

PORTFÓLIO DE PROJETOS:

PROJETO 02.

ESTUDO DE CASO:

PREDIÇÃO DE UM EMPRÉSTIMO (LOAN PREDICTION PROBLEM)

- PROJETO DE ETL (Extract Transform and Load),
APLICAÇÃO DE ALGORITMOS DE MACHINE
LEARNING E DEEP LEARNING COM LINGUAGEM DE
PROGRAMÇÃO PYTHON.

RELATÓRIO CONCLUSIVO DE ESTUDOS.

DATA: 16/05/2022

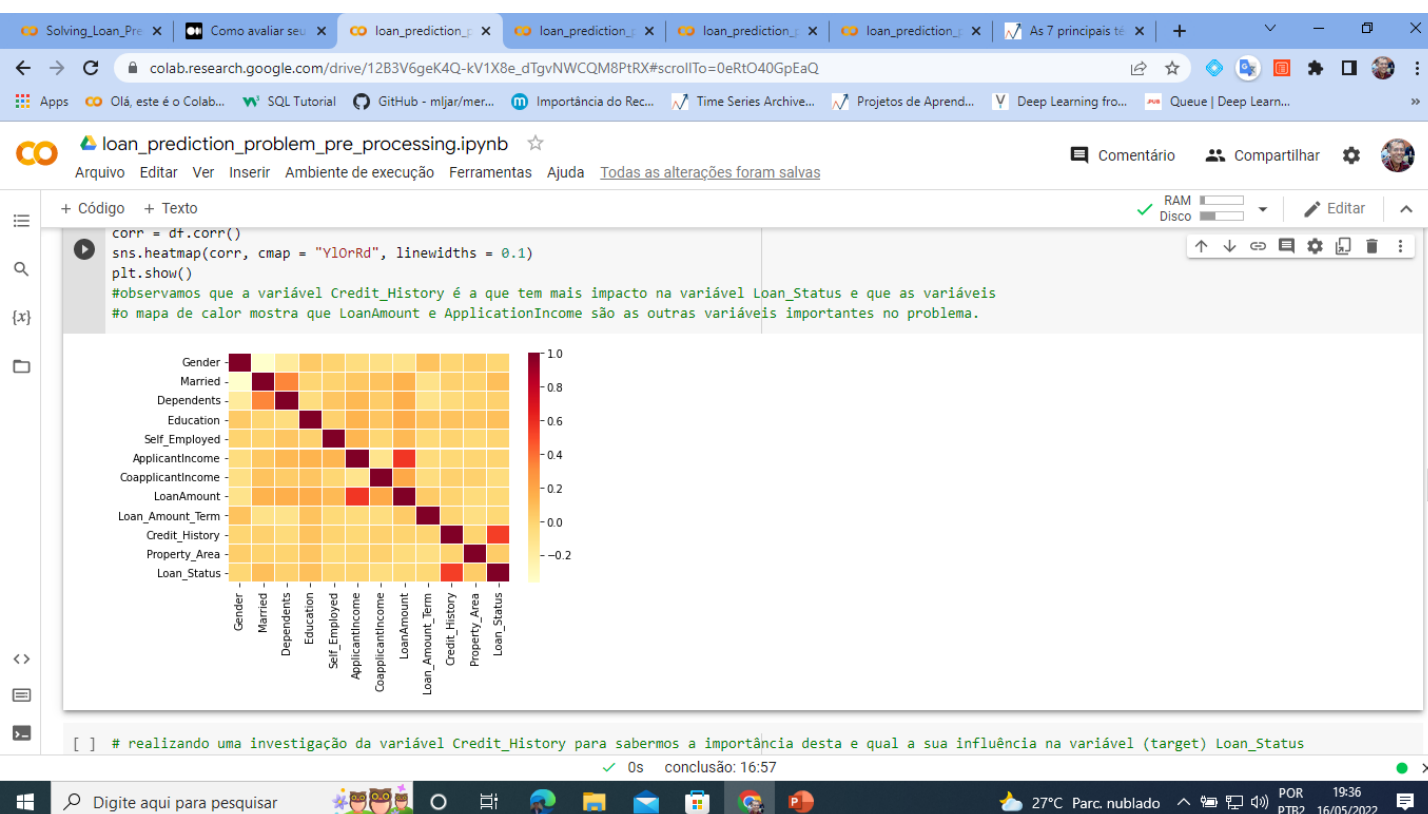
Descrição do Problema:

O Loan Prediction Problem é um problema de classificação no qual é preciso classificar se a solicitação de um empréstimo será aprovada ou não. Neste sentido uma Financeira precisa analisar a solicitação de empréstimo de um cliente e para realizar isto a empresa precisa de dados do cliente. Então a empresa solicita ao cliente que preencha um formulário com informações como; sexo, estado civil, grau de instrução, número de dependentes, renda, valor do empréstimo solicitado, histórico de crédito, dentre outros para que ela possa estudar a possibilidade do empréstimo.

- A empresa pretende automatizar o processo de elegibilidade de empréstimo com base nas informações detalhadas pelo cliente, sendo assim o gerente da empresa forneceu ao analista de dados um dataset para que este possa identificar os segmentos de clientes elegíveis para valores de empréstimo.
- O gerente precisa saber qual a probabilidade de um dado cliente honrar com o pagamento do empréstimo adquirido.

1ª Parte: Pre-Processing dos dados

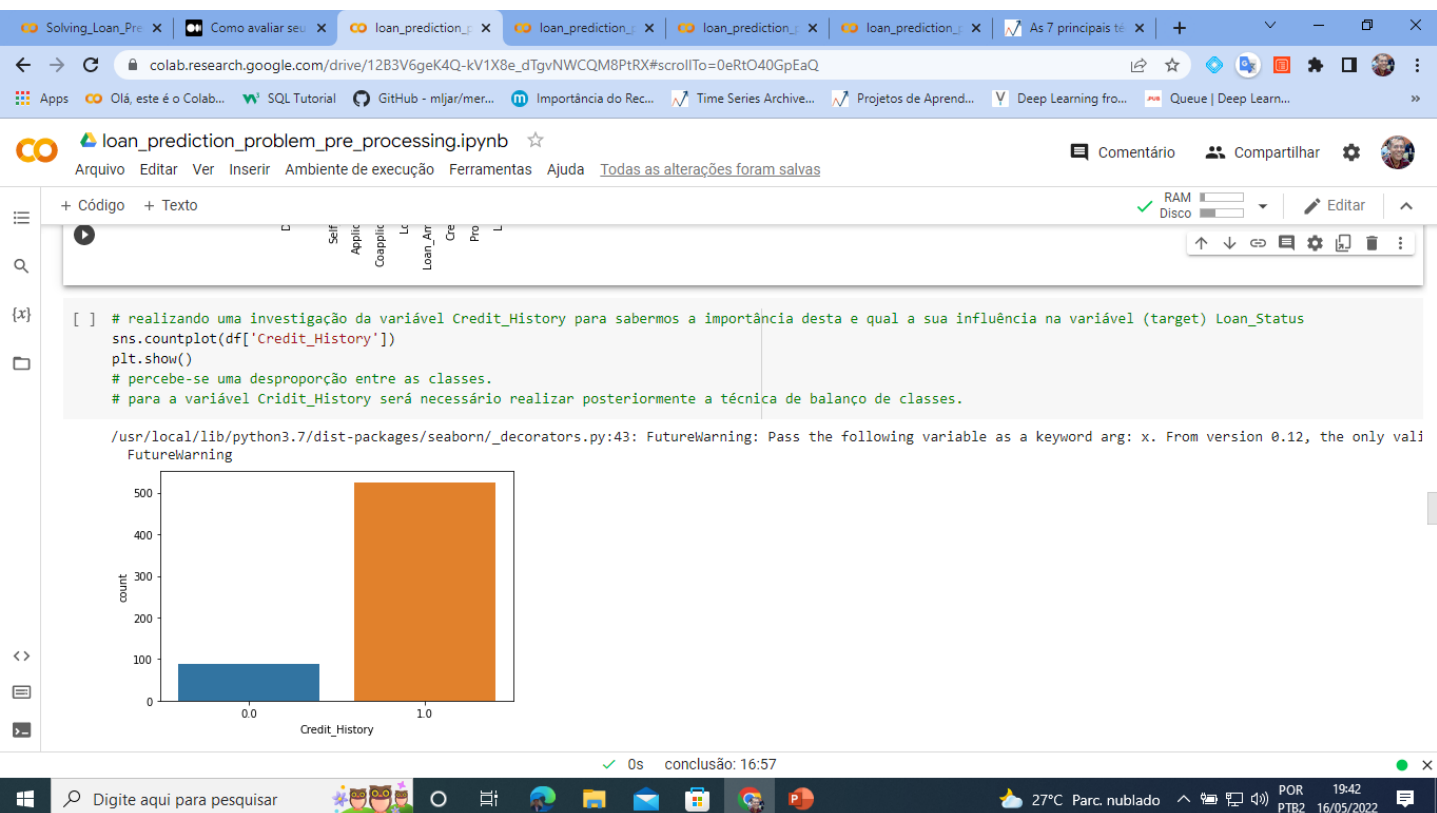
- Nesta parte foram feitas diversas modificações no dataset original (load_prediction.csv).
- Aplicou-se as técnicas de imputação de valores ausentes, cálculo de indicadores estatísticos (média aritmética, moda e mediana) e distribuição normal e normalização.
- Com o dataset modificado foi construído um mapa de calor para identificação das correlações mais fortes entre as variáveis independentes do problema.



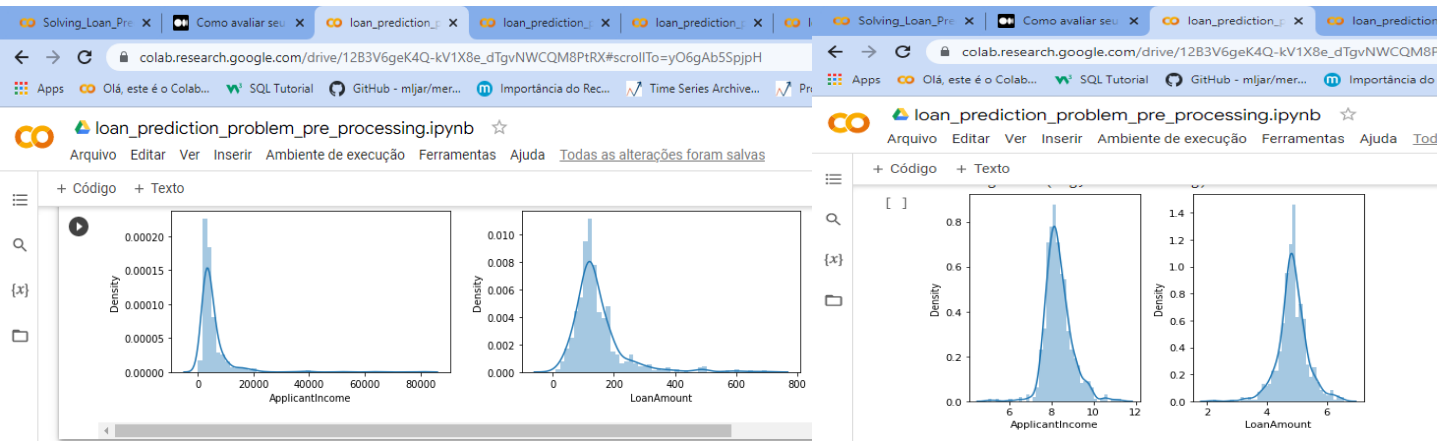
- Nesta etapa conclui-se que as variáveis Credit_History, LoanAmount e ApplicantIncome deveriam ter uma atenção especial.

Continuação da análise:

A variável Credit_History apresentou um elevado desbalanço de classes, sendo assim será necessário realizar posteriormente a técnica de balanço de classes para esta variável.



As variáveis ApplicantIncome e LoanAmount passaram por um processo de ajuste no 'shape' da distribuição normal pois estas estavam muito 'assimétricas'.



Após as transformações no dataset original foi gerado um novo dataset.

O código em python está no notebook:

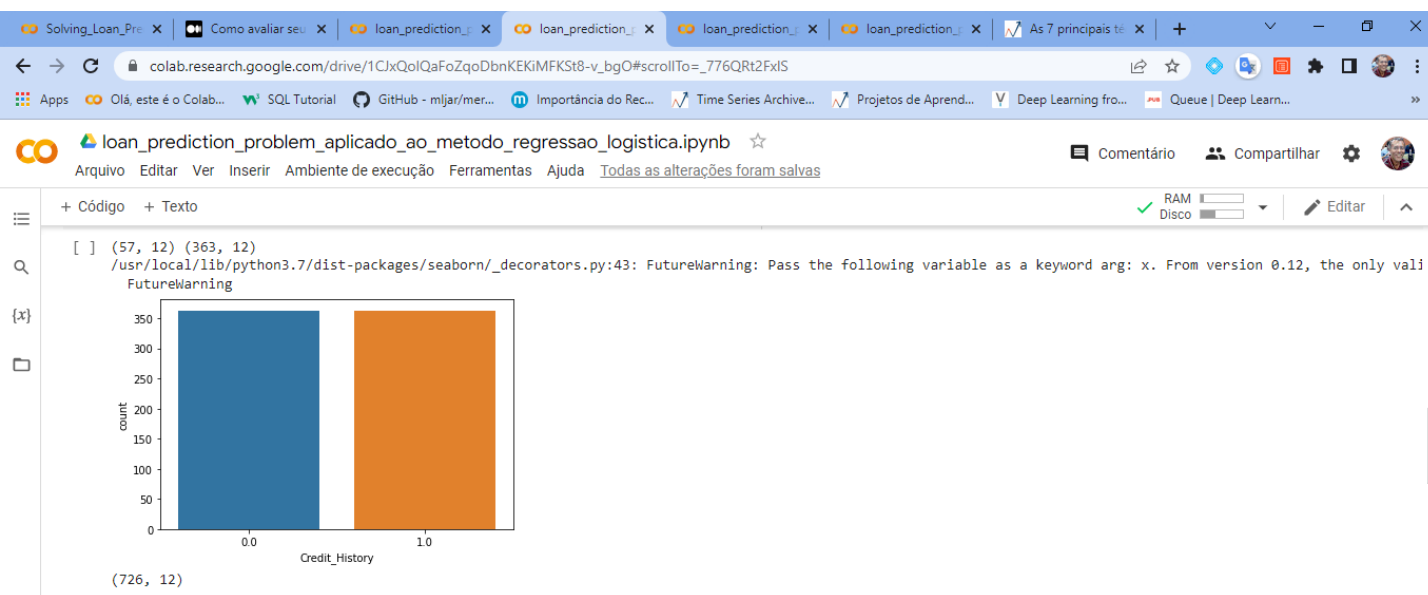
loan_prediction_problem_pre_processing.ipynb

Continuação:

O conjunto de dados do dataset foi separado em três conjuntos. Um conjunto de treino com 420 dados, um de teste com 180 dados e um para verificação com 13 dados.

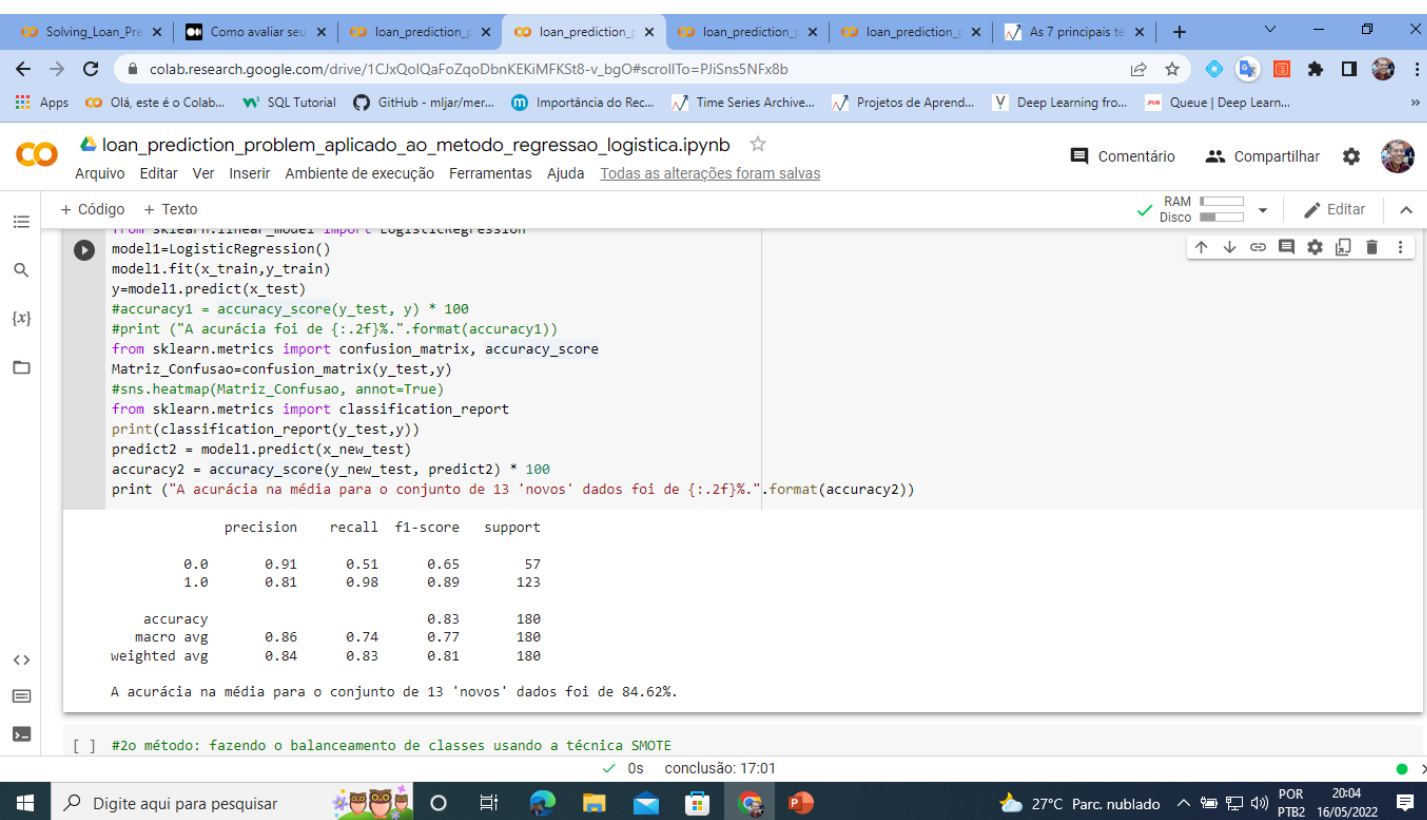
Foi feito o balanceamento de classes no conjunto de treino usando as técnicas resample e SMOTE. Estas técnicas são indicadas para que o algoritmo não tome maior tendência para uma dada classe.

O print abaixo mostra o resultado final do balanço de classes.



2ª Parte: Aplicação de Algoritmos de Machine Learning ao Problema.

- 1º Algoritmo: Regressão Logística
- A aplicação deste algoritmo resultou numa exatidão (accuracy) de 83% para o conjunto de dados de teste.
- Usando-se o conjunto de 13 dados que foram separados obteve-se uma exatidão que na média resultou em 84%.
- A precisão em classificar cada classe ficou cerca de 80% e 90%.



```
from sklearn.linear_model import LogisticRegression
model1=LogisticRegression()
model1.fit(x_train,y_train)
y=model1.predict(x_test)
#accuracy1 = accuracy_score(y_test, y) * 100
#print ("A acurácia foi de {:.2f}%".format(accuracy1))
from sklearn.metrics import confusion_matrix, accuracy_score
Matriz_Confusao=confusion_matrix(y_test,y)
#sns.heatmap(Matriz_Confusao, annot=True)
from sklearn.metrics import classification_report
print(classification_report(y_test,y))
predict2 = model1.predict(x_new_test)
accuracy2 = accuracy_score(y_new_test, predict2) * 100
print ("A acurácia na média para o conjunto de 13 'novos' dados foi de {:.2f}%".format(accuracy2))
```

	precision	recall	f1-score	support
0.0	0.91	0.51	0.65	57
1.0	0.81	0.98	0.89	123
accuracy			0.83	180
macro avg	0.86	0.74	0.77	180
weighted avg	0.84	0.83	0.81	180

A acurácia na média para o conjunto de 13 'novos' dados foi de 84.62%.

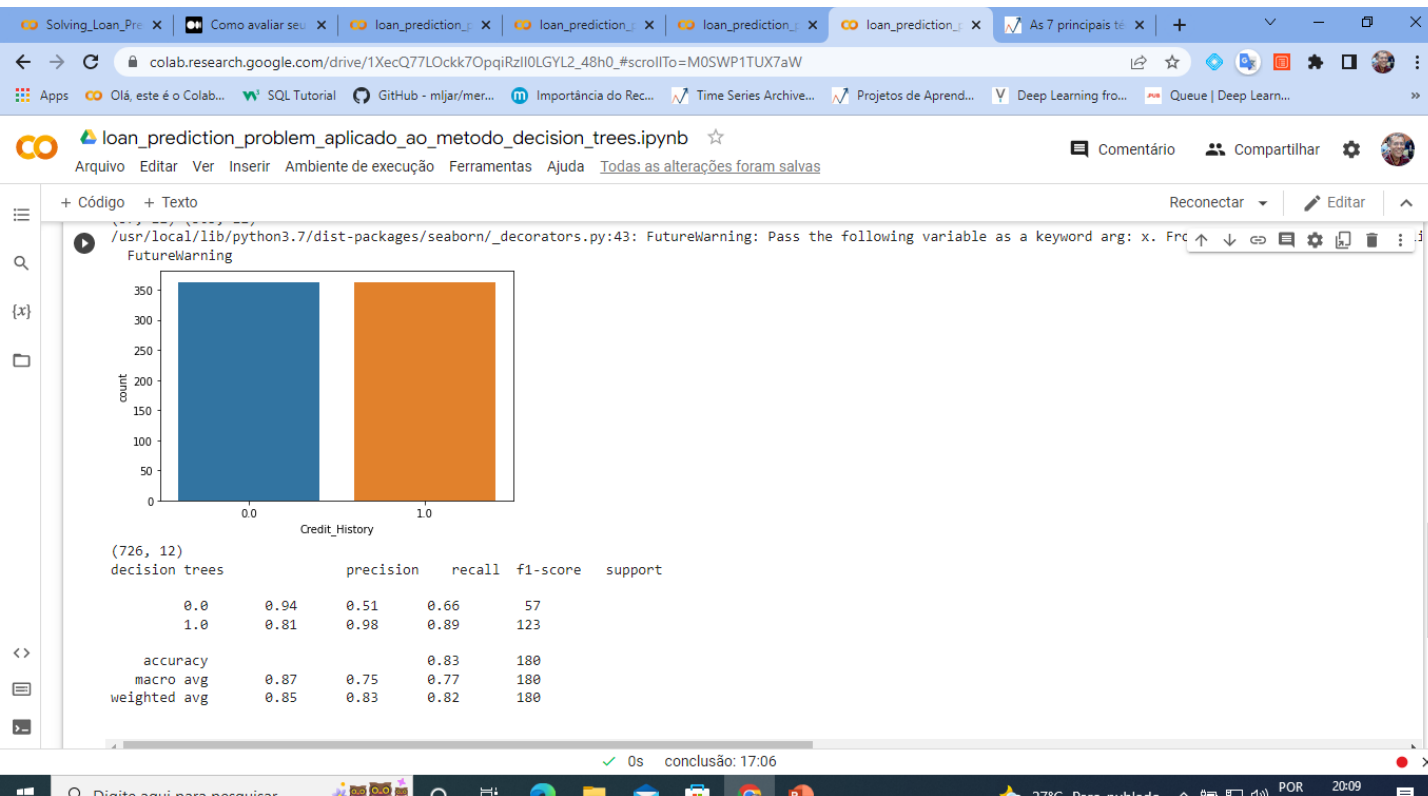
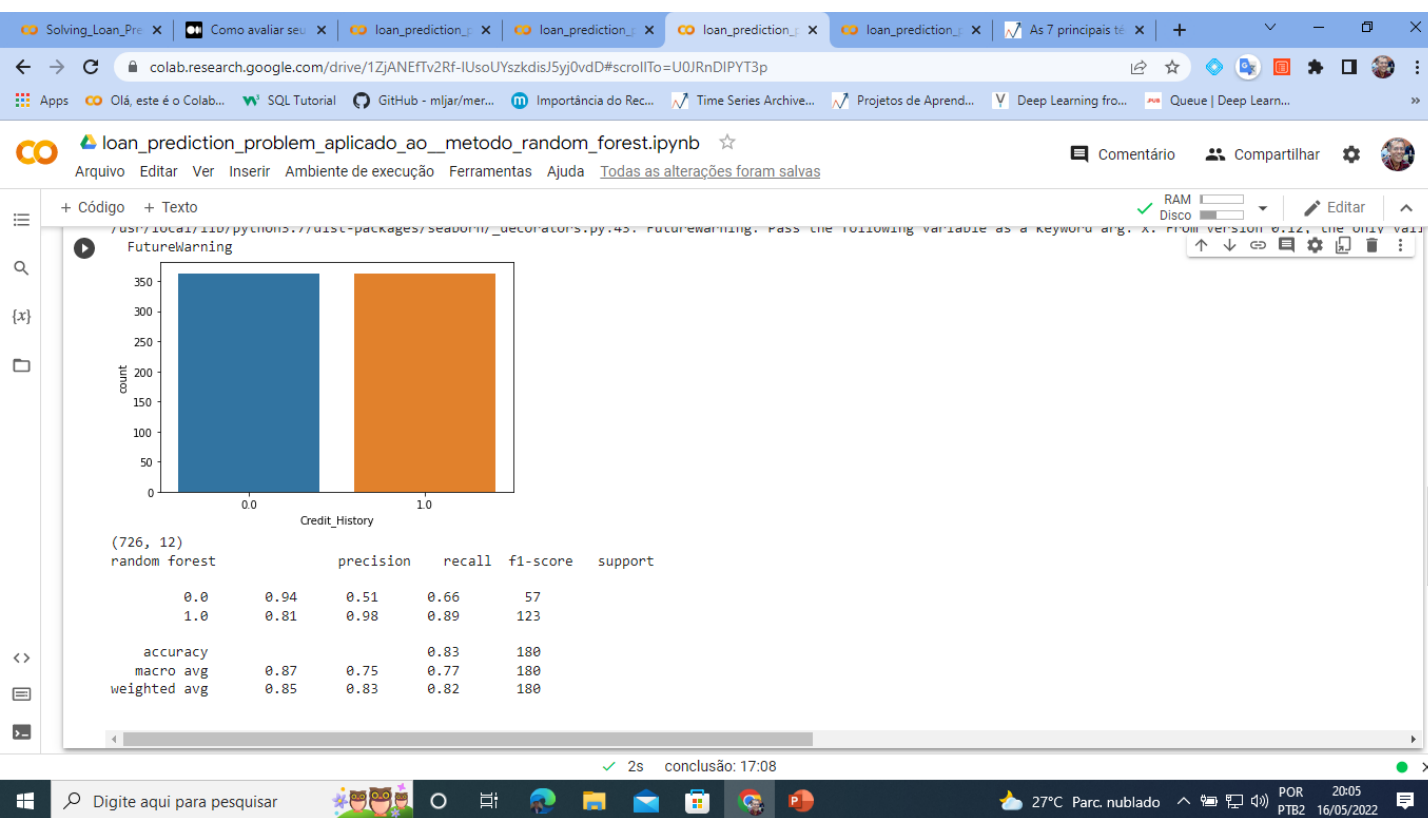
[] #2o método: fazendo o balanceamento de classes usando a técnica SMOTE

✓ 0s conclusão: 17:01

Continuação:

A seguir foi feito o estudo com outros dois algoritmos; random forest e decision trees.

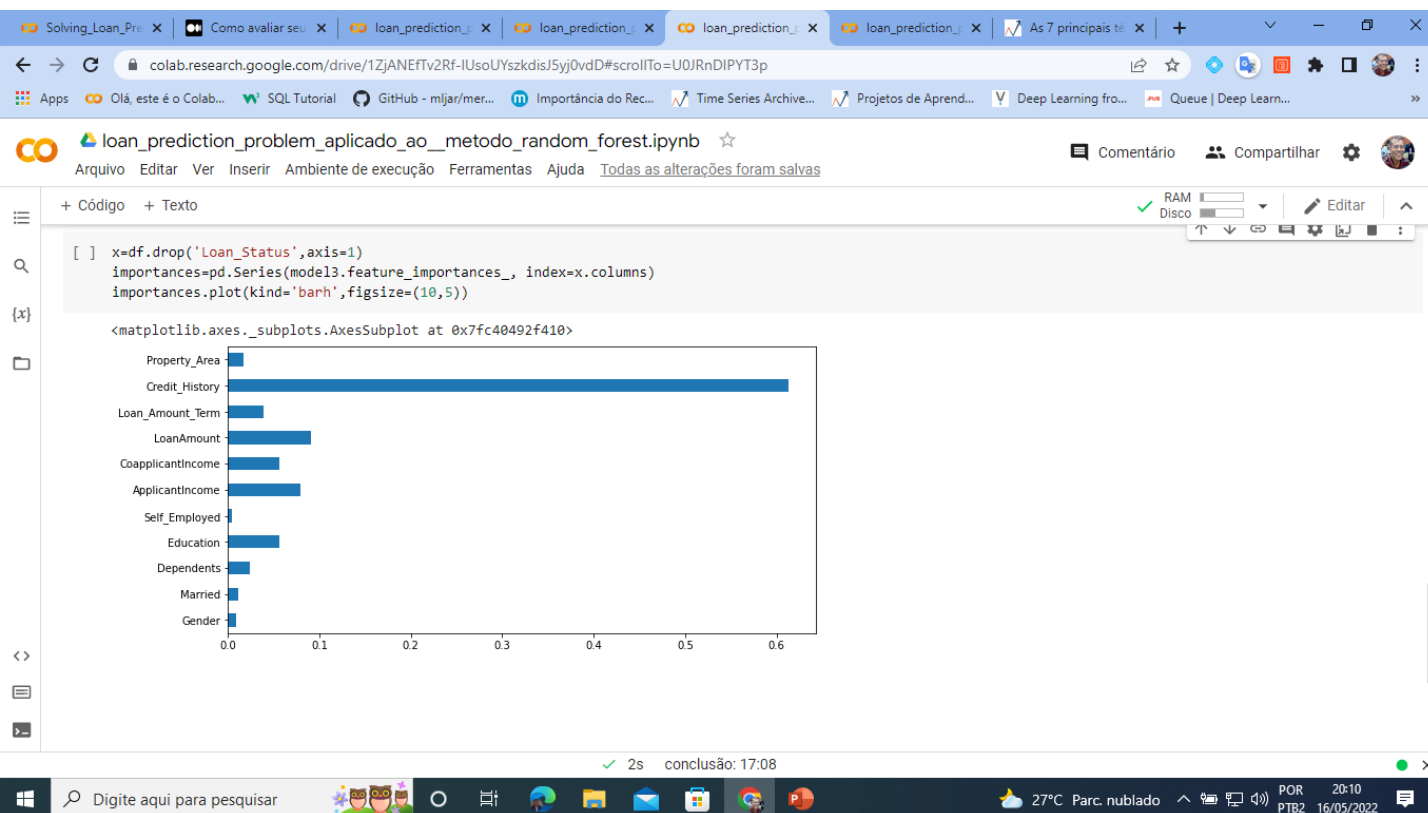
- Os prints a seguir mostram que estes algoritmos praticamente deram o mesmo resultado que o linear regression com exatidão de 83%.



Continuação:

Foi feito o plot que analisa a importância das variáveis independentes.

Após análise deste plot concluí-se, da mesma maneira, que a variável Credit_History é a mais importante do problema, seguida das variáveis LoanAmount e ApplicantIncome.



Os notebooks dos algoritmos foram:

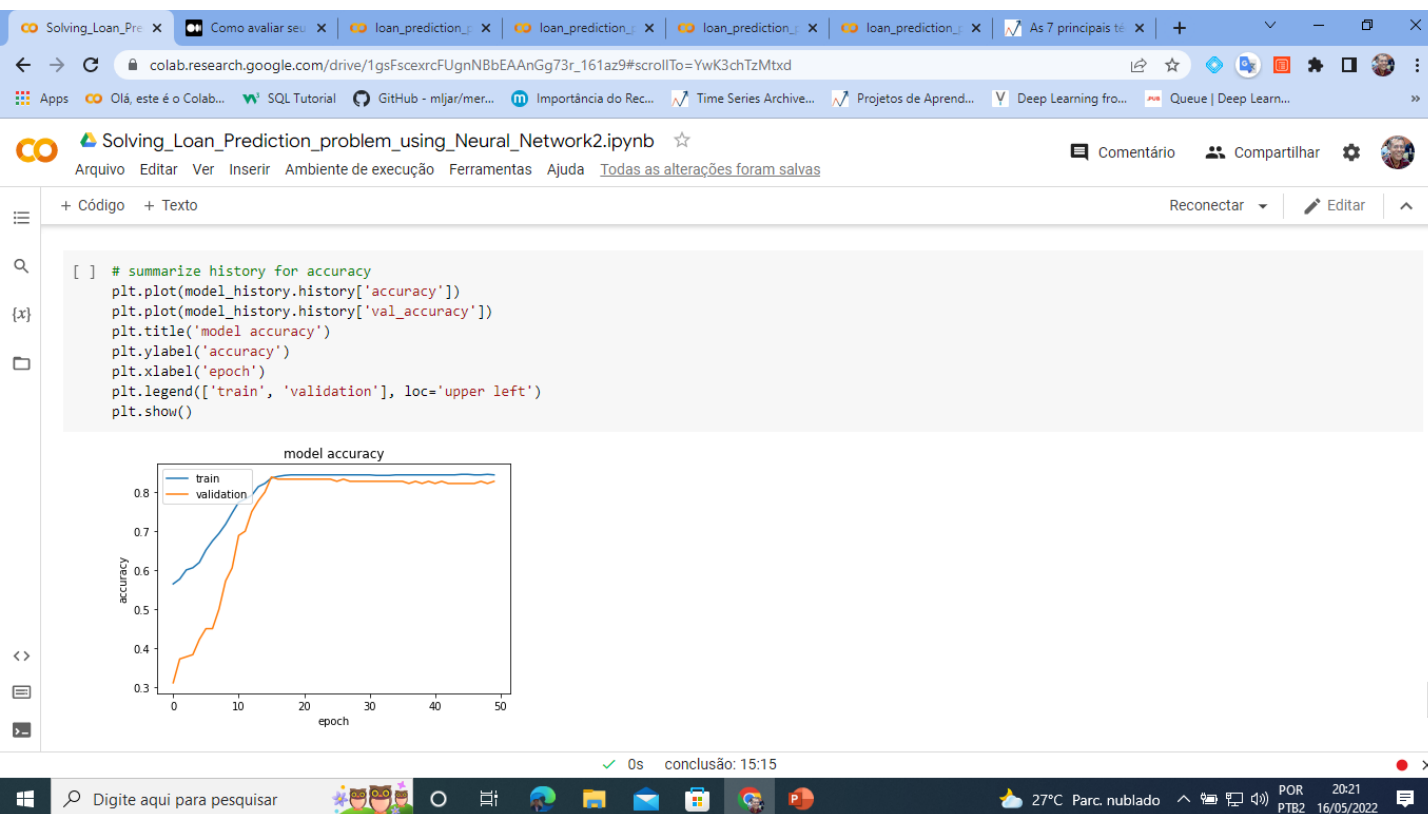
loan_prediction_problem_aplicado_ao_metodo_regressao_logistica.ipynb

loan_prediction_problem_aplicado_ao__metodo_random_forest.ipynb

loan_prediction_problem_aplicado_ao_metodo_decision_trees.ipynb

Continuação:
Foi construído um modelo de redes neurais para o problema e o resultado obtido foi também uma exatidão de 83%.

- Este algoritmo está no notebook:
- Loan_Prediction_problem_using_Neural_Network.ipynb



Conclusão Final do Estudo:

- Foram aplicadas diversas técnicas de ETL ao problema com o intuito de tratar o dataset original.

Isto mostrou que algumas variáveis tinham maior importância que outras; sobre tudo as variáveis Credit_History, LoanAmount e ApplicantIncome.

- Foram usados três algoritmos de ML;

logistic_regression, random_forest e decision_trees e ambos mostraram ter praticamente a mesma exatidão nos resultados da previsão de empréstimo. Algo em torno de 83% o que é uma previsão excelente.

- O mesmo resultado chegou-se com a aplicação de uma rede neural artificial.

- A precisão no 'acerto' de sim ou não para um empréstimo ficou em torno de 90% e 80% o que também é uma previsão excelente.