



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

**PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA (PIBIC)**

**RELATÓRIO FINAL DO ESTUDANTE
PIBIC – VIGÊNCIA 2023/2024**

RELATÓRIO FINAL

UTILIZANDO TÉCNICAS DE BALANCEAMENTO DE DADOS NA CLASSIFICAÇÃO DE RÓTULOS MUSICAIS

CURITIBA

2024

LUCAS SOTOMAIOR ALVES PEREIRA
CARLOS NASCIMENTO SILLA JUNIOR
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO - PPGIa
CNPq

UTILIZANDO TÉCNICAS DE BALANCEAMENTO DE
DADOS NA CLASSIFICAÇÃO DE RÓTULOS
MUSICAIS

Relatório Final apresentado ao Programa Institucional de Bolsas de Iniciação Científica (PIBIC), Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação da Pontifícia Universidade Católica do Paraná.

Orientador: Prof. Dr. Carlos Nascimento Silla Junior.

CURITIBA
2024

RESUMO

Este artigo investiga o impacto de diferentes técnicas de balanceamento de dados em um conjunto de dados de músicas tradicionais irlandesas desequilibrado, com o objetivo de mitigar o viés em modelos de aprendizado de máquina e melhorar a previsão, especialmente de classes minoritárias. O conjunto de dados consiste em 10.200 músicas, categorizadas em 11 gêneros, cada uma representada por 1.022 características musicais de alto nível. A técnica de Smote foi aplicada para equilibrar a base de dados, variando o número de vizinhos mais próximos (k) em 3 e 5, além de ser combinada com TomekLink, utilizando diferentes estratégias de *undersampling*. Diferentes algoritmos de classificação, incluindo *Decision Tree*, *Random Forest*, *Naive Bayes*, *Multilayer Perceptron* e *SVM*, foram combinados com as estratégias de balanceamento e comparados usando validação cruzada em 10 partições. A avaliação do desempenho abrangeu métricas como acurácia, precisão, recall e F-measure, além da análise de matrizes de confusão para cada modelo. Na análise buscou-se descobrir qual dos métodos de balanceamento e classificação é o mais preciso e acurado para prever cada classe, principalmente as classes minoritárias.

LISTA DE FIGURAS

<i>Figura 1 - Separação da base de dados em dois dataframes.....</i>	<i>4</i>
<i>Figura 2 – Exemplo de criação da instancia do classificador Random Forest utilizando Smote TomekLink com a estratégia all.</i>	<i>4</i>
<i>Figura 3 – Criação da variável responsável pela validação cruzada.....</i>	<i>4</i>
<i>Figura 4 – Particionamento dos dataframes partição de teste e treino, além da execução do classificador.</i>	<i>5</i>

LISTA DE TABELAS

<i>Tabela 1 - Detalhes da Base de Dados.....</i>	<i>2</i>
<i>Tabela 2 - Configurações dos métodos de balanceamento de dados.....</i>	<i>6</i>
<i>Tabela 3 - Métricas de cada classificador</i>	<i>7</i>
<i>Tabela 4 – Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Decision Tree.</i>	<i>8</i>
<i>Tabela 5 - Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Multilayer Perceptron.....</i>	<i>9</i>
<i>Tabela 6 – Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Naive Bayes.</i>	<i>9</i>
<i>Tabela 7 - Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Random Forest.....</i>	<i>10</i>
<i>Tabela 8 – Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador SVM.....</i>	<i>11</i>

LISTA DE ABREVIATURAS OU SIGLAS

M	- Modelo do Classificador
C	- Número da configuração da estratégia de resampling
Smote	- Synthetic Minority Oversampling Technique
DT	- Decision Tree
RF	- Random Forest
NB	- Naive Bayes
MLP	- Multilayer Perceptron
SVM	- Support Vector Machine
B	- Classe Barndance
H	- Classe Hornpipe
J	- Classe Jig
M	- Classe Mazurka
P	- Classe Polka
R	- Classe Reel
S	- Classe Slide
SJ	- Classe Slip_Jig
ST	- Classe Strathspey
TT	- Classe Three_Two
W	- Classe Waltz

SUMÁRIO

1 INTRODUÇÃO	1
2 OBJETIVOS.....	2
2.1 OBJETIVO GERAL	2
3 MATERIAIS E MÉTODO.....	2
4 RESULTADOS.....	6
5 DISCUSSÃO	11
6 CONCLUSÃO	12
6.1 RECOMENDAÇÕES PARA TRABALHOS FUTUROS (OPCIONAL)	ERRO!
INDICADOR NÃO DEFINIDO.	
7 USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA	13
REFERÊNCIAS.....	14

1 INTRODUÇÃO

A classificação musical é uma área de pesquisa atual e relevante no contexto da análise de dados e da inteligência artificial. Identificar com precisão o gênero musical de uma determinada faixa é crucial para uma variedade de aplicações e serviços, tais como recomendações musicais, organização de bibliotecas digitais e análise de tendências do mercado musical. O desafio, contudo, reside na diversidade e complexidade dos gêneros musicais, que são muitas vezes subjetivos e multifacetados. Em tal caso, o projeto explora o impacto de técnicas de balanceamento de dados na classificação de músicas tradicionais irlandesas. Foram aplicados algoritmos de aprendizado de máquina juntamente com o método de balanceamento de dados como Smote e TomekLink, para isso foi desenvolvido um programa em Python utilizando diversos frameworks para realizar a manipulação dos dados e classificação deles. Além disso foram realizadas análises para verificar qual é o classificador e a estratégia de balanceamento mais eficaz, foram analisadas matrizes de confusão e métricas estatísticas para que fosse possível ter uma tanto uma visão geral quanto específica dos resultados.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Verificar o impacto do uso de diferentes técnicas de balanceamento de dados para o problema de classificação de rótulos musicais.

3 MATERIAIS E MÉTODO

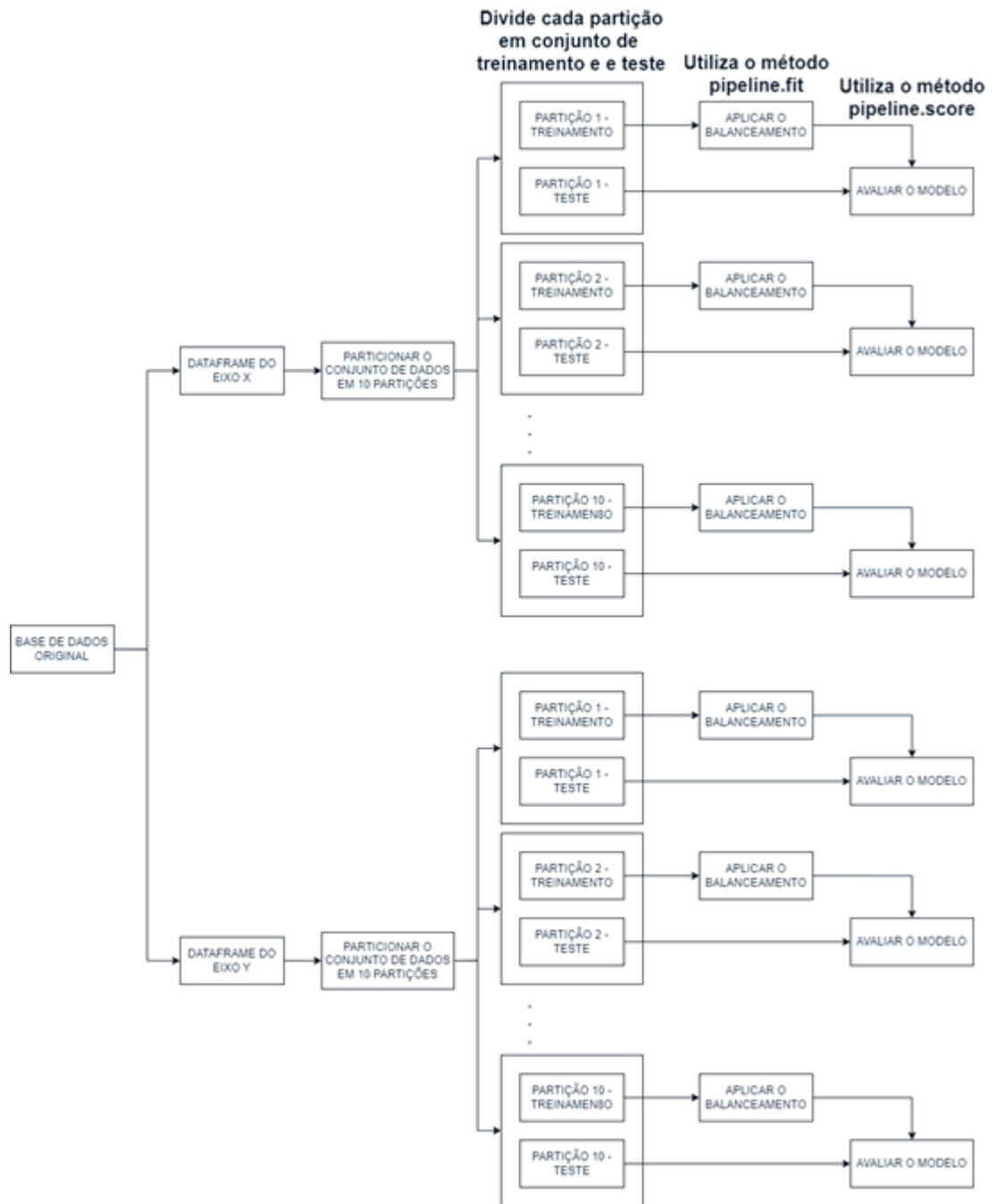
Inicialmente foram estudados diversos artigos sobre o assunto com o intuito de aprofundar o conhecimento sobre o funcionamento dos classificadores e sobre a base de dados que seria utilizada.

A base de dados utilizada no projeto foi desenvolvida por Martins & Silla Jr. (2015) e contém 10.200 músicas tradicionais irlandesas com 1,022 características pré-extraídas que são referentes a informações musicais de alto nível das músicas, como estatísticas sobre a frequência de notas e melodias. As músicas são classificadas em 11 diferentes gêneros musicais tradicionais irlandeses: *Barndance*, *Hornpipe*, *Jig*, *Mazurka*, *Polka*, *Reel*, *Slide*, *Slip Jig*, *Strathspey*, *Three-Two* e *Waltz*. A quantidade de músicas por gênero musical irlandês é apresentada no Quadro 1.

Tabela 1 - Detalhes da Base de Dados.

Gênero	Número de Músicas
Barndance	298
Hornpipe	843
Jig	2.666
Mazurka	116
Polka	965
Reel	3864
Slide	228
Slip Jig	380
Strathspey	329
Three-Two	78
Waltz	703

Figura 1 - Diagrama do particionamento da base de dados



O diagrama da Figura 1 demonstra o método de particionamento utilizado para aplicar o balanceamento de dados. A base de dados foi separada em dois dataframes, um para o eixo x e o outro para o eixo y, então foram criadas 10 partições de cada dataframe. Essas partições foram separadas em conjuntos de teste e treinamento. Para avaliar o modelo, as partições do eixo x e y são combinadas como podemos ver na Figura 5.

Durante o decorrer do projeto foi desenvolvido um algoritmo em Python que extrai as informações da base de dados e as armazena dois *dataframes* da biblioteca *pandas*, como mostra a Figura 1, em seguida cria a instancia do classificador no programa utilizando a biblioteca *sklearn*, como mostra a Figura 2, após isso realiza a validação cruzada utilizando o método *RepeatedStratifiedKFold* do pacote *model_selection* da biblioteca *sklearn* novamente, como é mostrado na Figura 3, por fim o programa particiona os *dataframes* em 10 vezes aplicando a validação cruzada criada anteriormente. Em cada uma das partições havia uma sessão de treinamento e uma de teste, onde a sessão de treinamento é comparada com a sessão de teste para realizar uma validação, como é mostrado na Figura 4.

Figura 2 - Separação da base de dados em dois *dataframes*.

```
# lê os dados do arquivo irish.csv
df = pd.read_csv('/content/irish.csv')

# divide x e y
x = df.drop('class', axis=1)
y = df['class']
```

Figura 3 – Exemplo de criação da instancia do classificador Random Forest utilizando Smote TomekLink com a estratégia all.

```
pipeline = Pipeline([
    ('sampling', SMOTETomek(sampling_strategy='all', random_state=15)),
    ('gnb', RandomForestClassifier())
])
```

Figura 4 – Criação da variável responsável pela validação cruzada.

```
rskf = RepeatedStratifiedKFold(n_splits=10, n_repeats=1, random_state=15)
```

Figura 5 – Particionamento dos dataframes partição de teste e treino, além da execução do classificador.

```
for train_index, test_index in rskf.split(X, y):
    # dividi os dados em treino e teste
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

    # usa o pipe do imblearn para aplicar o smote apenas nos subconjuntos de treinamento
    pipeline.fit(X_train, y_train)

    # usa os subconjuntos de teste para testar e obter as metricas
    score = pipeline.score(X_test, y_test)

    print("Session accuracy:", score)

    # matrizes de confusao
    y_pred = pipeline.predict(X_test)
    cm = confusion_matrix(y_test, y_pred)

    precision = precision_score(y_test, y_pred, average='weighted')
    recall = recall_score(y_test, y_pred, average='weighted')
    f1 = f1_score(y_test, y_pred, average='weighted')

    print("Precision: ", precision)
    print("Recall: ", recall)
    print("F-measure: ", f1)

    print(cm, "\n")
```

Em relação as estratégias de balanceamentos, durante a criação das instâncias dos classificadores, foram definidas parâmetros que por sua vez definem as estratégias de balanceamento como Smote e Smote com TomekLink. Os testes com Smote foram realizados variando o número de vizinhos mais próximos (k) em 3 e 5, já os testes com TomekLink fora realizado usando 4 diferentes estratégias de *undersampling*: *all* (redimensiona ambas as classes), *minority* (redimensiona apenas a classe minoritária), *not minority* (redimensiona as classes que não seja a minoritária) e *not majority* (redimensiona as classes que não sejam a majoritária). Além disso, foram realizados testes sem nem uma forma de balanceamento para que pudéssemos ter uma forma de comparação.

Cada um destes testes foi feito para cada um dos classificadores utilizados: *Decision Tree (DT)*, *Random Forest (RF)*, *Naive Bayes (NB)*, *Multilayer Perceptron (MLP)* e *SVM*.

Estes testes geraram uma matriz de confusão para cada uma das 10 partições que foram a partir da base de dados inicial. Além disso de foram calculados a Acurácia, a Precisão, o Recall e o F-measure de cada partição, sendo assim possível calcular a média de cada uma das métricas para classificador, assim como a média de acertos para classe individualmente.

4 RESULTADOS

Após todos os testes, foram calculados Acurácia, Precisão, Recall e F-measure médios para cada configuração de classificador testado juntamente com a estratégia de balanceamento, além da quantidade de acertos médios de cada classe individualmente.

Tabela 2 - Configurações dos métodos de balanceamento de dados.

Número da configuração (C)	Smote	K value	Tomek Link	Estratégia
1	Não	Não	Não	Não
2	Sim	3	Não	Não
3	Sim	5	Não	Não
4	Sim	5	Sim	all
5	Sim	5	Sim	minority
6	Sim	5	Sim	not majority
7	Sim	5	Sim	not minority

Tabela 3 - Métricas de cada classificador

M	C	Acurácia (Desvio Padrão)	Precisão (Desvio Padrão)	Recall (Desvio Padrão)	F-measure (Desvio Padrão)
DT	1	90.81% (0.0079)	91.28% (0.0097)	90.81% (0.0079)	91.00% (0.0083)
DT	2	89.62% (0.0111)	90.98% (0.0066)	89.62% (0.0111)	90.21% (0.0088)
DT	3	89.71% (0.0109)	91.31% (0.0057)	89.71% (0.0109)	90.41% (0.0079)
DT	4	89.48% (0.0088)	91.32% (0.0054)	89.48% (0.0088)	90.27% (0.0070)
DT	5	91.94% (0.0064)	91.71% (0.0049)	91.94% (0.0064)	91.78% (0.0054)
DT	6	89.53% (0.0084)	91.39% (0.0065)	89.53% (0.0084)	90.32% (0.0073)
DT	7	88.98% (0.0108)	90.95% (0.0062)	88.98% (0.0108)	89.81% (0.0086)
MLP	1	93.09% (0.0063)	92.40% (0.0058)	93.09% (0.0063)	92.30% (0.0057)
MLP	2	90.18% (0.0180)	92.72% (0.0073)	90.18% (0.0180)	91.12% (0.0125)
MLP	3	90.54% (0.0208)	92.63% (0.0085)	90.54% (0.0208)	91.28% (0.0129)
MLP	4	90.65% (0.0228)	92.63% (0.0048)	90.65% (0.0228)	91.41% (0.0149)
MLP	5	93.13% (0.0065)	91.97% (0.0108)	93.13% (0.0065)	92.08% (0.0073)
MLP	6	90.68% (0.0100)	92.68% (0.0038)	90.68% (0.0100)	91.44% (0.0074)
MLP	7	90.53% (0.0172)	92.91% (0.0030)	90.53% (0.0172)	91.41% (0.0109)
NB	1	89.0% (0.0064)	91.66% (0.0067)	89.0% (0.0064)	89.87% (0.0063)
NB	2	88.92% (0.0045)	92.28% (0.0038)	88.92% (0.0045)	90.19% (0.004)
NB	3	88.77% (0.0044)	92.15% (0.0039)	88.77% (0.0044)	90.07% (0.0039)
NB	4	88.78% (0.0052)	92.16% (0.0043)	88.78% (0.0052)	90.08% (0.0045)
NB	5	89.22% (0.0061)	91.55% (0.0055)	89.22% (0.0061)	90.00% (0.0056)
NB	6	88.78% (0.0052)	92.16% (0.0043)	88.78% (0.0052)	90.08% (0.0045)
NB	7	88.78% (0.0052)	92.16% (0.0043)	88.78% (0.0052)	90.08% (0.0045)
RF	1	92.08% (0.0073)	90.73% (0.0132)	92.08% (0.0073)	90.58% (0.0081)
RF	2	92.90% (0.0075)	92.49% (0.0074)	92.90% (0.0075)	92.59% (0.0074)
RF	3	92.51% (0.006)	92.23% (0.0062)	92.51% (0.006)	92.29% (0.0063)
RF	4	92.48% (0.0065)	92.36% (0.0063)	92.48% (0.0065)	92.32% (0.0065)
RF	5	91.86% (0.0078)	90.28% (0.0147)	91.86% (0.0078)	90.20% (0.0089)
RF	6	92.51% (0.0104)	92.30% (0.0096)	92.51% (0.0104)	92.32% (0.0101)
RF	7	92.48% (0.0086)	92.27% (0.0075)	92.48% (0.0086)	92.15% (0.0086)
SVM	1	59.14% (0.0105)	47.64% (0.0311)	59.14% (0.0105)	49.84% (0.0107)
SVM	2	39.05% (0.0259)	55.18% (0.025)	39.05% (0.0259)	40.78% (0.0293)
SVM	3	40.52% (0.0357)	56.93% (0.0279)	40.52% (0.0357)	42.24% (0.0396)
SVM	4	40.34% (0.0357)	57.05% (0.0279)	40.34% (0.0357)	42.04% (0.0396)
SVM	5	47.99% (0.0343)	47.31% (0.0278)	47.99% (0.0343)	43.57% (0.0387)
SVM	6	40.34% (0.0343)	57.05% (0.0278)	40.34% (0.0343)	42.04% (0.0387)
SVM	7	40.58% (0.0365)	57.06% (0.0283)	40.58% (0.0365)	42.41% (0.0407)

Uma análise geral sugere que a técnica de balanceamento de dados Smote com TomekLink usando a estratégia de minority obteve um resultado melhor que as outras técnicas. Já as técnicas de Smote com k=5, Smote TomekLink com estratégia all e not majority não obtiveram resultados melhores em nem uma métrica com nenhum classificador. Além disso foram obtidos resultado melhores que a baseline (configuração 1) na maioria dos classificadores.

Tabela 4 – Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Decision Tree.

M	C	B	H	J	M	P	R	S	SJ	ST	TT	W
DT	1	6.8 (2.7)	70.4 (3.777)	266.6 (0.516)	5.1 (1.792)	69.5 (0.527)	357.8 (4.59)	22.8 (0.422)	38 (0)	18.2 (1.933)	7.8 (0.422)	63.2 (3.645)
DT	2	7.8 (2.658)	69.1 (3.665)	266.6 (0.516)	4.1 (1.524)	69.5 (0.53)	347.8 (10.706)	22.8 (0.422)	38 (0)	17.6 (2.634)	7.8 (0.422)	62.9 (1.287)
DT	3	8.6 (3.098)	69 (4)	266.6 (0.516)	5.2 (1.932)	69.5 (0.53)	346.8 (8.270)	22.8 (0.422)	38 (0)	19 (2.357)	7.8 (0.422)	61.7 (3.164)
DT	4	10 (2.261)	68.1 (5.216)	266.6 (0.516)	5.1 (1.663)	69.5 (0.53)	344.4 (5.461)	22.8 (0.422)	38 (0)	19.1 (1.912)	7.8 (0.422)	61.2 (2.486)
DT	5	6.2 (1.229)	71.7 (4.449)	266.6 (0.516)	4.7 (1.252)	69.5 (0.53)	365.4 (3.806)	22.8 (0.422)	38 (0)	18.6 (2.119)	7.8 (0.422)	66.4 (1.776)
DT	6	9.5 (1.841)	68.4 (5.147)	266.6 (0.516)	5.4 (1.578)	69.5 (0.53)	343.6 (4.402)	22.8 (0.422)	38 (0)	19.3 (3.093)	7.8 (0.422)	62.2 (2.7)
DT	7	9.9 (2.47)	67.5 (5.759)	266.6 (0.516)	4.7 (2.263)	69.5 (0.53)	343.4 (7.763)	22.8 (0.422)	38 (0)	18.5 (3.136)	6.2 (1.229)	60.4 (2.459)

Referente ao classificador Decision Tree (DT), pode-se perceber que a técnica de Smote TomekLink utilizando a estratégia de *minority* obteve resultados gerais melhores que a baseline. Ao analisar as classes minoritárias (barndance, mazurka, slide e three_two) percebe-se que não houve melhora da média de acertos dessas classes individualmente em relação a baseline.

Ainda assim, quando comparado os resultados com e sem o Smote como um todo não se vê uma melhora significativa nos resultados em que foram implementados métodos de *resampling*, por exemplo quando utilizado Smote com k=5 a métrica de Precisão melhora de 91.28% para 91.31%, o que não é muito relevante, e quando utilizamos TomekLink a métrica de Precisão melhora para 91.71% usando a estratégia *minority*, como pode-se ver na Tabela 3. A taxa com que a métrica melhora é muito baixa para levar em consideração em uma análise geral.

Tabela 5 - Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Multilayer Perceptron.

M	C	B	H	J	M	P	R	S	SJ	ST	TT	W
MLP	1	4.2 (2.974)	75.0 (6.848)	266.6 (0.516)	4.9 (2.514)	69.3 (0.675)	374.5 (5.21)	22.7 (0.675)	37.9 (0.316)	20.7 (5.539)	6.7 (1.34)	66.9 (2.18)
MLP	2	12.9 (3.785)	75.2 (4.131)	266.3 (0.483)	6.7 (1.338)	69.2 (0.6325)	335.9 (20.08)	22.6 (0.8433)	37.8 (0.422)	21.0 (4.619)	7.5 (0.972)	64.6 (1.84)
MLP	3	12.3 (5.697)	72.4 (6.501)	266.4 (0.699)	6.3 (1.829)	69.2 (0.633)	342.7 (22.336)	22.8 (0.422)	37.9 (0.316)	21.3 (3.199)	7.5 (0.707)	64.6 (2.83)
MLP	4	12.2 (2.658)	73.4 (4.351)	266.5 (0.527)	7.0 (1.414)	69.1 (0.738)	343.3 (24.513)	22.5 (0.707)	37.9 (0.316)	22.2 (2.7)	7.2 (0.789)	63.2 (1.99)
MLP	5	3.7 (5.1)	76.8 (3.824)	266.5 (0.527)	5.1 (2.685)	68.9 (1.197)	378.2 (5.412)	22.7 (0.675)	37.7 (0.483)	16.9 (5.259)	7.3 (1.059)	66.0 (5.42)
MLP	6	13.5 (1.65)	72.4 (8.514)	266.4 (0.699)	5.9 (1.912)	69.3 (0.675)	344.2 (8.829)	22.6 (0.516)	37.9 (0.316)	20.9 (3.071)	7.1 (1.197)	64.6 (3.2)
MLP	7	13.7 (2.75)	72.3 (4.97)	266.5 (0.527)	6.7 (1.767)	69.2 (0.789)	341.7 (18.44)	22.7 (0.483)	37.9 (0.316)	22.2 (3.225)	6.7 (1.338)	63.7 (3.53)

A análise do classificador Multilayer Perceptron (MLP) apresenta resultados melhores quando utilizado Smote TomekLink com estratégia minority. Para as classes 'three_two' (que foi de 6,7 na baseline para 7,3), 'mazurka' (que foi de 4,9 para 5,1), hornpipe (que foi de 75 para 76,8). É relevante notar que o desvio padrão neste classificador específico para a classe minoritária 'three_two' diminuiu em relação a baseline. É relevante notar que as classes em geral obtiveram mais pioras do que melhoras.

Tabela 6 – Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Naive Bayes.

M	C	B	H	J	M	P	R	S	SJ	ST	TT	W
NB	1	12.4 (3.204)	75.2 (3.853)	266.6 (0.516)	9.7 (1.494)	69.5 (0.53)	345.2 (3.676)	22.8 (0.422)	38 (0)	12.4 (1.647)	7.8 (0.422)	48.1 (3.315)
NB	2	13.3 (3.199)	71.2 (4.492)	266.6 (0.516)	9.1 (1.287)	69.5 (0.53)	338.2 (3.259)	22.8 (0.422)	38 (0)	17.6 (2.319)	7.8 (0.422)	52.8 (3.293)
NB	3	13.2 (3.048)	71.1 (4.149)	266.6 (0.516)	8.8 (1.229)	69.5 (0.53)	336.8 (4.104)	22.8 (0.422)	38 (0)	17.5 (2.55)	7.8 (0.422)	53.3 (2.908)
NB	4	13.3 (3.302)	71.0 (4.690)	266.6 (0.516)	8.8 (1.22)	69.5 (0.527)	337.0 (4.163)	22.8 (0.422)	38 (0)	17.4 (2.547)	7.8 (0.422)	53.3 (2.908)
NB	5	11.8 (2.974)	75.2 (3.553)	266.6 (0.516)	9.3 (1.338)	69.5 (0.527)	347.1 (4.701)	22.8 (0.422)	38 (0)	11.8 (1.135)	7.8 (0.422)	50.1 (2.726)
NB	6	13.3 (3.302)	71.0 (4.690)	266.6 (0.516)	8.8 (1.229)	69.5 (0.53)	337.0 (4.163)	22.8 (0.422)	38 (0)	17.4 (2.547)	7.8 (0.422)	53.3 (2.907)
NB	7	13.3 (3.302)	71.0 (4.690)	266.6 (0.516)	8.8 (1.229)	69.5 (0.527)	337.0 (4.163)	22.8 (0.422)	38 (0)	17.4 (2.547)	7.8 (0.422)	53.3 (2.908)

Para o classificador Naive Bayes (NB) a métrica F-measure obteve melhora em todos os testes em relação a baseline. O Smote TomekLink usando a estratégia minority apenas obteve piora em relação a baseline na métrica precisão, a qual todos os outros experimentos com esse classificador obtiveram melhora em relação a baseline, como é apresentado na Tabela 3.

A análise detalhada do classificador Naive Bayes demonstra que muitas classes continuaram constantes a baseline. Já a classe minoritária ‘mazurka’ obteve piora, em relação a baseline em todos os testes, já a classe ‘barndance’ obteve piora apenas com Smote TomekLink usando a estratégia minority. As outras classes minoritárias continuaram contantes. Além disso as outras classes não minoritárias obtiveram mais pioras do que melhoras.

Tabela 7 - Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador Random Forest.

M	C	B	H	J	M	P	R	S	SJ	ST	TT	W
RF	1	1.3 (0.949)	80.6 (1.506)	265.9 (1.287)	3.6 (1.578)	66.4 (1.647)	382.0 (1.633)	17.6 (1.43)	32.0 (2.261)	19.2 (3.327)	4.8 (1.398)	65.7 (2.003)
RF	2	8.9 (2.767)	78.4 (1.838)	265.2 (1.476)	5.0 (2.404)	67.7 (1.703)	370.1 (3.213)	21.0 (0.667)	35.6 (2.221)	22.5 (2.369)	6.8 (1.033)	66.3 (1.16)
RF	3	9.4 (1.174)	78.2 (1.932)	265.4 (1.506)	5.0 (2.16)	67.2 (1.619)	367.9 (2.998)	20.8 (0.919)	34.8 (2.098)	22.5 (2.068)	6.7 (0.823)	65.6 (1.35)
RF	4	9.9 (1.595)	78.0 (1.7)	265.1 (1.663)	4.8 (2.201)	67.6 (1.35)	366.1 (4.28)	21.2 (0.789)	35.3 (2.163)	22.7 (3.268)	6.7 (0.949)	65.8 (1.135)
RF	5	0.5 (0.527)	80.2 (1.687)	266.0 (1.054)	3.6 (1.713)	65.1 (3.143)	383.3 (2.111)	16.9 (1.663)	29.7 (3.199)	18.7 (2.497)	7.6 (0.516)	65.3 (2.111)
RF	6	9.6 (1.955)	78.1 (2.079)	265.1 (1.524)	5.0 (2.261)	67.6 (1.838)	366.5 (4.428)	20.7 (1.16)	35.7 (1.337)	23.1 (2.558)	6.5 (1.08)	65.6 (1.955)
RF	7	9.9 (1.595)	78.6 (2.591)	265.1 (1.449)	5.1 (2.283)	67.4 (1.265)	368.6 (4.248)	20.7 (1.567)	35.9 (1.287)	23.1 (2.558)	2.9 (1.524)	65.9 (1.37)

Com relação ao classificador Random Forest (RF) a análise geral, de acordo com a Tabela 3, mostra que apenas o Smote TomekLink com a estratégia minority obteve métricas piores que a baseline, já todos os outros testes performaram melhor que baseline em todas as métricas. Em geral foi melhor classificador.

Em relação as classes minoritárias todas elas obtiveram mais melhoras do que pioras, ressaltando o fato de que na maioria das vezes o desvio padrão das classes ‘slide’ e ‘three_two’ diminuiu em relação a baseline quando a média de acertos também diminuiu.

Tabela 8 – Taxa de acerto de cada classe individualmente com desvio padrão utilizando o classificador SVM.

M	C	B	H	J	M	P	R	S	SJ	ST	TT	W
SVM	1	0.0 (0.0)	1.1 (1.197)	242.9 (2.923)	0.0 (0.0)	6.2 (1.398)	321.9 (7.838)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	31.1 (2.601)
SVM	2	6.3 (3.7)	93.8 (62.12)	112.1 (63.33)	11.1 (7.42)	79.3 (52.74)	220.8 (136.98)	29.8 (19.8)	27.7 (18.74)	17.5 (10.52)	5.6 (3.84)	56.7 (36.52)
SVM	3	5.0 (1.5)	55.8 (3.795)	80.5 (31.44)	6.5 (1.58)	48.0 (3.197)	134.3 (11.963)	17.9 (1.449)	15.7 (2.003)	10.2 (1.619)	3.4 (1.51)	36.0 (5.142)
SVM	4	4.9 (1.5)	56.0 (3.795)	80.4 (31.44)	6.5 (1.58)	47.9 (3.197)	132.4 (11.963)	17.9 (1.449)	15.7 (2.003)	10.2 (1.619)	3.5 (1.51)	36.0 (5.142)
SVM	5	0.0 (1.5)	0.0 (3.712)	243.1 (30.24)	0.0 (1.58)	0.9 (3.247)	232.1 (11.9)	0.0 (1.449)	0.0 (2.003)	0.0 (1.619)	5.8 (1.65)	7.5 (5.142)
SVM	6	4.9 (1.5)	56.0 (3.712)	80.4 (30.24)	6.5 (1.58)	47.9 (3.247)	132.4 (11.9)	17.9 (1.449)	15.7 (2.003)	10.2 (1.619)	3.5 (1.65)	36.0 (5.142)
SVM	7	6.2 (1.6)	56.2 (3.882)	81.7 (30.71)	6.4 (1.65)	47.8 (3.458)	135.7 (12.979)	17.9 (1.449)	15.7 (2.003)	10.3 (1.494)	0.0 (0.0)	36.0 (5.207)

Para o classificador SVM as métricas de acurácia, recall e F-measure não obtiveram melhora em nenhum experimento em relação a baseline. Já a métrica precisão obteve piora apenas no Smote TomekLink com a estratégia minority, de acordo com a Tabela 3.

Ao analisar as classes individualmente percebe-se uma melhora muito significativa em relação a baseline, mas isso não é muito relevante, pois a baseline não é um padrão de comparação muito significativo, ou seja, mesmo os outros experimentos sendo melhores que a baseline quando comparando as classes que cada um acertou, em uma visão geral ambos são imprecisos. Também é importante comentar que os valores de desvio padrão são muito altos e inconsistentes.

5 DISCUSSÃO

Apesar do classificador Random Forest ter se saído melhor que os outros em um aspecto geral, durante a análise destes experimentos ficou claro que não existe um classificador absoluto ou então uma estratégia de balanceamento que seja muito melhor que as outras. Cada um dos testes realizados performou melhor para uma classe ou métrica. O melhor classificador para a métrica de F-measure é o Random Forest com Smote k=3 que alcançou 92,59% e um desvio padrão de 0.0074 nesta métrica. Já para a classe minoritária 'three_two' a melhor taxa de acerto média individual foi de 7,8 com 0,422 de desvio padrão.

6 CONCLUSÃO

Concluimos que quando aplicadas as técnicas de balanceamento de dados Smote e TomekLink os classificadores performaram melhor que quando se utilizou da base de dados inalterada, principalmente com o classificador Random Forest que obteve resultados melhores que os outros classificadores com quase todas as técnicas de balanceamento. Entretanto, não foi uma melhora constante o suficiente para afirmar que ou o Smote ou o TomekLink extremamente superior ao outro, seria preciso mais pesquisas para determinar qual é a melhor estratégia para realizar o balanceamento de dados.

7 USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA

PERGUNTAS SOBRE O USO DE IA GENERATIVA
1) Para escrita deste relatório, alguma ferramenta de inteligência artificial generativa foi utilizada? Sim (<input type="checkbox"/>) Não (<input type="checkbox"/>)
2) Qual(is) ferramenta(s) de IA generativa você utilizou? Não se aplica (<input type="checkbox"/>) Se sim, cite quais:
3) Indique quais os usos de IA generativa foram aplicadas no neste relatório. Não se aplica (<input type="checkbox"/>) Correção gramatical (ortografia e concordância) (<input type="checkbox"/>) Formatação das referências (<input type="checkbox"/>) Gerar partes do texto escrito (ex: frases, parágrafos, conceitos) (<input type="checkbox"/>) Gerar a totalidade do texto escrito (<input type="checkbox"/>) Gerar citações (<input type="checkbox"/>) Criação/edição das imagens e gráficos (<input type="checkbox"/>) Correção/auxílio na formatação final dos códigos estatísticos (ou outro software) (<input type="checkbox"/>) Outros usos – especificar:

REFERÊNCIAS

BROWNLEE, J. **Random Oversampling and Undersampling for Imbalanced Classification**. Disponível em: <<https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/>>.

MARTINS, M. L. G.; SILLA JR., C. N. Irish Traditional Ethnomusicology Analysis Using Decision Trees and High Level Symbolic Features. **Proceedings of the 12th Sound and Music Computing Conference**, pp. 455-462, 2015.

SILLA JR., C. N.; FREITAS, A. A. A survey of hierarchical classification across different application domains. **Data Mining and Knowledge Discovery**, v. 22, n. 1-2, p. 31–72, 7 abr. 2010.