[1]

Classificação Clientes Inadimplentes

Fernanda F. Verdelho e Vinicius Cebalhos, *SIMEPAR*

Aprendizado de máquina nos proporciona ferramentas como formas de classificação onde podemos rotular, por exemplo, entre "default" ou "no default", geralmente descobrimos que em conjunto de dados de treinamento temos que cada amostra é uma classe, ou seja, da qual temos muito poucas amostras. Contudo, a base de dados apresentada tem uma característica muito importante que nos mostrando que a amostra está desbalanceada, onde normalmente afeta os algoritmos em seu processo de generalizar informações e prejudicar as classes e não ter custo computacional muito grande e resultados ruins. Quando identificamos para com isso partir para otimização da base de dados e redução de dimensionalidade e depois os testes de classificação Regressão Logística, Near Miss, Random Oversampling, Smote, Ensemble e SVM/Árvore de Decisão. Onde dessas classificações observamos suas taxas de precision, recall, F1-score e Curva Roc e do melhor algoritmo obter seleção de características para ter o melhor resultado sobre clientes inadimplentes.

I. Introdução

Em problemas de classificação em que temos de rotular, por exemplo, entre "spam" ou "não spam" (Grus, J. 2016) geralmente descobrimos que em nosso conjunto de dados de treinamento temos algumas das classes de amostra é uma classe, ou seja, da qual temos muito poucas amostras. Isso causa um desequilíbrio nos dados que usaremos para treinar nossa máquina. No caso da base de dados em questão contém 672 atributos e dados de clientes que contraíram empréstimos junto a instituição financeira. O que realizaram o pagamento em até 30 dias possuem valor zero na variável Y. Caso contrário, Y=1. A base contém 219984 registros.

Uma característica muito importante na base de dados é que são desbalanceadas, onde normalmente afeta os algoritmos em seu processo de generalizar informações e prejudicar as classes.

II. Metodologia empregada

Para trabalharmos com classificação, foram implementados os parâmetros padrões para todos os classificadores como KNN e Naive Bayes, onde inicialmente tive uma resposta ruim e um custo computacional muito grande. Assim sendo, foram realizados ajustes específicos nos classificadores testados, até perceber uma análise importante que a base de dados nos fornecia, onde foi mudada a abordagem. Foi aplicada o PCA - Análise de Componentes Principais e depois os testes de classificação 'Regressão Logística', 'Regressão Logística Balanceamento', 'Near Miss', 'Random Oversampling', 'Smote', 'Ensemble' e 'SVM/Árvore de Decisão. Onde dessas classificações observamos suas taxas de precision, recall, F1-score e Curva Roc e do melhor algoritmo obter seleção de características para ter o melhor resultado sobre clientes inadimplentes.

A. PCA - Análise de Componentes Principais

Um dos principais problemas ao trabalhar com a base de dados foi a dificuldade em trabalhar com uma base grande de dados e conseguir tirar a melhor otimização do algoritmo, como inicialmente estava empregando técnicas de classificação comum, onde o algoritmo de aprendizado for muito lento porque a dimensão de entrada é muito alta. Onde explorando os dados observaram-se que suas únicas dimensões não correspondem ao eixo x e y, uma maneira mais comum de acelerar um algoritmo de aprendizado de máquina é usando a Análise de Componentes Principais (PCA). Outra forma de otimizar entradas da PCA e acelerar um algoritmo de aprendizado de máquina (regressão logística) na base. No caso da base em questão foi aplicada a redução de PCA de n=20, onde 0,996 variação foi recuperada.

B. Regressão Logística

A primeira condição para regressão logística é que a variável de resposta (ou variável dependente) deve ser uma variável categórica. E essa variável categórica também binomial. Isso significa que deve ter apenas dois valores - 1/0. Mesmo se tiver dois valores, mas na forma de Sim / Não ou Verdadeiro / Falso, devemos primeiro converter isso na forma 1/0. O modelo de regressão logística é um dos modelos de aprendizado de máquina mais simples que é usado para classificação. E isso também apenas para a classificação de duas classes. Se sua variável de resposta tiver mais de 2 opções, como sistema de notas (onde podemos ter mais de 2 notas), não podemos aplicar regressão logística. Nesse caso, o algoritmo de floresta aleatória em python ou o algoritmo de árvore de decisão em python é recomendado.

A regressão logística em python é muito fácil de implementar e é um ponto de partida para qualquer problema de classificação binária. Ajuda a criar a relação entre uma variável dependente categórica binária com as variáveis independentes.

C. SMOTE

No que lhe concerne, quando trabalhamos com dados desbalanceados, terá um desempenho ruim em seleção de classes minoritárias, embora normalmente seja o desempenho na classe da minoria que é mais importante, uma maneira de lidar com dados desbalanceados e resolver problemas de classes minoritárias é usando subamostragem. Neste trabalho consideramos com o SMOTE, que tem como objetivo ser uma

[1]

técnica de sobreamostragem de minoria sintética.

D. *Random Oversampling*

Uma abordagem para resolver o problema de desequilíbrio de classe é reamostrar aleatoriamente o conjunto de dados de treinamento. Às duas principais abordagens para reamostrar aleatoriamente um conjunto de dados desequilibrado são excluir exemplos da classe majoritária, chamada de subamostragem, e duplicar exemplos da classe minoritária, chamada sobreamostragem. Isso quer dizer que, em vez de duplicar cada amostra na classe, algumas delas podem ser escolhidas aleatoriamente com substituição.

E. *Near Miss*

Near Miss é uma técnica de subamostragem. Em vez de reamostrar a classe minoritária, usando uma distância, isso tornará a classe majoritária igual à classe minoritária.

F. *Ensemble Balanceado*

Métodos de Ensemble é uma técnica que combina vários modelos de base para produzir um modelo preditivo ideal. Ensemble Balanced Bagging Classifier é um algoritmo de combinação que se ajusta a vários modelos em diferentes subconjuntos de um conjunto de dados de treinamento e, em seguida, combina as previsões de todos os modelos. Árvore de decisão é uma extensão de ensacamento que também seleciona aleatoriamente subconjuntos de recursos usados em cada amostra de dados.

G. *SVM - "Support Vector Machine"/ Árvore de decisão*

Como inicialmente começamos trabalhar com Regressão logística, uma alternativa que junta regressão e classificação que é o SVM - "Support Vector Machine". Os algoritmos de aprendizagem de máquina (SVM) têm como objetivo a determinação de limites de decisão que produzam uma separação ótima entre classes por meio da minimização dos erros (Vapnik, 1995). Os modelos com aprendizagem associados algoritmos de que analisam dados e produz uma precisão significativa com menos poder de computação, pode ser usado para tarefas de regressão e classificação.

A árvore de decisão é ferramenta que usa uma gráfico ou modelo das decisões e suas possíveis consequências, incluindo o acaso resultados de eventos, custos de recursos e utilidade. É uma maneira de exibir um algoritmo que contém apenas instruções de controle condicional. Árvores de decisão são comumente usadas em pesquisas operacionais, especificamente na análise de decisão, para ajudar a identificar uma estratégia com maior probabilidade de atingir uma meta, mas também são uma ferramenta popular no aprendizado de máquina.

No caso do trabalho usado em conjunto para melhorar a classificação da base.

.

H. *Feature Selection - Seleção de Caracteristicas*

Para seleção de característica usando o Select From Model é uma outra função do sklearn que funciona da seguinte forma: a partir de um modelo, o SFM irá remover todas as features que não passem do *threshold* que você informa em seus argumentos.

III. Experimentos

Como as informações não estão ligadas devidamente a clientes específicos, não sabemos realmente o que significam os recursos e eles são chamados de V1, V2, V3, V4, etc. Sabemos que no eixo Y temos valores de classes são 0 e 1, onde a variável 0 realizaram o pagamento em até 30 dias e 1 que não realizaram. Como você pode imaginar, o conjunto de dados é muito desbalanceado e teremos muito poucas amostras rotuladas como inadimplentes.

Vemos que existem 219984 linhas e apenas 672 atributos, onde dessas tenho 174921 são a classe minoritária com casos de inadimplentes. Eles representam 0,795% das amostras.

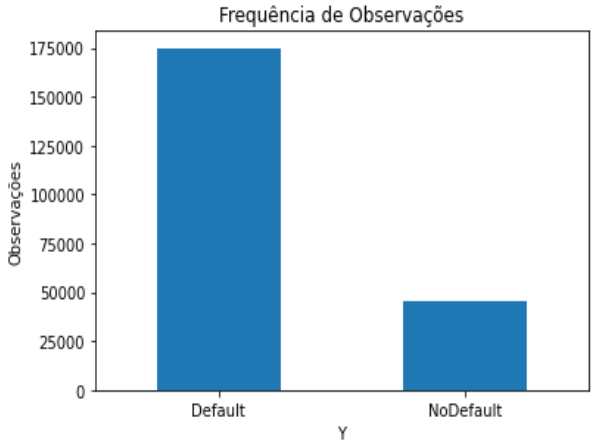

Fig.1 - Frequência de observações da amostra de pagamento nos últimos 30 dias (No Default) e inadimplência (Default) na variável Y.

A. *Análise Geral*

Montando modelos sem e com balanceamento, com qualquer uma das técnicas que aplicamos, melhorar o modelo de classificação de Regressão Logística, Regressão Logística Balanceamento, Near Miss, Random Oversampling, Smote, Ensemble e SVM/Árvore de Decisão.

Já o ***F1 Score***, que é a média ponderada de precisão e recuperação, essa pontuação considera tanto os falsos positivos quanto os falsos negativos. Se o custo de falsos positivos e falsos negativos for muito diferente, é melhor olhar para ***Precision*** e ***Recall separados***.

Todos os algoritmos foram testados(test), treinados(train) e validados(cross validation),onde foram feitos teses de 50%, 30% e 20%.

	algoritmo	precision	recall	F1	Roc	overall
0	Regressão Logística	0.74	0.79	0.71	0.50	0.6850
6	SVM/Árvore de Decisão	0.70	0.70	0.70	0.53	0.6575
1	Regressão Logística Balanceamento	0.68	0.56	0.60	0.51	0.5875
4	Smote	0.68	0.56	0.60	0.51	0.5875
3	Random Oversampling	0.68	0.55	0.59	0.51	0.5825
5	Ensemble	0.65	0.33	0.34	0.48	0.4500
2	NearMiss	0.70	0.31	0.32	0.46	0.4475

Fig.2 - Experimentos com algoritmos testados(test), treinados(train) e validados(cross validation), onde foram feitos testes de 50% para 'Regressão Logística', 'Regressão Logística Balanceamento', 'Near Miss', 'Random Oversampling', 'Smote', 'Ensemble' e 'SVM/Árvore de Decisão, para obter análise geral através dos dados de precision, recall, f1-score e curva roc.

	algoritmo	precision	recall	F1	Roc	overall
0	Regressão Logística	0.74	0.79	0.71	0.50	0.6850
1	Regressão Logística Balanceamento	0.69	0.67	0.68	0.52	0.6400
3	Random Oversampling	0.69	0.67	0.68	0.52	0.6400
4	Smote	0.69	0.67	0.68	0.52	0.6400
6	SVM/Árvore de Decisão	0.67	0.63	0.58	0.49	0.5925
2	NearMiss	0.70	0.39	0.42	0.52	0.5075
5	Ensemble	0.65	0.31	0.30	0.48	0.4350

Fig.3 - Experimentos com algoritmos testados(test), treinados(train) e validados(cross validation), onde foram feitos testes de 30% para 'Regressão Logística', 'Regressão Logística Balanceamento', 'Near Miss', 'Random Oversampling', 'Smote', 'Ensemble' e 'SVM/Árvore de Decisão, para obter análise geral através dos dados de precision, recall, f1-score e curva roc.

	algoritmo	precision	recall	F1	Roc	overall
0	Regressão Logística	0.74	0.80	0.71	0.50	0.6875
1	Regressão Logística Balanceamento	0.69	0.66	0.68	0.53	0.6400
3	Random Oversampling	0.69	0.66	0.68	0.53	0.6400
4	Smote	0.69	0.65	0.67	0.52	0.6325
6	SVM/Árvore de Decisão	0.69	0.60	0.63	0.52	0.6100
5	Ensemble	0.65	0.32	0.32	0.48	0.4425
2	NearMiss	0.63	0.33	0.34	0.46	0.4400

Fig.4 - Experimentos com algoritmos testados(test), treinados(train) e validados(cross validation), onde foram feitos testes de 20% para 'Regressão Logística', 'Regressão Logística Balanceamento', 'Near Miss', 'Random Oversampling', 'Smote', 'Ensemble' e 'SVM/Árvore de Decisão, para obter análise geral através dos dados de precision, recall, f1-score e curva roc.

Com qualquer uma das técnicas que aplicamos, melhoramos o modelo inicial de Regressão Logística, que alcançou um recall de 0,80 para a classe. E não vamos esquecer que existe um enorme desequilíbrio de classes no conjunto de dados. Os algoritmos como Regressão Logística Balanceamento, SVM/Árvore de decisão e Random Oversampling não se saíram tão ruins também. Sempre vemos uma melhoria em relação ao modelo inicial com um recall de 0,66 até 0,70 e f1-score 0,68 e 0,70 dependendo do experimento entre os testes, treino e validação.

Após aplicação dos experimentos o algoritmo de Regressão logística com aplicação de redução de dimensionalidade PCA, se saiu como melhor algoritmo em observando o "overall" da média ponderada "weighted avg" entre precision, recall, f 1 e curva roc.

B. Regressão Logística com Classificador de Características.

Após aplicação dos experimentos o algoritmo de Regressão logística com aplicação de redução de dimensionalidade PCA, aqui vemos a matriz de confusão e na classe 2(0:1), vemos 51 falhas e 35001 acertos, dando um recall de 0,50 "macro avg" e 0,80 "weighted avg" e esse é o valor que queremos melhorar. Também na coluna f1-score obtemos resultados muito bons de acurácia de 0,80, mas eles realmente não devem nos enganar, pois, refletem uma realidade parcial. A verdade é que nosso modelo não pode detectar corretamente os casos de inadimplência.

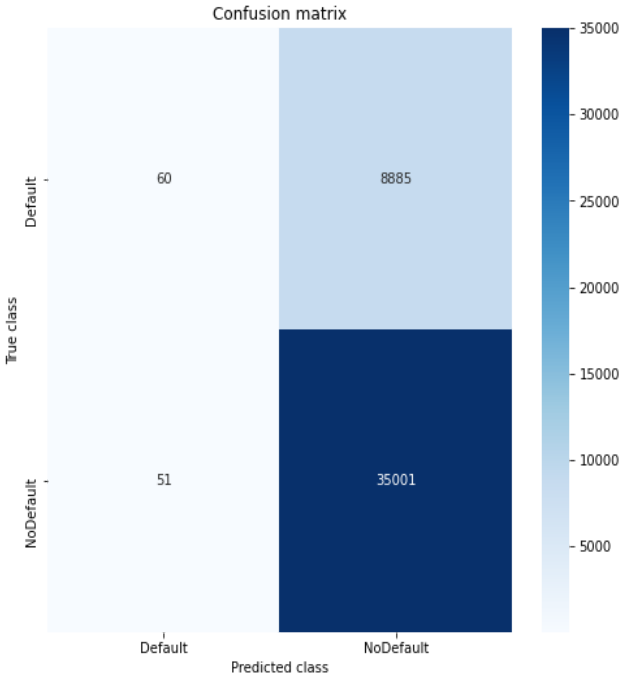

Fig.5 - Matriz de Confusão do algoritmo de regressão logística com PCA.

	precision	recall	f1-score	support
0	0.54	0.01	0.01	8945
1	0.80	1.00	0.89	35052
accuracy			0.80	43997
macro avg	0.67	0.50	0.45	43997
weighted avg	0.75	0.80	0.71	43997

0.5026263383693074

Fig.6 - Precision, Recall, F1-score e Curva-Roc do algoritmo Regressão Logística com PCA.

Após os primeiros testes, foi aplicado a classificação de características para esse algoritmo.

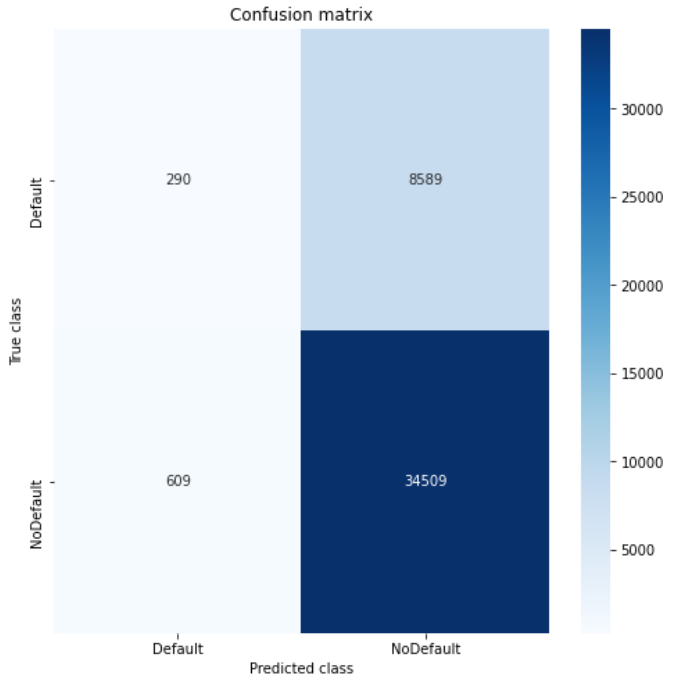

Fig.7 - Matriz de Confusão do algoritmo de regressão logística com PCA e aplicado a seleção de características.

	precision	recall	f1-score	support
0	0.32	0.03	0.06	8879
1	0.80	0.98	0.88	35118
accuracy			0.79	43997
macro avg	0.56	0.51	0.47	43997
weighted avg	0.70	0.79	0.72	43997

0.5076599007400346

Fig. 8 - Precision, Recall, F1-score e Curva-Roc do algoritmo Regressão Logística com PCA e aplicado seleção de características.

Assim que aplicados o modelo de seleção de características vemos mudança nas falhas da matriz de confusão, onde temos 609 falhas e 34509 acertos, dando um recall de 0,51 "macro avg" e 0,70 "weighted avg" e esse é o valor que queremos melhorar. Na coluna f1-score obtemos resultados muito bons de acurácia de 0,79, mas eles realmente não devem nos enganar, refletem uma realidade parcial. Como já observado na matriz anterior, sem seleção de característica nosso modelo não pode detectar corretamente os casos de inadimplência.
A Curva Roc nos decepciona no resultado forçando o modelo chegamos a 0,55 de acurácia, um resultado bastante ruim. Nesse caso, modelos de classificação derivados a partir da curva mais próxima do ponto 0,100% serão melhores que os modelos derivados a partir da outra curva, independentemente das condições operacionais, algo que não ocorreu neste modelo margem para falsos positivos. Em suma, implica que realmente não existe um valor de corte em que a troca de sensibilidade valha a pena. Brincar com o ponto de corte não melhora muito as classificações, pois diminuiria a especificidade em cerca do mesmo tempo que aumentaria a sensibilidade.

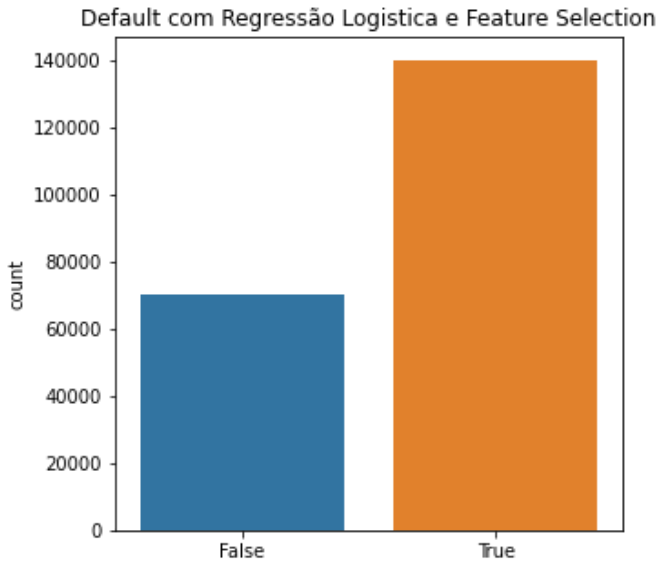

Fig.9 - Default com Regressão Logística e Feature Selection para os valores de inadimplência.

Default com Regressão Logística e Feature Selection para os valores de inadimplência (1), onde após a contagem do amostras temos Verdadeiro = 139803 e Falso = 69901.

IV. Conclusão

Ao trabalharmos com os classificadores como já dito no início do experimento, foram implementados os parâmetros padrões para todos os classificadores como KNN e Naive Bayes, onde inicialmente tive uma resposta ruim e um custo computacional muito grande e resultados ruins. Quando identificamos os erros e partimos para otimização da base de dados tanto usando regressão logística e algoritmos que nos ajudava a desbalancear a base, podemos testar novas formas de classificação, onde dos classificadores testados observamos suas taxas de precision, recall, F1-score e Curva Roc e do melhor algoritmo obter seleção de características para ter o melhor resultado sobre clientes inadimplentes, onde uma das

técnicas que aplicamos, o modelo inicial de Regressão Logística, que alcançou um recall de 0,80 "weighted avg" para a classe, onde usamos testes, treinamento e validação de 20%. Ao aplicados o modelo de seleção de características vemos mudança nas falhas da matriz de confusão, onde temos 609 falhas e 34509 acertos, dando um recall de 0,51 "macro avg" e 0,70 "weighted avg". Como já observado na matriz da regressão logística com seleção de características nosso modelo não pode detectar corretamente os casos de inadimplência, mas nos deu valores que inicialmente eram de 174921 (0,795%) são a classe minoritária com casos de inadimplentes, para 139803 (0,635%) após a aplicação do modelo de regressão logística com redução de dimensionalidade PCA e seleção de características.

Referências

https://sites.icmc.usp.br/gbatista/files/ieee_la2008.pdf

https://jair.org/index.php/jair/article/view/10302

https://medium.com/@saeedAR/smote-and-near-miss-in-python-machine-learning-in-imbalanced-datasets-b7976d9a7a79

https://www.irjet.net/archives/V4/i8/IRJET-V4I857.pdf

https://imbalanced-learn.readthedocs.io/ principais modelos usados

https://www.kaggle.com/nixiam/credit-card-fraud-detection-smote-classifier importante de onde eu tirei a base

Chen, Chao, Andy Liaw, and Leo Breiman. "Using random forest to learn imbalanced data." University of California, Berkeley 110, 2004. - essa parte do ensemble

https://www.worldscientific.com/doi/abs/10.1142/S0218213007003163 -svm

Vapnik, V. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995. - svm

https://www.researchgate.net/publication/224330873_ADASYN_Adaptive_Synthetic_Sampling_Approach_for_Imbalanced_Learning

He, Haibo & Bai, Yang & Garcia, Edwardo & Li, Shutao. (2008). ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. Proceedings of the International Joint Conference on Neural Networks. 1322 - 1328. 10.1109/IJCNN.2008.4633969.

https://machinelearningmastery.com/framework-for-imbalanced-classification-projects/

Manski, Charles F., and Steven R. Lerman. "The Estimation of Choice Probabilities from Choice Based Samples." *Econometrica* 45, no. 8 (1977): 1977-988. Accessed October 01, 2020. doi:10.2307/1914121.

King, Gary, and Langche Zeng. 2001. Logistic regression in rare events data. Political Analysis 9(2): 137-163.

https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/

https://www.kdnuggets.com/2019/05/fix-unbalanced-dataset.html

Grus, Joel. Data Science do Zero: Primeiras Regras com o Python.2016