadas

```
from itertools import zip_longest
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.svm import SVC
from sklearn.metrics._plot.confusion_matrix import confusion_matrix
from sklearn.preprocessing import StandardScaler,OneHotEncoder
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
```

# 4 Análise dos Resultados

Como foi utilizado três tipos de classificadores, os resultados foram diferentes. O primeiro classificador utilizado foi o classificador SVC. Modificamos os parâmetros para que seja realizado no máximo 1000 iterações com o kernel *poly*. A acurácia do resultado usando este classificador foi de 32.12% conforme mostrado na Figura 4. Além de que a precisão para acertar as classes autorizado e negado foram de 43% e 32% respectivamente.

Figura 4: Resultados usando o classificador SVC

```
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
```

Como o objetivo era conseguir o melhor resultado possível, escolhemos outro classifador para treinamento. O próximo classficador é semelhante ao anterior quando se utiliza o kernel *linear*, mas ele escala melhor quando se lida com um número maior de amostras, que o caso da competição. No entanto, o seu tempo de treinamento era na casa de horas o que o torna pouco eficiente, apesar de que se olharmos sua precisão e acurácia são bem melhores que o classificador anterior, conforme mostrado na Figura 5.

Figura 5: Resultados usando o classificador LinearSVC

```
Report:
              precision    recall  f1-score   support

  Autorizado       0.70      0.96      0.81     30832
      Negado       0.62      0.14      0.22     14593

    accuracy                           0.70     45425
   macro avg       0.66      0.55      0.52     45425
weighted avg       0.67      0.70      0.62     45425

Matriz de confusão:
 [[29599  1233]
 [12612  1981]]
Acurácia:
 0.6952118877270226
```

No terceiro classifador, a árvore de decisão, conseguimos o melhor resultado, tanto em acurácia e agilidade no treinamento. Na árvore de decisão usamos o critério de Impureza Gini e optamos por não definir um valor

máximo para a profundidade da árvore. Usando esse classificador obtemos uma acurácia de 71.95% conforme mostrado na Figura 6.

Figura 6: Resultados usando 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:
 [[28502  2330]
 [10401  4192]]
Acurácia:
 0.7197358282883874
```

# 5 Trabalhos Futuros

Essa primeira competição foi determinante para a primeira parte da nota da disciplina de Inteligência Computacional, sendo assim teremos a segunda competição onde irá compor a segunda parte da nota.

# Bibliografia

- https://scikit-learn.org/stable/ (Acessado em 19/12/2022)
- Russell, S., Norvig, P. Inteligência Artificial, Editora Campus, 2004