Richter's Predictor: Modeling Earthquake Damage

Alunos: João Pedro da Silva e Wyctor Fogos da Rocha

Professora: Mariana Rampinelli

Disciplina: Aprendizagem Máquinas

Turma: 2020/2

INSTITUTO FEDERAL
Espírito Santo

Visão Geral

- Com bases nos aspectos de localização e construção individual dos edifícios, prever o nível de dano causado pelo terremoto de Gorka em 2015;

- Os dados foram coletados através dos sobreviventes pelas empresas Kathmandu Living Labs e Central Bureau of Statistics ;

- Esta pesquisa contém informações sobre os impactos do terremoto, condições familiares e dados socioeconômicos demográficos.

Fonte: Wikimedia Commons

Problema

Estamos tentando prever a variável *damage_grade* que é categorizada em três níveis:

- Dano Baixo – 1;
- Dano Médio – 2;
- Destruição quase completa – 3.

Pashupatinath Temple - Nepal
Fonte: Wikimedia Commons

INSTITUTO FEDERAL
Espírito Santo

Métrica de desempenho

- Estamos prevendo um nível de dando de 1 até 3. Como é variável ordinal, a ordenação é importante.
- O desempenho do algoritmo é medido pela pontuação F1 que equilibra a precisão e o recall de um classificador.
- Para avaliar os 3 níveis é necessário utilizar a pontuação F1 micro média:

$$F_{micro} = \frac{2 \cdot P_{micro} \cdot R_{micro}}{P_{micro} + R_{micro}}$$

$$P_{micro} = \frac{\sum_{k=1}^{3} TP_k}{\sum_{k=1}^{3} (TP_k + FP_k)}, \quad R_{micro} = \frac{\sum_{k=1}^{3} TP_k}{\sum_{k=1}^{3} (TP_k + FN_k)}$$

INSTITUTO FEDERAL
Espírito Santo

Dataset

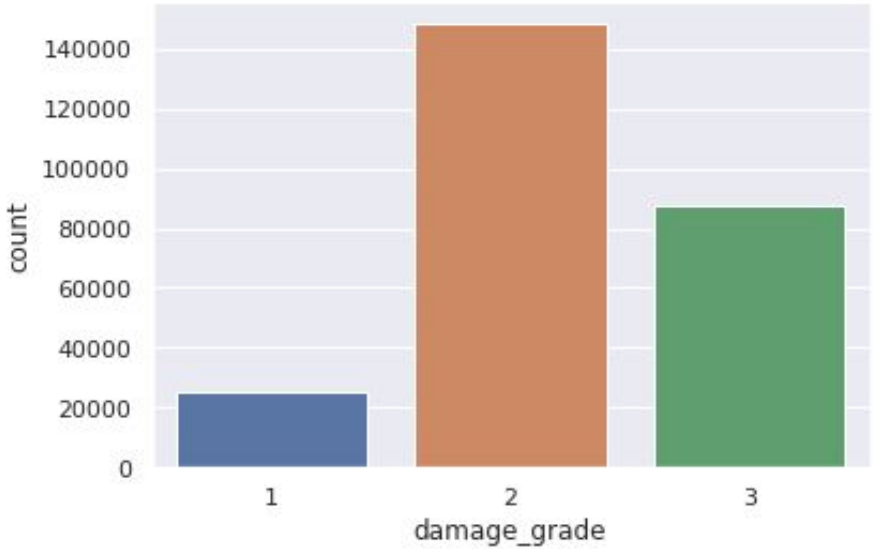
- O conjunto de dados consiste em informações estruturais dos edifícios e sua propriedade legal.
- O dataset de treino está dividido em 39 colunas, divididas em 3 tipos de datas:

Binário	Categórico	Inteiro
has_superstructure_adobe_mud	legal_ownership_status	building_id
has_superstructure_mud_mortar_stone	plan_configuration	geo_level_1_id, geo_level_2_id geo_level_3_id
has_superstructure_stone_flag	position	count_floors_pre_eq
has_superstructure_cement_mortar_stone	other_floor_type	age
as_superstructure_mud_mortar_brick	ground_floor_type	area_percentage
has_superstructure_cement_mortar_brick	roof_type	height_percentage
has_superstructure_timber	foundation_type	count_families
has_superstructure_bamboo	land_surface_condition	
has_superstructure_rc_non_engineered		
has_superstructure_rc_engineered		
has_superstructure_other		

Dataset

- O dataset se encontra inicialmente desbalanceado, como mostrado na figura abaixo:

- Em número absolutos:
  - 1- 25124
  - 2- 148259
  - 3- 25124

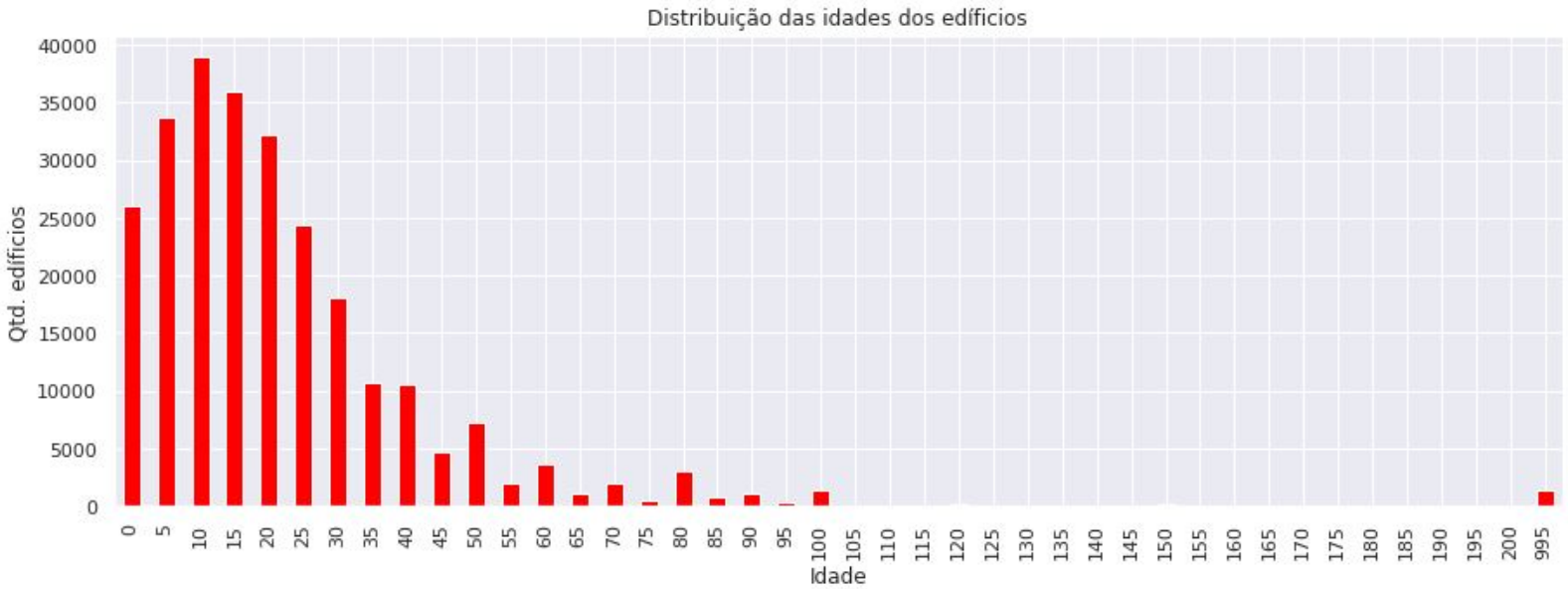


Fonte: código desenvolvido.

INSTITUTO FEDERAL
Espírito Santo

Dataset

- Extraindo informações sobre a idade de cada edifício:

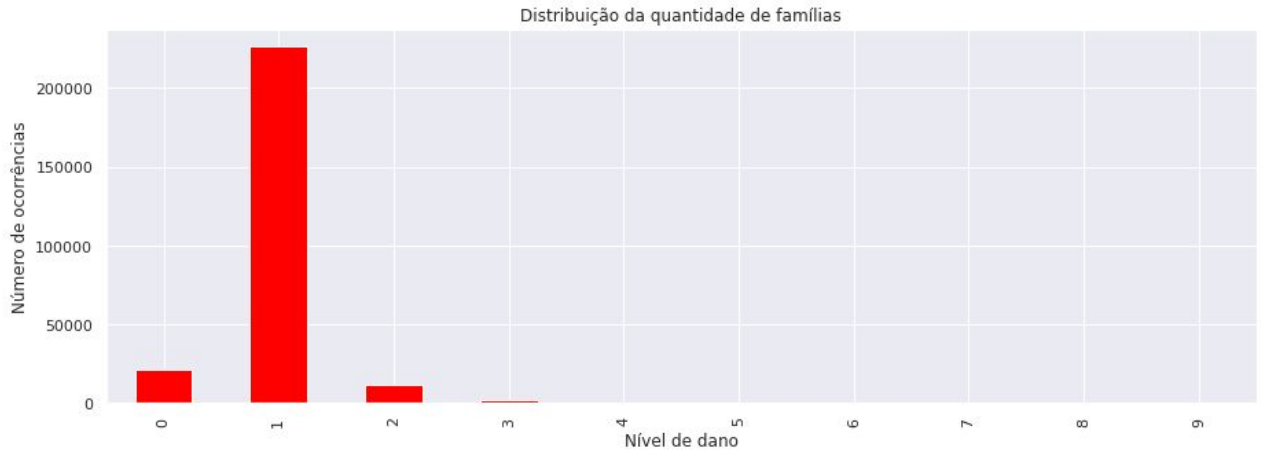


Fonte: código desenvolvido.

INSTITUTO FEDERAL
Espírito Santo

Dataset

- Quantidade de famílias afetadas por construção:



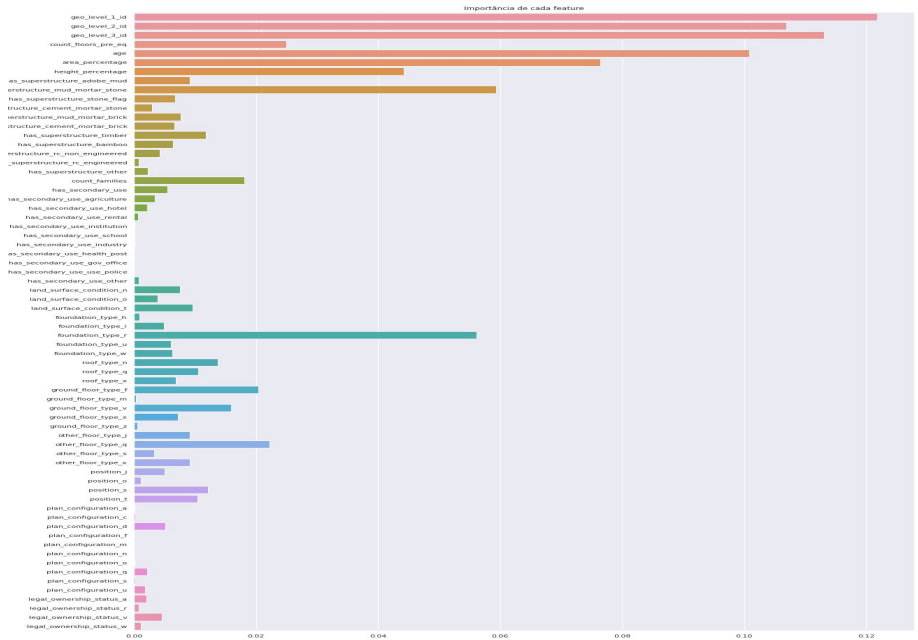
0	20862
1	226115
2	11294
3	1802
4	389
5	104
6	22
7	7
8	2
9	4

Fonte: código desenvolvido.

INSTITUTO FEDERAL
Espírito Santo

Classificador Random Forest (Floresta Aleatória)

Foi utilizado o recurso de *feature_importances* para obtenção dos valores da importância de cada feature.



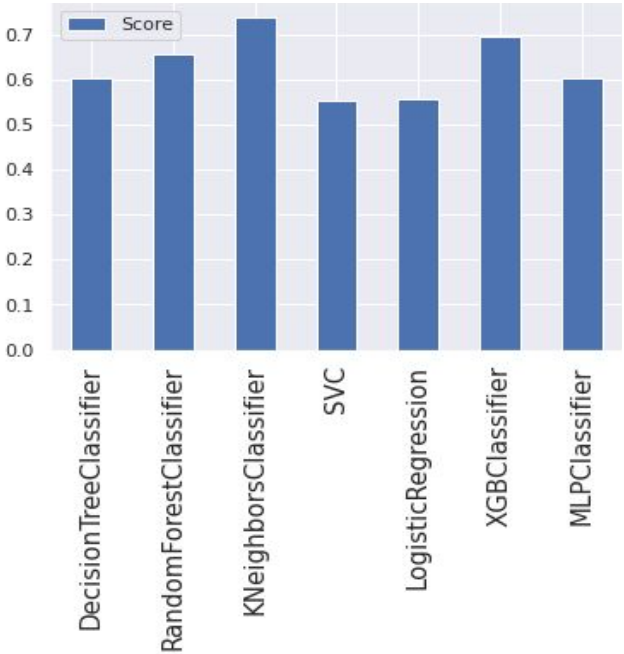
Fonte: código desenvolvido.

INSTITUTO FEDERAL
Espírito Santo

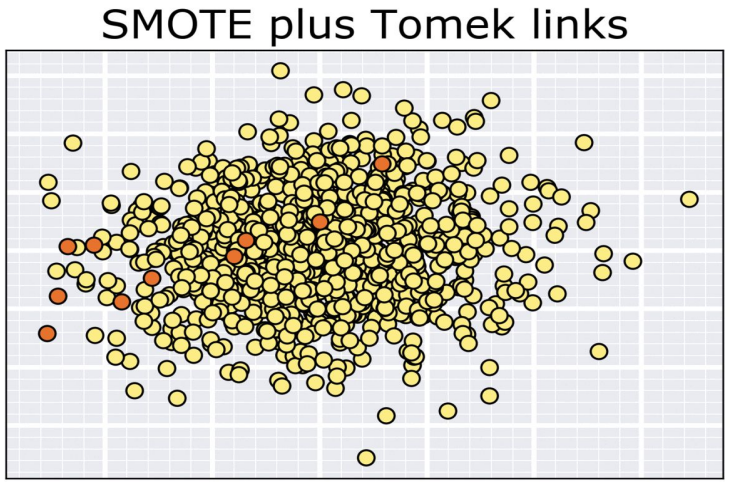
Seleção do classificador

Antes de tudo, o dataset original foi devidamente balanceado com a técnica de *oversampling*.

Foram utilizados alguns modelos para selecionar previamente qual o melhor a ser utilizado.



Fonte: código desenvolvido.



Fonte: TOWARDS DATA SCIENCE (2019).

INSTITUTO FEDERAL
Espírito Santo

Teste com os classificadores com melhores resultados

Com o melhor resultado prévio, foi analisado o *KNeighborsClassifier* para 10 valores de dados próximos (vizinhos), utilizando as features selecionadas pelo Classificador Random Forest.

Número de pontos vizinhos	Acurácia (%)	F1-Score (%)
1	77.0	77.0
2	75.0	75.0
3	77.0	77.0
4	77.0	76.0
5	76.0	76.0
6	76.0	76.0
7	76.0	75.0
8	76.0	75.0
9	75.0	75.0
10	75.0	74.0

INSTITUTO FEDERAL
Espírito Santo

Teste com os classificadores com melhores resultados

Com o melhor resultado encontrado utilizando o *KNeighborsClassifier* para 3 valores de dados próximos (vizinhos), utilizando as features selecionadas pelo Classificador Random Forest.

```
import time

clf=KNeighborsClassifier(3)
tempo_inicial=time.time()
clf.fit(X_train, Y_train)
ypred = clf.predict(X_val)
print(classification_report(ypred,Y_val))
print("Tempo de treino:{} s".format(time.time()-tempo_inicial
```

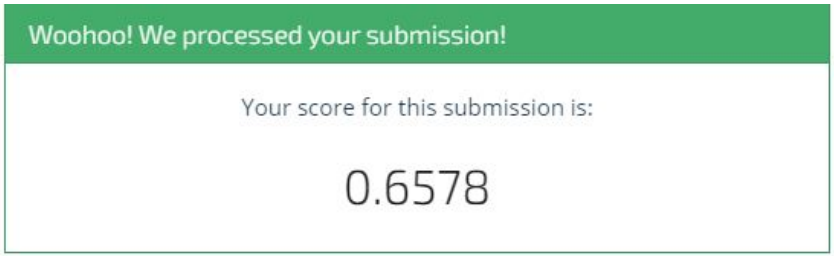
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:
  """
              precision    recall  f1-score   support

           1       0.94      0.85      0.89     48753
           2       0.64      0.74      0.68     38567
           3       0.78      0.76      0.77     46114

    accuracy                           0.79    133434
   macro avg       0.79      0.78      0.78    133434
weighted avg       0.80      0.79      0.79    133434

Tempo de treino:9.757726907730103 s
```

Fonte: código desenvolvido.



Fonte: Resultado da submissão.

INSTITUTO FEDERAL
Espírito Santo

Teste com os classificadores com melhores resultados

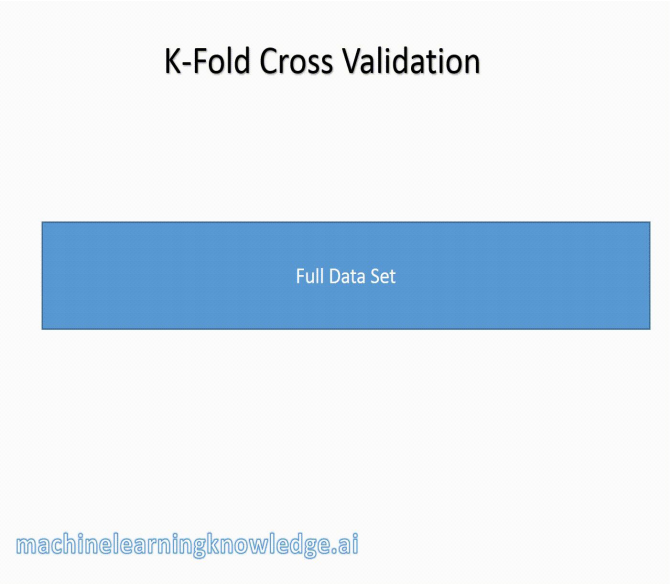
- Com o segundo melhor resultado prévio, foi analisado o *XGBClassifier* utilizando as melhores combinações encontradas com o uso do *RandomizedSearchCV* junto processo *K-fold*.

- No processo da validação cruzada busca estimar o quão preciso é o modelo na prática, ou seja, o seu desempenho para um conjunto de dados.

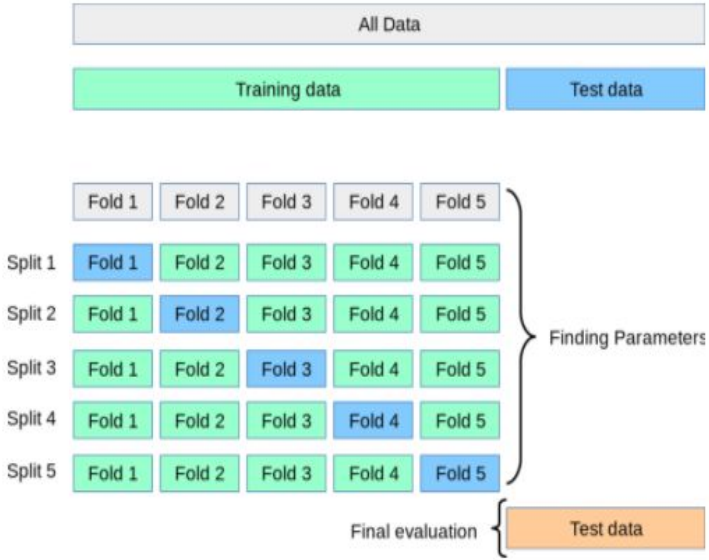
- No método K-fold, o conjunto de dados é dividido em k subconjuntos mutuamente exclusivos e com mesmo tamanho, com isso um subconjunto é utilizado para teste e os outros k-1 são utilizados para estimação dos parâmetros, realizando o cálculo da acurácia. O processo é repetido até ser realizado em todos k subconjuntos.

Teste com os classificadores com melhores resultados

- Processo ilustrativo do K-Fold.



Fonte: MLK (2018).



Fonte:scikit-learn.org/.

INSTITUTO FEDERAL
Espírito Santo

Algoritmo XGBoost

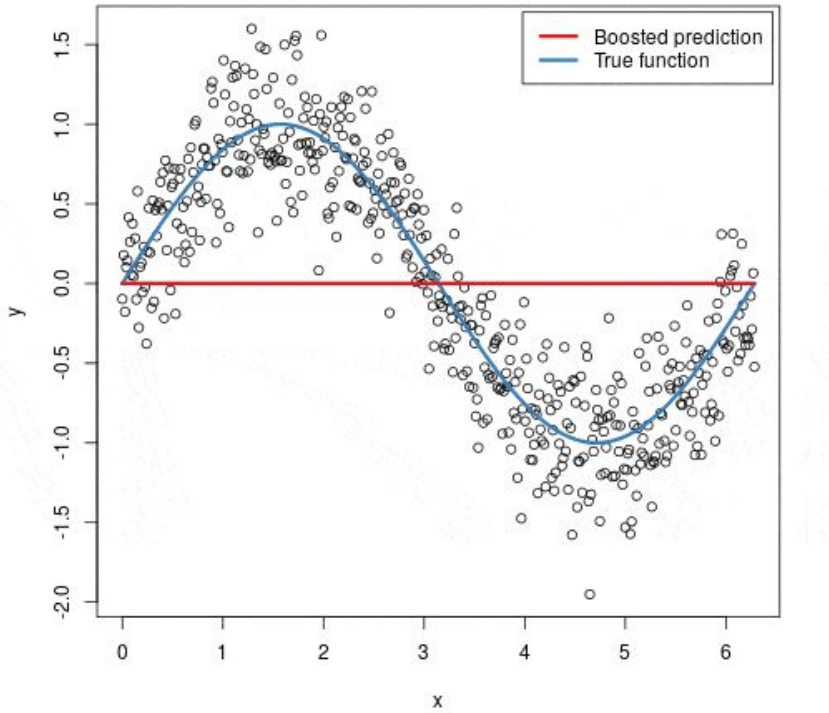
- Utilizado em problemas envolvendo classificação, pontuação e regressão;
- É um algoritmo baseado em árvore de decisão e aumento de gradiente:
  - Aumento de gradiente é a minimizar as perdas (loss) enquanto novos modelos são adicionados.

Fonte: MLK (2018).

Fonte:scikit-learn.org/.

INSTITUTO FEDERAL
Espírito Santo

Algoritmo XGBoost



Fonte:https://sigmoidal.ai/

INSTITUTO FEDERAL
Espírito Santo

Implementação do k-fold e RandomGridSearchCV

- Abaixo os parâmetros usados para se fazer as combinações:

@Parâmetros do XGBOOST

```
[ ]  1 n_jobs=[-1]
     2 n_estimators=np.arange(600,1200,600)
     3 learning_rate=[0.1]
     4 max_depth=np.arange(10)
     5
     6 param_grid={'n_jobs':n_jobs,
     7               'n_estimators':n_estimators,
     8               'max_depth':max_depth,
     9               'learning_rate':learning_rate
    10              }
```

```
1
2 num_folds=5S
3 kfold = KFold(n_splits=num_folds, shuffle=True)
4 model = XGBClassifier()
5 rs=RandomizedSearchCV(model,param_distributions=param_grid,cv=kfold,scoring='f1_micro')
6 tempo_inicial=time.time()
7 rs.fit(inputs, targets.ravel())
8 print("Tempo de treino:{} s".format(time.time()-tempo_inicial))
```

```
Tempo de treino:11993.682957649231 s
```

Tempo em horas: 3,3315.

Fonte: código desenvolvido.

INSTITUTO FEDERAL
Espírito Santo

Implementação do k-fold e RandomGridSearchCV

- Abaixo os parâmetros usados para se fazer as combinações:

```
rs.best_params_

{'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 600, 'n_jobs': -1}
```

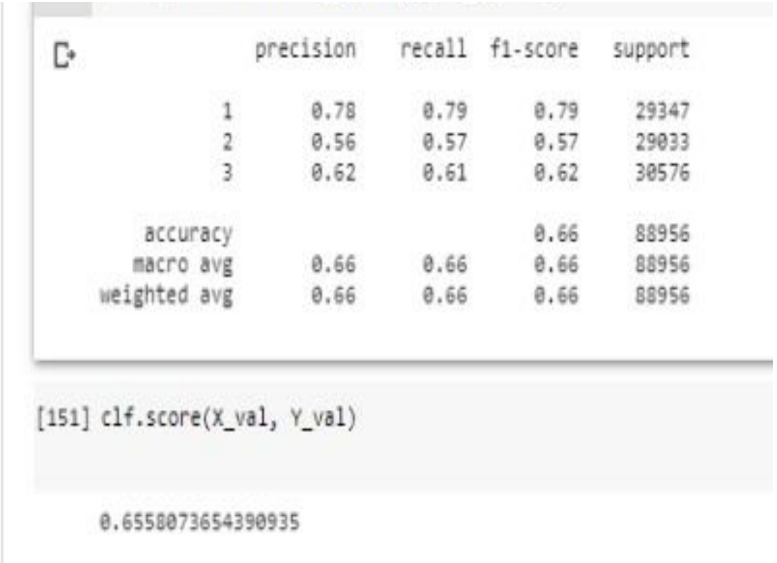
```
Tempo de treino:11993.682957649231 s
```

Fonte: código desenvolvido.

INSTITUTO FEDERAL
Espírito Santo

Implementação do PCA variância de 85%

- Quando utilizamos o PCA como seleção das features houve uma piora nos resultados:

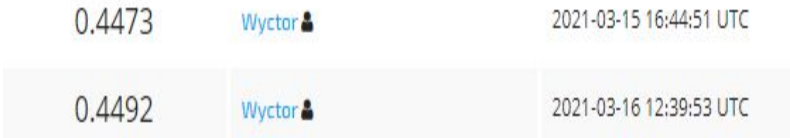


	precision	recall	f1-score	support
1	0.78	0.79	0.79	29347
2	0.56	0.57	0.57	29033
3	0.62	0.61	0.62	30576
accuracy			0.66	88956
macro avg	0.66	0.66	0.66	88956
weighted avg	0.66	0.66	0.66	88956

```
[151] clf.score(X_val, Y_val)

0.6558073654390935
```

Fonte: código desenvolvido.



0.4473	Wyctor	2021-03-15 16:44:51 UTC
0.4492	Wyctor	2021-03-16 12:39:53 UTC

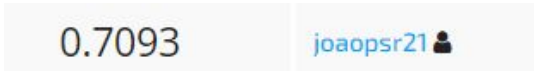
Fonte: Resultado da submissão.

INSTITUTO FEDERAL
Espírito Santo

Teste com os classificadores com melhores resultados

Com o melhor resultado prévio, foi analisado o *XGBClassifier*, utilizando as features selecionadas pelo Classificador RandomForest.

Considerando n_stimators = 200:



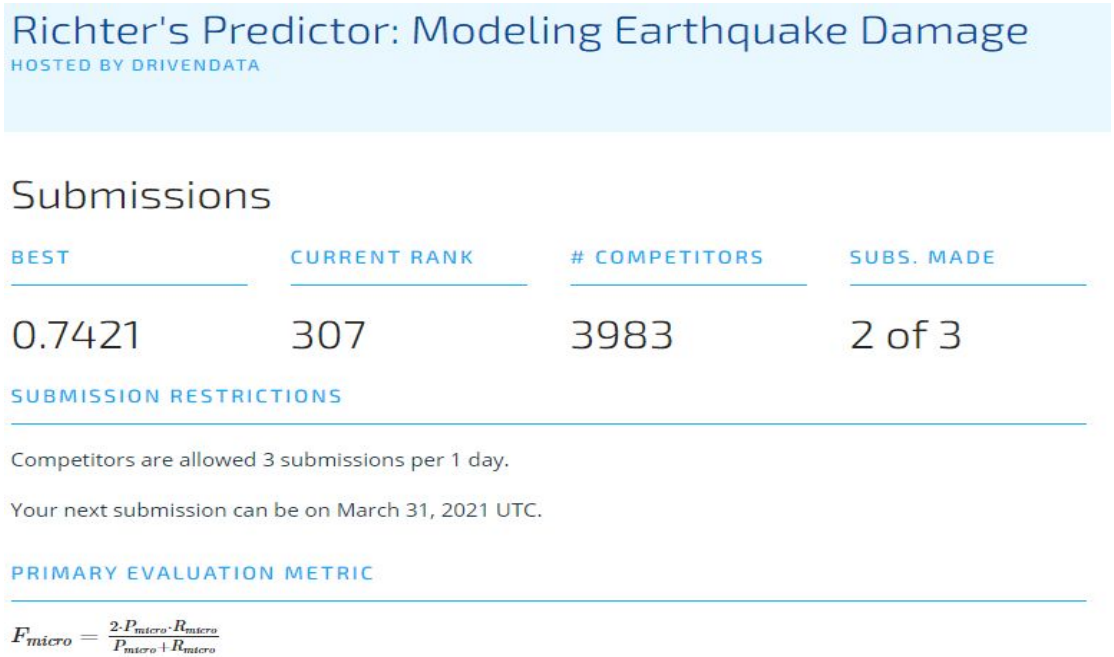
Considerando n_stimators = 600:

Resultado sem a seleção das features:

INSTITUTO FEDERAL
Espírito Santo

Teste com os classificadores com melhores resultados

Colocação final com o melhor resultado encontrado.



INSTITUTO FEDERAL
Espírito Santo