INSPER – INSTITUTO DE ENSINO E PESQUISA

Ciência dos Dados

PROF. Fabio Ayres

Aplicação de regressão nos jogos da steam

Lucas Fukada

MARCELO MIGUEL

Eiki Yamashiro

Beatriz muniz

**SUMÁRIO**

SÃO PAULO, 19 de novembro de 2019

**1 INTRODUÇÃO/OBJETIVOS.......................................... 3**

**2 DATASET........................................................................... 3**

**3 PROCEDIMENTOS EXPERIMENTAIS........................ 4**

**4 RESULTADOS E DISCUSSÕES....................................... 6**

**5 CONCLUSÕES................................................................... 7**

**1 INTRODUÇÃO/OBJETIVOS**

O projeto consiste na utilização de 3 métodos de regressão (Regressão Linear, RandomForestRegression e Regressão Polinomial) nos jogos da plataforma Steam. Dessa forma, para os 3 tipos de regressão, o objetivo é de prever qual seria a avaliação de um jogo, com base no preço em seu preço, a quantidade de categorias que ele se encaixa, data de lançamento, média de tempo jogado e a mediana de tempo jogado.

**2 DATASET**

O [DataSet](https://www.kaggle.com/nikdavis/steam-store-games/data) foi retirado do site [kaggle](http://www.kaggle.com), contendo as seguintes categorias : código de identificação, nome, título do jogo, data de lançamento, linguagem do jogo, desenvolvedor, editora, plataforma, idade mínima para poder jogar, categoria, gênero, tags (atribuídos pelos usuários), conquistas, votos (ratings) positivos, votos negativos, média de tempo jogado, mediana de tempo jogado, quantidade de usuários, preço.

**3 PROCEDIMENTOS EXPERIMENTAIS**

Os procedimentos experimentais serão divididos em alguns tópicos com o intuito de uma melhor organização dos métodos utilizados para a obtenção de resultados e os acontecimentos ocorridos no decorrer do projeto.

3.1 LIMPEZA DO DATASET

Inicialmente, para analisar apenas os jogos do banco de dados, foi retirado todos os Apps, com base em seus gêneros, como por exemplo foi retirado os apps com o gênero “Web Publishing”, dessa forma, o DataSet passou a ter apenas jogos. Além disso, foi excluído os jogos que possuíam mediana ou média de horas de jogo igual a zero, pelo fato de não ser explicado o porquê desses jogos terem avaliações, mas não terem horas de jogos.

3.2 ANÁLISE DAS COLUNAS E DECISÃO DO OBJETIVO DO PROJETO

Após a limpeza do DataSet, foi feito a análise dos dados, tentando encontrar alguma correlação entre as colunas, porém, o grupo não conseguiu encontrar grandes correlações entre as variáveis.

Dessa forma, após a análise das correlações terem demonstrado haver pouca correlação, o grupo decidiu explorar as avaliações dos jogos, transformando as avaliações positivas e negativas totais em porcentagem, para evitar que jogos mais antigos tenham um número maior de avaliações. Além disso, as avaliações foram classificadas em muito bom (maior que *0.75*), bom (maior que *0.5*) e ruim (menor que *0.5*).

Sendo assim, foi analisado a correlação entre a porcentagem das avaliações positivas e as outras variáveis, focando no preço, número de categorias, variáveis de tempo e data de lançamento, porém, como visto anteriormente, nenhuma tinha variável tinha uma grande correlação, sendo a maior, de preço, igual a *0,18*. Dessa forma, a análise tinha o intuito de ressaltar a baixa correlação e uma possível justificativa para os resultados das análises que serão feitas a seguir.

3.3 ANÁLISE DAS AVALIAÇÕES UTILIZANDO MÉTODO DE CLASSIFICAÇÃO

De início, o grupo decidiu utilizar o método de classificação RandomForestClassifier da biblioteca sklearn para prever se o jogo seria “bom”, “muito bom”, ou “ruim”, com base no preço, número de categorias, data de lançamento e pela média e mediana de horas jogadas.

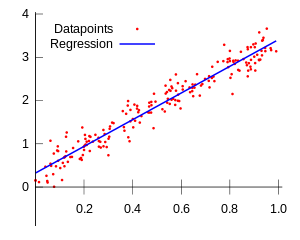
Porém, pelo fato das variáveis terem pouca correlação com as avaliações, a acurácia do modelo era sempre igual ou pior ao número de avaliações (bom, muito bom, ou ruim).

Dessa forma, o grupo, não satisfeito com os resultados, decidiu utilizar os métodos de regressões (Linear, Random Forest Regressor e Regressão Polinomial) a fim de encontrar melhores resultados (sendo, nesse caso, o *R²* ).

3.4.1 REGRESSÃO LINEAR

A regressão linear simples é um estimador, ou seja, um modelo que permite com que se faça previsões de uma população de dados com base na própria população de dados. A métrica escolhida pelo grupo para avaliar o desempenho da regressão foi o R².

A regressão linear basicamente é uma função de primeiro grau que permite que se faça a estimativa:

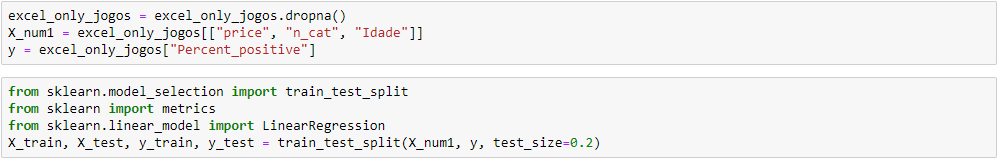


*- Figura 1: Gráfico Regressão Linear Simples –*

No gráfico acima, os pontos vermelhos seriam a população de dados e a reta azul seria a função que representa a regressão linear simples.

Para calcular o coeficiente de determinação (*R²*), usou-se a função r2\_score() da biblioteca sklearn. Esse coeficiente pode variar de 1 a -∞, sendo que o melhor resultado possível seria o 1.

A regressão linear será feita com o auxílio da biblioteca sklearn e outras, como: numppy, stats, seaborn e math. Através da função train\_test\_split() do sklearn, dividiu-se o Dataframe em dois, um para treino e outro para teste:



*- Figura 2: Importando as bibliotecas e definindo algumas variáveis –*

Escolheu-se as séries “price”, “n\_cat” e “Idade” do DataFrame, que representam respectivamente: preço do jogo, número de categorias e idade. Essas 3 séries são as variáveis de análise, já a série “Percent\_positive”, que representa a porcentagem positiva dos ratings, é a variável-alvo. Após esse procedimento, usou-se a função .predict() para se fazer a estimativa:

C:\Users\eikis\Desktop\3664691d-e337-4285-96cc-965f5a1187c5.jpg

*- Figura 3: Código da estimativa –*

Dessa forma, o resultado foi que, o *R²* calculado foi de: *0,06*, utilizando as variáveis preço, número de categorias e data de lançamento; *0,6*, incluindo as variáveis de tempo; *0,02*, removendo a variável de preço; e 0,01, removendo a variável de categorias e incluindo novamente a variável de preço

3.4.2 RANDOM FOREST REGRESSOR

A utilização do RANDOM FOREST REGRESSOR foi importante pelo fato de que esse tipo de análise utiliza de várias arvores de regressão treinadas com conjuntos aleatórios de, após finalizado as árvores de regressão

Dessa forma, o resultado foi que, o *R²* calculado foi de: *0,08*, utilizando as variáveis preço, número de categorias e data de lançamento; *0,02*, incluindo as variáveis de tempo; -*0,01*, removendo a variável de preço. Sendo assim, percebe-se que, apesar de ter um *R²* melhor que o da regressão linear em alguns aspectos, ainda pode ser considerado baixo.

3.4.3 REGRESSÃO POLINOMIAL

.

**4 RESULTADOS E DISCUSSÕES**

.

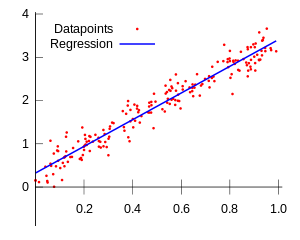
**5 CONCLUSÕES**

Apesar da ampla quantidade de dados apresentados, a baixa correlação entre as variáveis e o alvo e a acurácia impediram a criação de um método efetivo e apurado para que se pudesse prever a avaliação de um jogo com base em seus atributos.

A verdade é que os video-games são muito subjetivos, e jogos do mesmo genêro e com as mesmas tags podem possuir mecânicas e dinâmicas distintas que podem ser a diferença entre uma nota positiva ou negativa. E enquanto características como gênero, preço, tags ou data de lançamento podem influenciar as avaliações desses jogos, elas influenciam muito pouco, sendo o maior fator a jogabilidade e o conteúdo apresentado.

**Regressão Linear Simples**

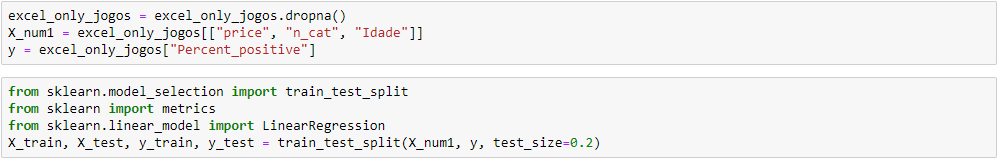
1. **Definição:**
   1. A regressão linear simples é um estimador, ou seja, um modelo que permite com que se faça previsões de uma população de dados com base na própria população de dados. A métrica escolhida pelo grupo para avaliar o desempenho da regressão foi o R².
   2. A regressão linear basicamente é uma função de primeiro grau que permite que se faça a estimativa:



*- Figura 1: Gráfico Regressão Linear Simples –*

* 1. No gráfico acima, os pontos vermelhos seriam a população de dados e a reta azul seria a função que representa a regressão linear simples.

1. **Cálculo do R²:** 
   1. Para calcular o coeficiente de determinação (R²), usou-se a função r2\_score() da biblioteca sklearn. Esse coeficiente pode variar de 1 a -∞, sendo que o melhor resultado possível seria o 1.
2. **Implementação no Python:**
   1. A regressão linear será feita com o auxilio da biblioteca sklearn e outras, como: numppy, stats, seaborn e math. Através da função train\_test\_split() do sklearn, dividiu-se o Dataframe em dois, um para treino e outro para teste:



*- Figura 2: Importando as bibliotecas e definindo algumas variáveis –*

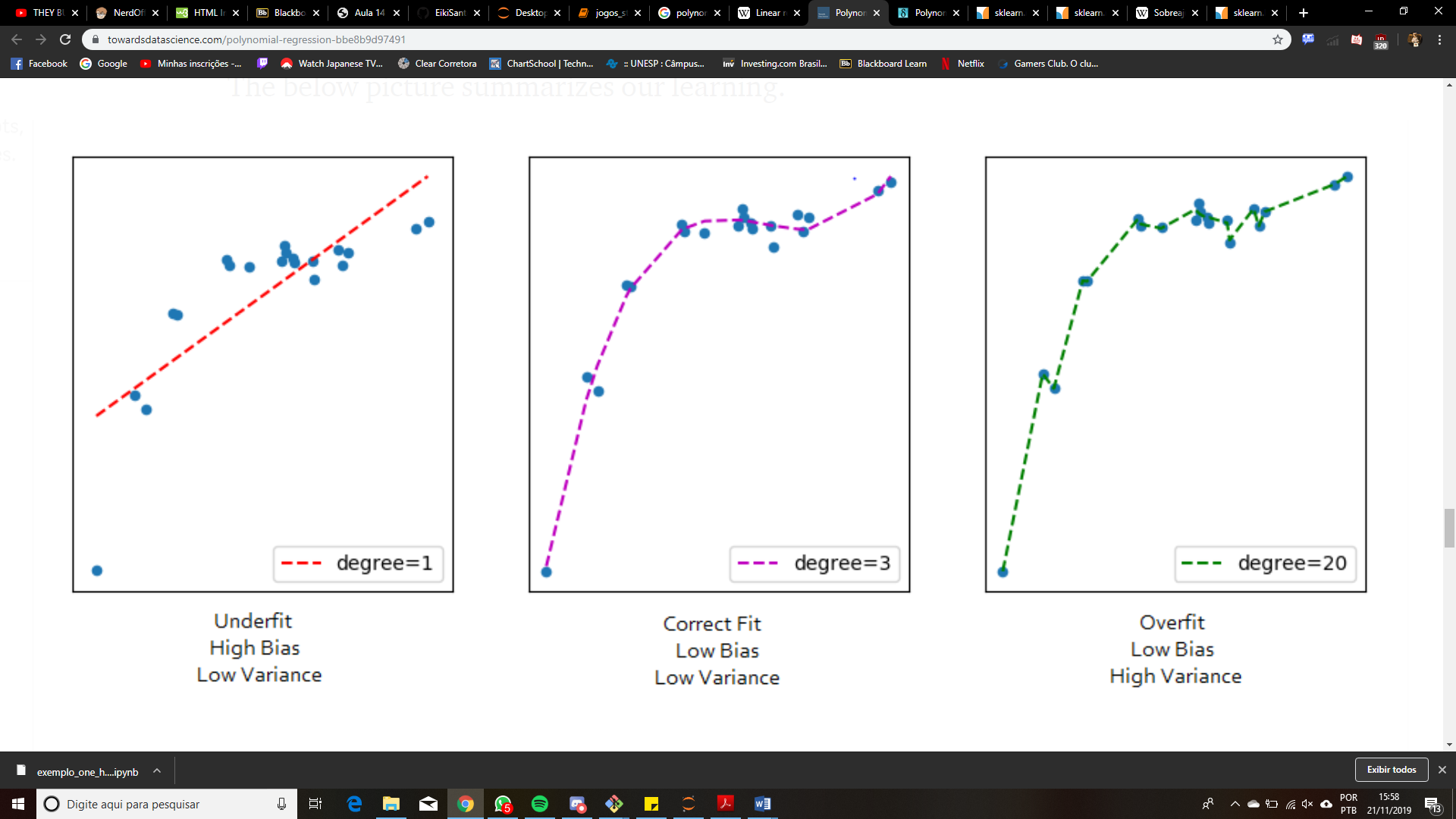
* 1. Escolheu-se as séries “price”, “n\_cat” e “Idade” do DataFrame, que representam respectivamente: preço do jogo, número de categorias e idade. Essas 3 séries são as variáveis de análise, já a série “Percent\_positive”, que representa a porcentagem positiva dos ratings, é a variável-alvo. Após esse procedimento, usou-se a função .predict() para se fazer a estimativa:

C:\Users\eikis\Desktop\3664691d-e337-4285-96cc-965f5a1187c5.jpg

* 1. Na última célula, a função .score já foi utilizada, portanto o R² do nosso modelo foi de 0,06. Ou seja, não foi possível obter um modelo confiável através da regressão linear com base nas séries “price”, “n\_cat”, e “Idade” para a série “Percent\_positive”.

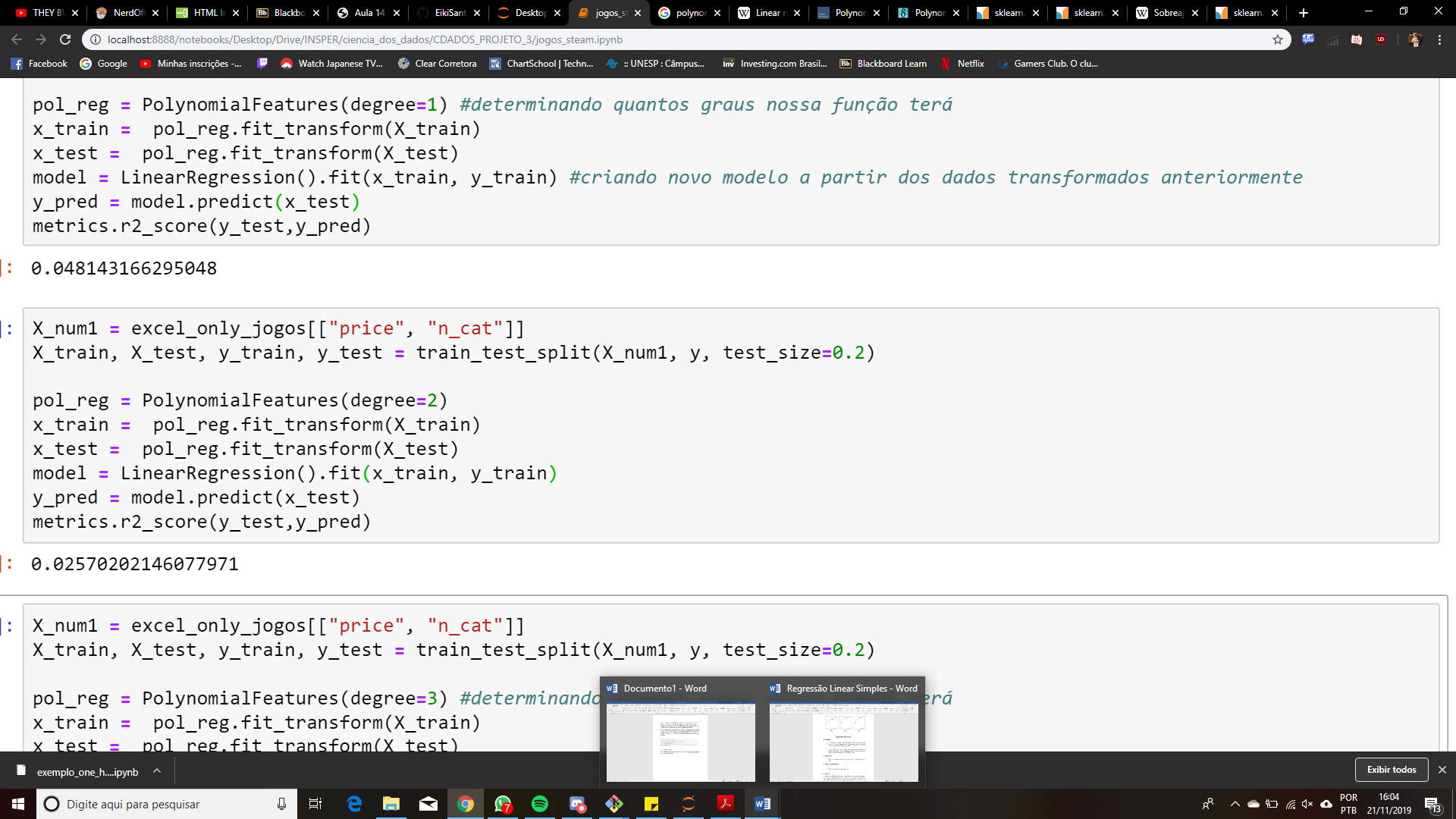
1. **Conclusão:**

Através das variáveis de análise não foi possível obter um modelo confiável, visto que o R² ficou muito longe do 1, e muito próximo do 0. Entretanto, se mudarmos o grau da função da regressão linear é possível que se obtenha um R² mais próximo de 1. Como podemos observar no exemplo abaixo, para diferentes graus podemos obter um modelo mais confiável:



**Regressão Polinomial**

1. **Definição:**
   1. A regressão linear simples é um outro estimador, mas suas previsões são baseadas em uma função que pode ser de maior grau, ou seja, o modelo pode ter a aparência de uma curva. A métrica para avaliar o desempenho da regressão é o R².
   2. A regressão pode ter várias formas. Isso é exemplificado pela figura apresentada anteriormente, onde a primeira reta mostra um modelo de grau 1, a segunda uma de grau 3 e a terceira de 20. O modelo segue a seguinte fórmula:
2. **Cálculo do R²:** 
   1. Para este modelo também foi utilizado a função r2\_score() para obter a o R².
3. **Implementação no Python:**
   1. Para a polinomial, os códigos extras são:



A primeira linha escolhe o grau do modelo, a segunda e terceira transformam os dados para o modelo polinomial e por fim a quarta gera os coeficientes de ajuste da curva.

1. **Conclusão:**

Como mostrado pelos resultados para diferentes graus, não foi possível obter valores satisfatórios no modelo polinomial, ou seja, os R² apresentados estão longe de 1, causado pela falta de correlação entre as variáveis de análise.