Projeto UC – Inteligência Artificial

Professores Responsáveis:

● Cleber Silva

● Vinicius

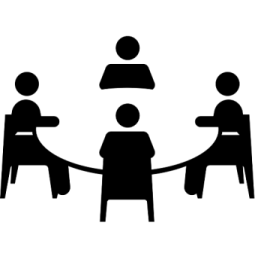
Grupo:

**Aluno1:** Jorge Leandro Piva RA: 820268722

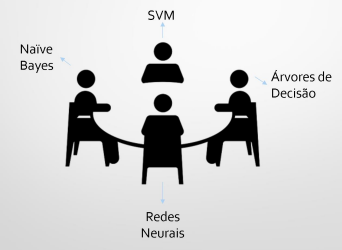
**Aluno2:** Valério Vanuzi Pereira RA: 32213195

O que será o projeto?

O projeto da UC de Inteligência Artificial será fazer um Comitê de Classificadores



Ou seja, cada grupo analisará uma base de dados e aplicará métodos de IA para chegar a algum resultado, após a aplicação desses métodos, deverá compará-los para chegar à conclusão de qual teve o melhor resultado, melhor custo benefício, etc.



Quais os métodos que poderão ser aplicados?

● Árvores de Decisão

● Naive Bayes

● Redes Neurais

● SVM

● KNN

● etc

Base de dados

Foi escolhida um dataset no Kaggle sobre Fraudes bancárias. Esta UC está sendo ministrada simultaneamente com a UC de Análise de dados e Big Data, desta forma utilizamos o mesmo dataset para as duas atividades A3, porém, estamos utilizando técnicas de análise exploratória com Pandas e Numpy além de criação de alguns gráficos com Matplotlib e Seaborn para facilitar algumas visualizações, estas técnicas nos permite conhecer melhor os dados que serão utilizados nos algoritmos.

O dataset pode ser encontrado no site do Kaggle no link.

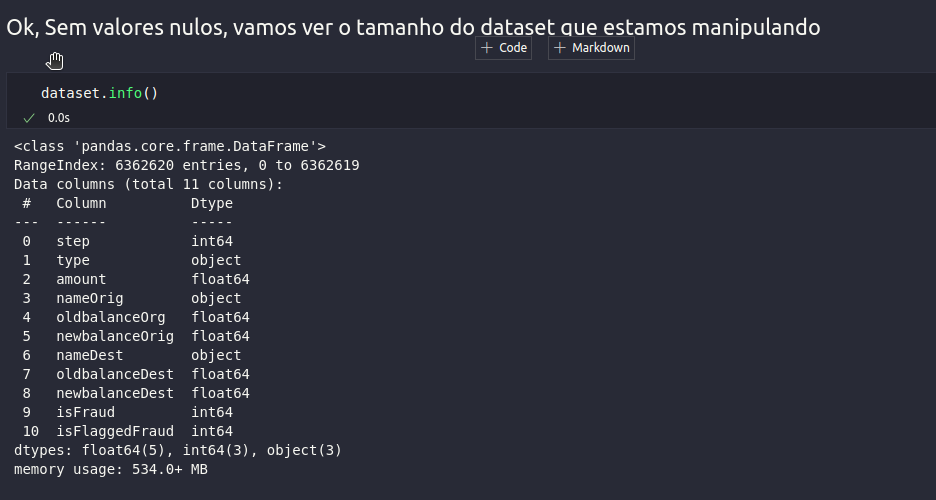
<https://www.kaggle.com/datasets/chitwanmanchanda/fraudulent-transactions-data>

Se por algum motivo o Kaggle remover o dataset do ar fizemos um backup no google drive que pode ser baixado por qualquer pessoa através do link:

<https://drive.google.com/file/d/1Uwtd9_sW51Au0M1RtDBCGHL7Q9Ar14Gu/view?usp=sharing>

Este dataset é muito grande, já que estamos falando de IA e Análise de Dados e “Big Data” tentamos pelo menos trazer um V desse big data com um conjunto de dados que não seria possível nem abri-lo no excel, ou seja, teriamos que analisar com técnicas mais avançadas.

As dimesões desse relatório são 6362620 linhas e 11 colunas, após subir o CSV em um dataset do pandas foi possível utilizar a função info para descobrir essas dimensões.



Este dataset possuí 11 colunas sendo elas

step - mapeia uma unidade de tempo no mundo real. Neste caso, 1 passo é 1 hora de tempo. Total de etapas 744 (simulação de 30 dias).

type- CASH-IN, CASH-OUT, DEBIT, PAYMENT e TRANSFER.

amount- valor da transação em moeda local.

nameOrig - cliente que iniciou a transação

oldbalanceOrg - saldo inicial antes da transação

newbalanceOrig - novo saldo após a transação

nameDest - cliente que é o destinatário da transação

oldbalanceDest - destinatário do saldo inicial antes da transação. Observe que não há informações para clientes que começam com M (Comerciantes).

newbalanceDest - novo destinatário do saldo após a transação. Observe que não há informações para clientes que começam com M (Comerciantes).

isFraud - São as transações feitas pelos agentes fraudulentos dentro da simulação. Neste conjunto de dados específico, o comportamento fraudulento dos agentes visa lucrar tomando controle ou contas de clientes e tentando esvaziar os fundos transferindo para outra conta e depois sacando do sistema.

isFlaggedFraud - O modelo de negócios visa controlar transferências massivas de uma conta para outra e sinaliza tentativas ilegais. É uma flag criada pelo próprio modelo.

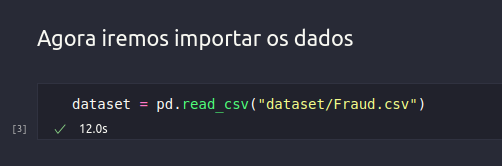
Como chegar aos resultados?

● Estudar a base para entende-la

Toda a análise exploratória, os gráficos gerados além dos algoritmos utilizados estão disponíveis no github no repositório: <https://github.com/jorgelpiva/A3_IAvsBigData>

A análise está sendo conduzida em um jupyter notebook [analise\_exploratoria.ipynb](https://github.com/jorgelpiva/A3_IAvsBigData/blob/main/analise_exploratoria.ipynb)

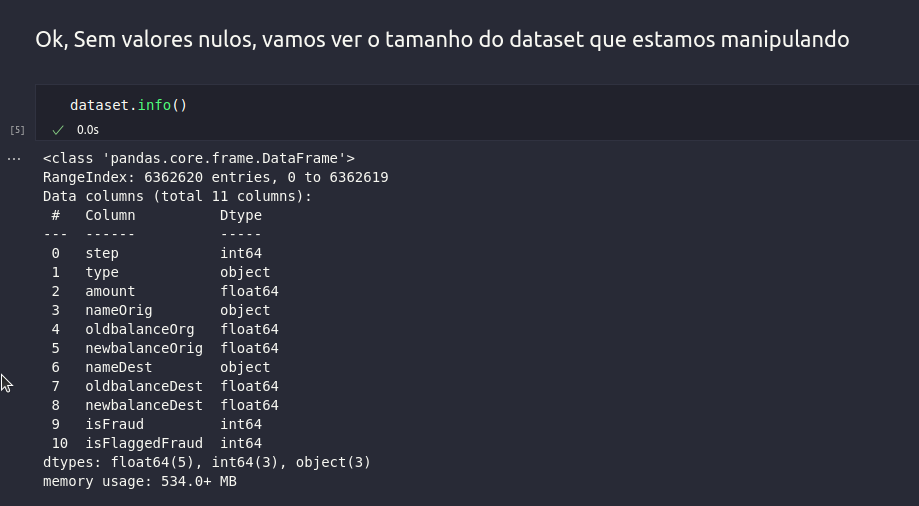
Primeiramente importamos o arquivo.



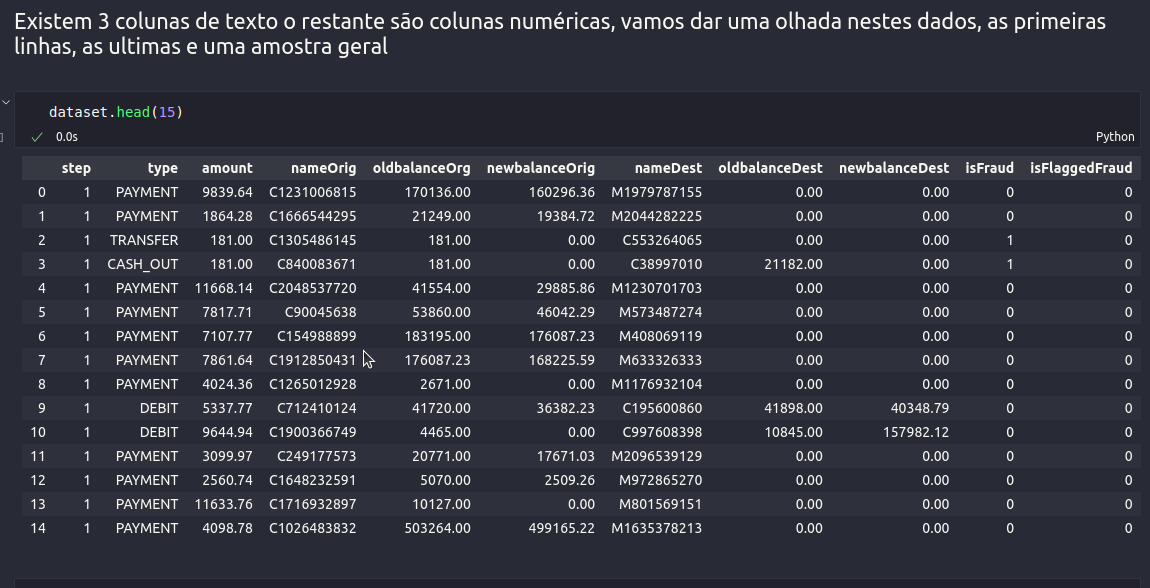
Depois verificamos se haviam valores nulos:

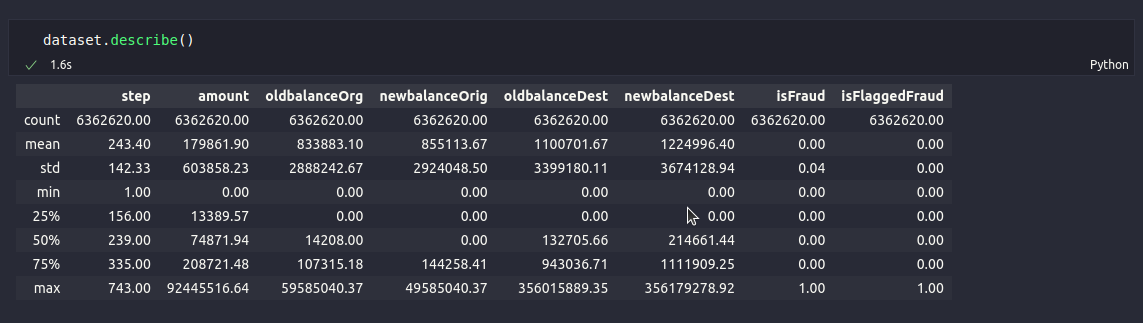


Coletamos maiores informações sobre a base de dados.

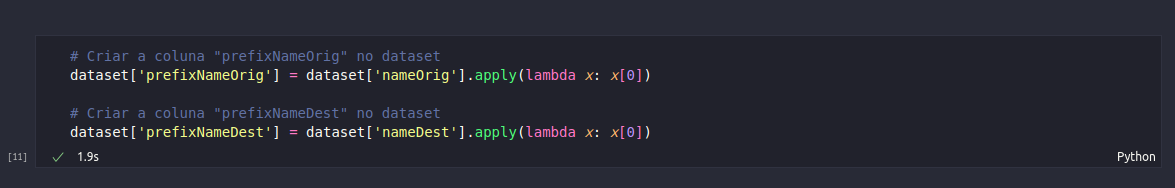


E demos uma olhada nos dados

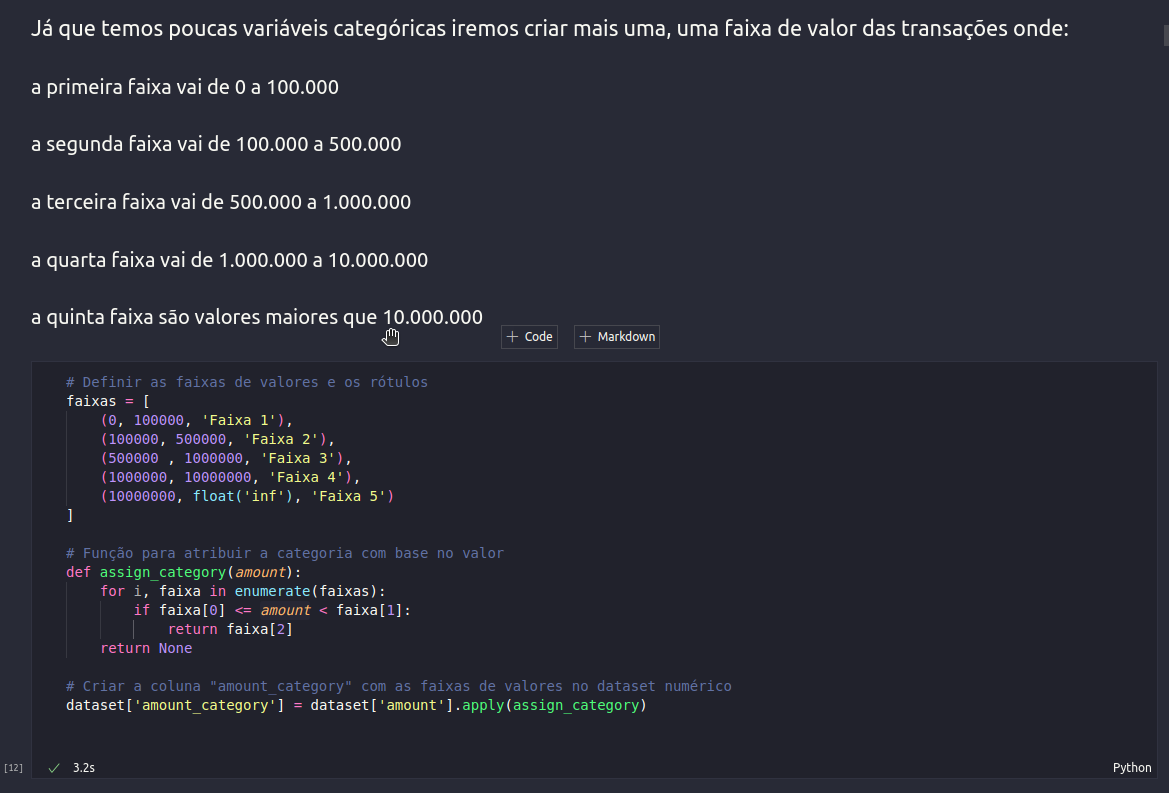
Também utilizamos a função describe para entender os valores.



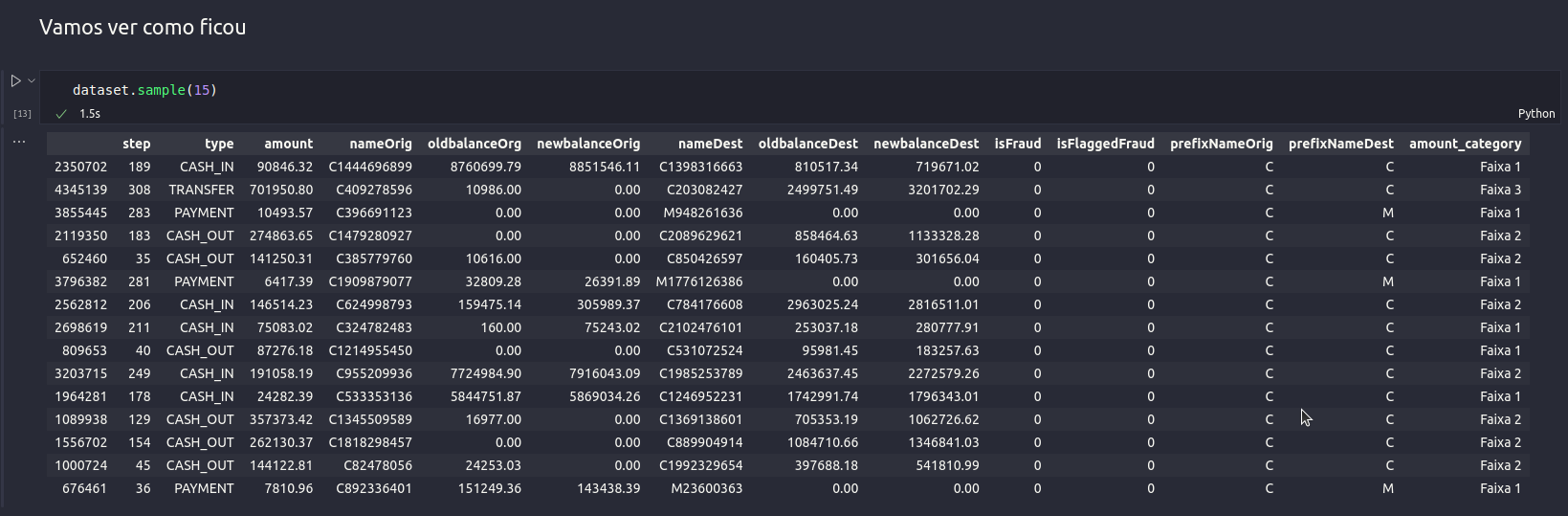
Depois criamos mais 3 colunas no dataset o prefixo do nameDest, o Prefixo do nameOri que seria a primeira letra de cada nome.



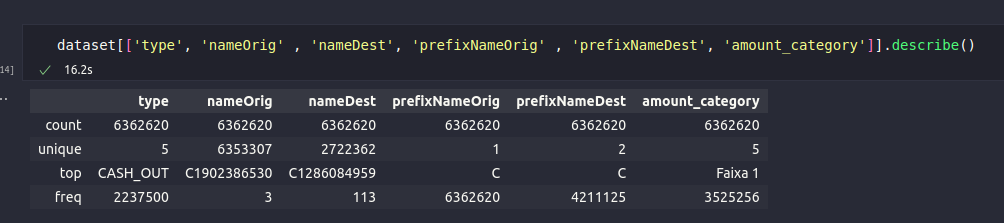
Criamos uma faixa para os valores das transações.



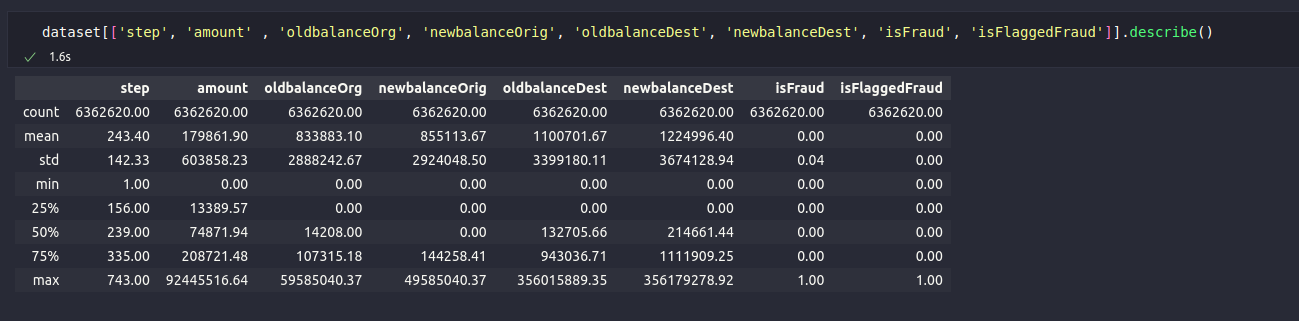
Depois demos mais uma olhadinha de como ficou:



Aí fizemos o describe só nas colunas de texto.



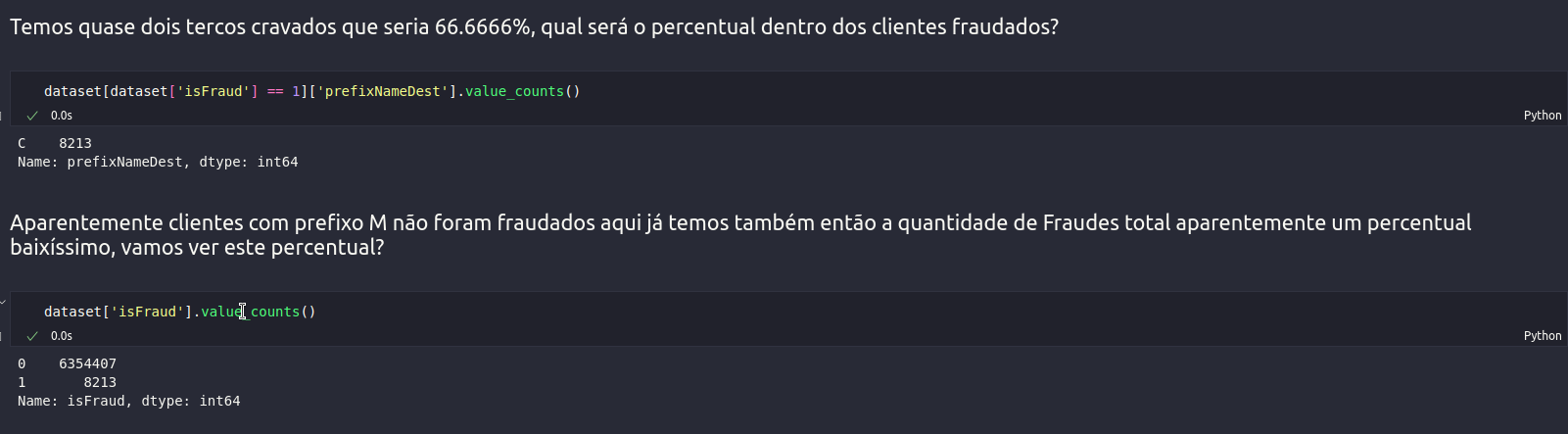
E somente nas colunas numéricas também.



Então descobrimos que existiam dois prefixos de nameDest, C e M



Descobrimos que 1/3 dos prefixos era M mas nenhum deles foi fraudado.

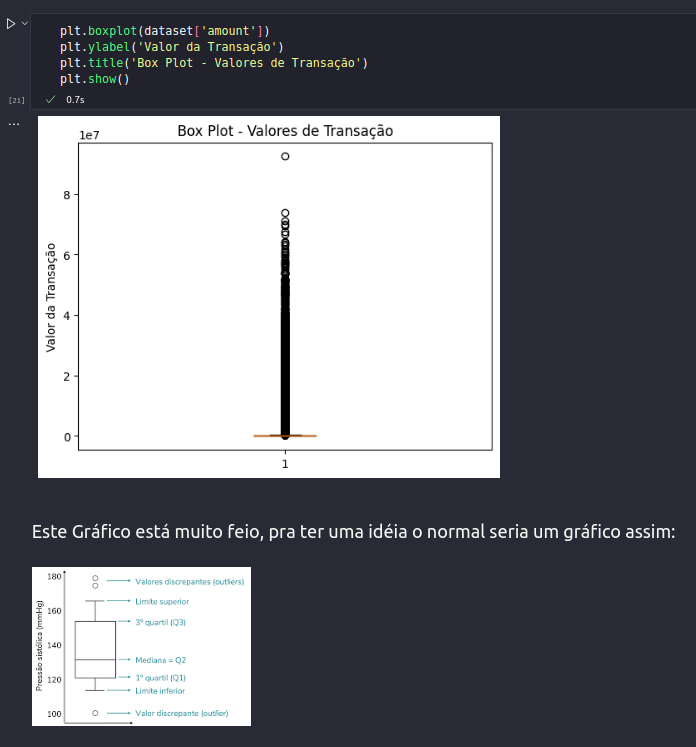


E junto com esta descoberta veio uma informação bombástica, a quantidade de fraudes era muito pequena.



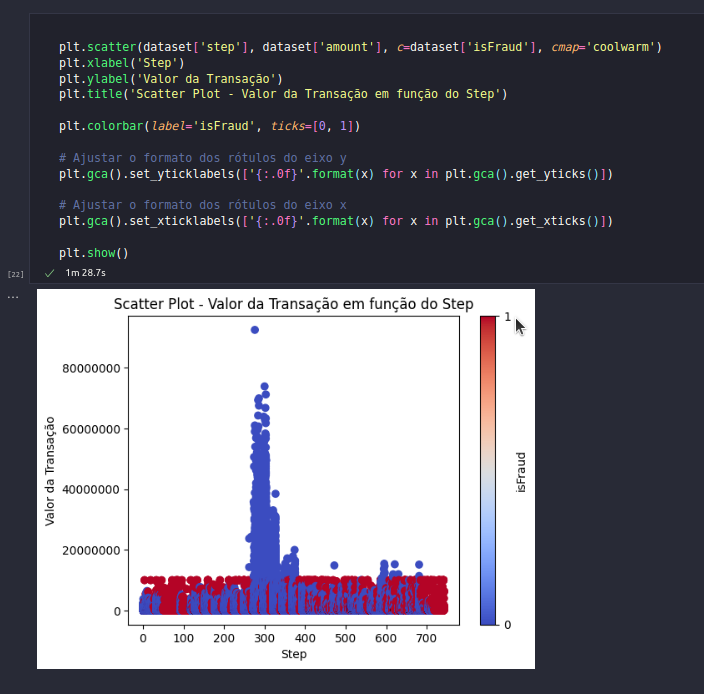
Dentro de uma base de 6 milhões de registros apenas 8 mil eram fraudulentas, o que dificulta muito para os algoritmos de IA e para nós mesmo conduzirmos análises, a partir deste momento, deveríamos trabalhar sempre olhando para o geral e para as operações fraudulentas de forma apartada.

Tentamos identificar outliers através de um box-plot



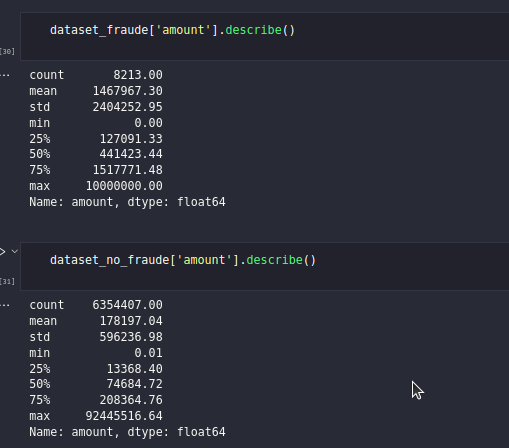
E descobrimos que tinhamos muitos outliers.

Então plotamos um scatterplot e ficou evidente que tínhamos um outlier, um momento de sazonalidade entre as transações ok e havia regularidade entre as transações fraudadas.



Aparentemente as transações fraudulentas possuem uma regularidade e ela está bem abaixo das transações de maior valor, porém, na regularidade, as trasações fraudulentas possuem uma média de valor maior do que o dataset geral.

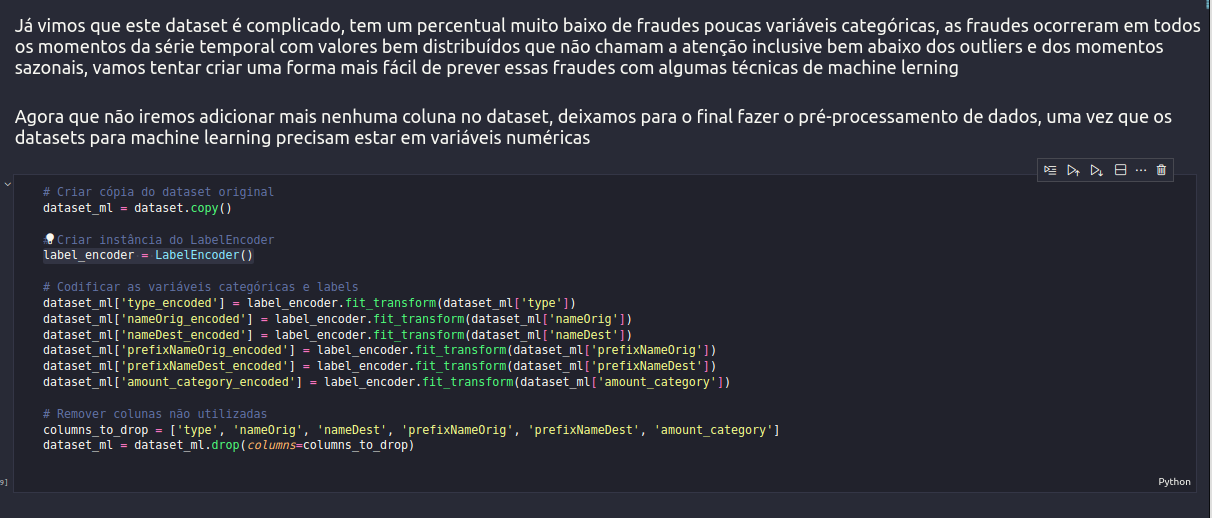
Isso fica evidente quando dividimos o dataset entre transações fraudadas e não fraudadas, quando aplicamos a função describe a média fica evidente.



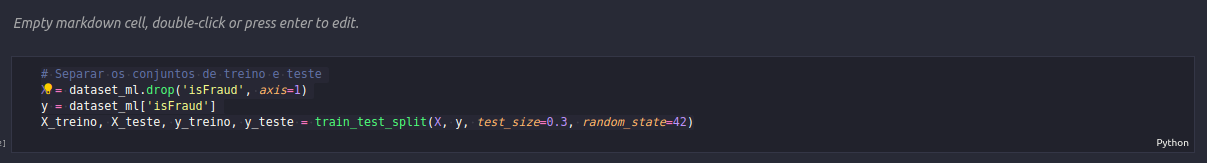
Então tendo a noção da complexidade do que acontece, começamos a aplicar os algoritmos de machine learning em nossa variável target que foi escolhida o **isFraud**.

Criamos um novo dataset para gravar os dados como tempo de excecução e os resultados.

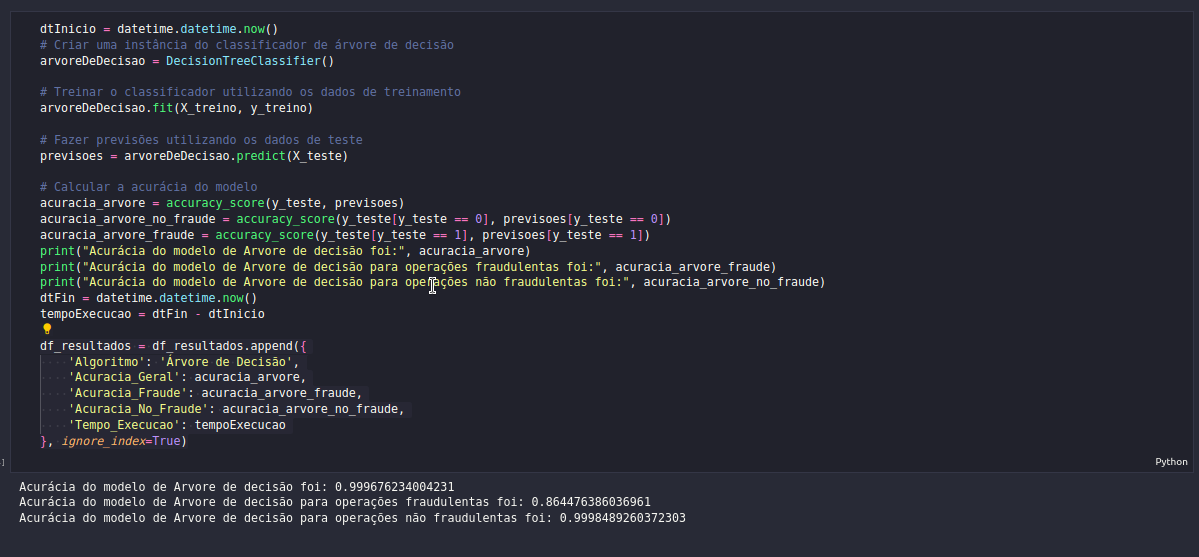
Fizemos alguns pré processamentos, primeiramente com a transformação do dataset geral para um dataset apenas com tipos numéricos com a biblioteca label encoder do sklearn



Dividimos o dataset em 70% para treino e 30% para teste com a biblioteca train\_test\_split



Então processamos os resultados com a árvore de decisão:



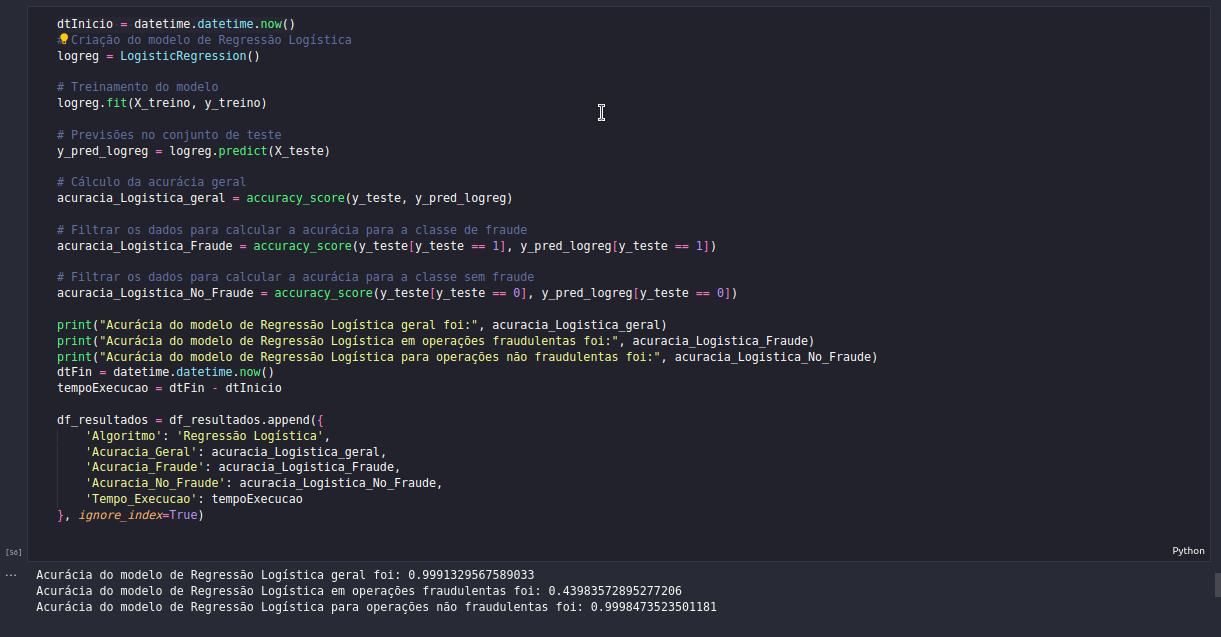
Como comentado anteriormente medimos o resultado geral, de forma a olhar os acertos apenas nas fraudes e olhando também os acertos dos não fraudados, para nossa supresa a árvore de decisão entregou um resultado geral de 99,99% e de fraudados de 86% um número fomidável para o nível de complexidade do dataset, porém, fizemos os outros modelos.

Cabe ressaltar que como temos 99,99% dos registros como transações normais, caso um algoritmo colocasse todas as ocorrências como isFraud = 0 o algoritmo acertaria 99,99% do geral e 0% dos fraudados por isso temos que dividir os acertos para mensurar a eficácia.

Antes de prosseguir, fizemos um pré processamento para padronização dos valores, do contrário, alguns algoritmos como o KNN iriam dar erro, esse pré processamento foi feito com a biblioteca StandardScaler também do sckitlearn



Então fizemos a regressão logística.

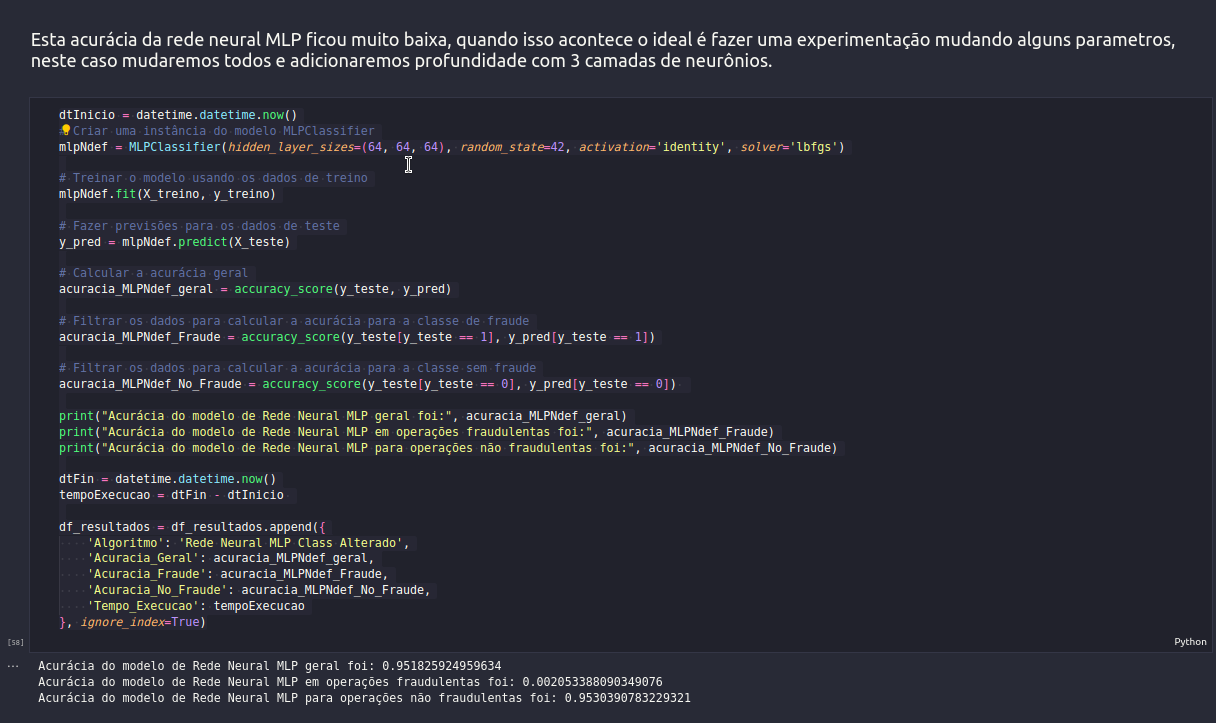


Tivemos um resultado interessante, cerca de 44% de acerto para as fraudes mas bem abaixo da árvore de decisão.

Então partimos para a rede neural MLP, e aí tivemos uma grande decepção:



Foi um resultado de menos de 4% inconformados, mudamos alguns parâmetros e rodamos de novo.



Então percebemos que precisamos estudar mais os parametros para conseguir uma melhor avaliação, como os tempos de execução estavam crescendo à medida que estávamos aumentando o número de camadas da rede neural estávamos caminhando em direção ao *DeepLearning*, decidimos passar para o próximo algoritmo o SVC .

E o SVC se saiu muito melhor que a rede neural MLP mas abaixo da regressão logística e muito abaixo da Árvore de Decisão com apenas 33% de acertos nas operações fraudulentas.

Ficamos convictos que teríamos um melhor aproveitamento com o naive bayes então seguimos para o mesmo.

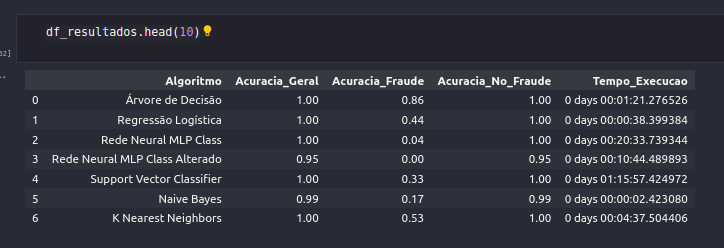


E novamente tivemos uma leve decepção, quando o Naive Bayes acertou apenas 17% das fraudes nossa útlima esperança para elevar a taxa de acerto era o KNN.

Rodamos o KNN e ele bateu todos os anteriores, menos a Árvore de Decisão chegando a quase 53% de acertos.



Confrome mencionado, criamos um dataset para computar os valores do experimento e no final da execução ele ficou da seguinte forma:

Além da primeira execução ter sido a mais assertiva ela também foi uma das execuções mais rápidas.

Conclusão

Neste caso então podemos dizer que a primeira opção era a opção mais viável, mas como saber sem testar? Tivemos que fazer diversas baterias de teste para saber que a primeira opção era a mais viável, e se tivéssemos começado pelo KNN teríamos um valor bom, mas que poderia ser melhorado. Se tivéssemos rodado a árvore de decissão por último, só então teríamos encontrado o melhor aproveitamento, e ainda rodamos o MLP duas vezes, essa é rotina de um Cientista de Dados, de um Engenheiro de IA, experimentação e teste, vale lembrar que a ciência vive de testes, pesquisas e experimentos e vale muito a pena fazer todo este caminho mesmo que seja para descobrir no final que o primeiro era o mais assertivo, sabendo que você utilizou de todo o seu conhecimento e recursos disponíveis a fim de entregar o melhor resultado, todos os testes e resultados obtidos são execução de algoritmos complexos que demandam grande poder computacional em um dataset de mais de 6 milhões de linhas, este projeto pode ser aprimorado futuramente com o estudo mais especializado dos parâmetros presentes na biblioteca do sklearn e também com o uso de novas bibliotecas como o tensor flow e o pytorch, porém, é muito satisfatório saber que treinamos 7 modelos de aprendizado de máquina e conseguimos uma acurácia de 86% em identificar fraudes, com uma especialização maior podemos trazer dados ainda mais assertivos, mas para este estudo preliminar ficamos muito satisfeitos com os resultados e interessados em testar os modelos em outros datasets talvez com mais variáveis categóricas, talvez com uma variável target com um percentual maior de positivos, nestas situações, pode ser que a rede neural MLP saia melhor que a Árvore de Decisão ou o Modelo SVC? É possível, mas só será possível descobrir se houver, experimentação, teste e metodologia científica e vontade de tirar informações úteis dos dados.