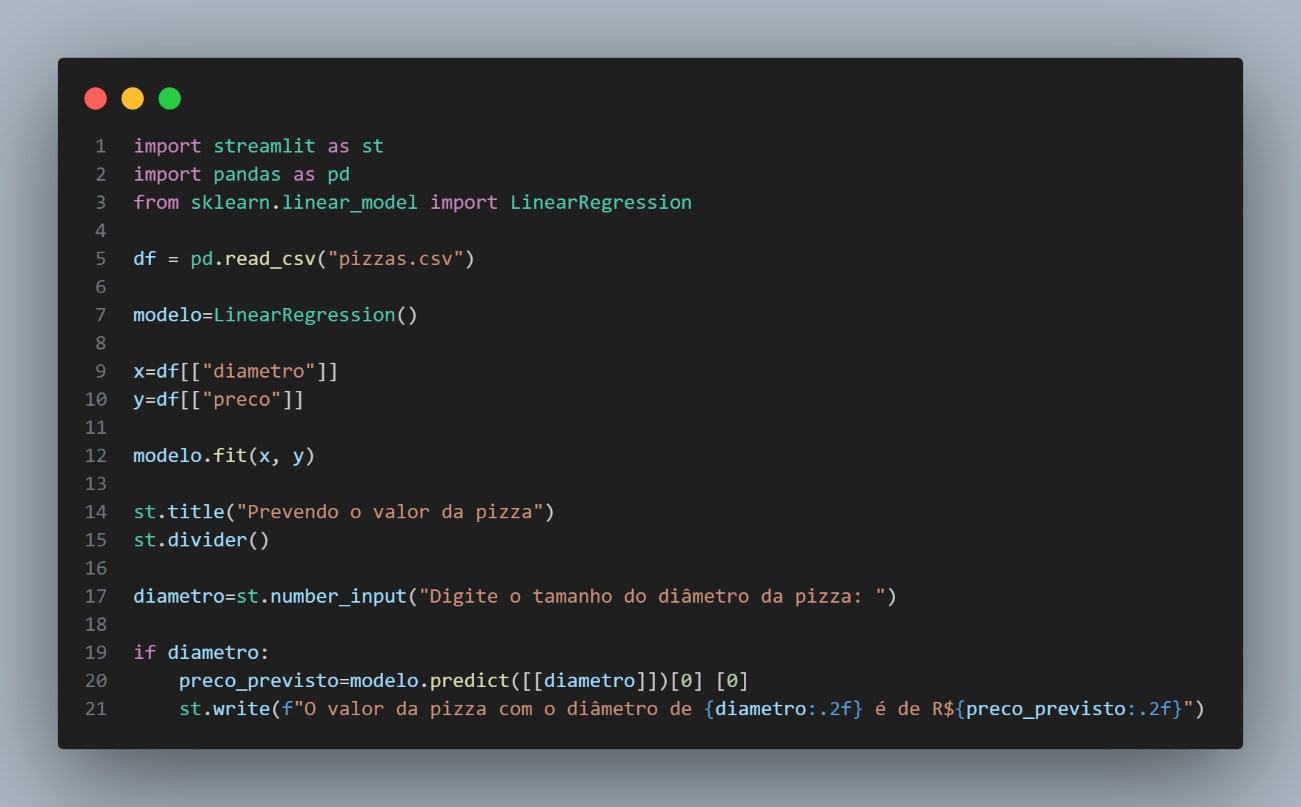
Esse código em Python cria um modelo de IA que usa lógica matemática para relacionar o tamanho da pizza (diâmetro) com o seu preço. Ele faz isso usando regressão linear, que é uma técnica de machine learning para encontrar a melhor reta que descreve a relação entre uma variável independente (diâmetro) e uma variável dependente (preço). print do código completo:

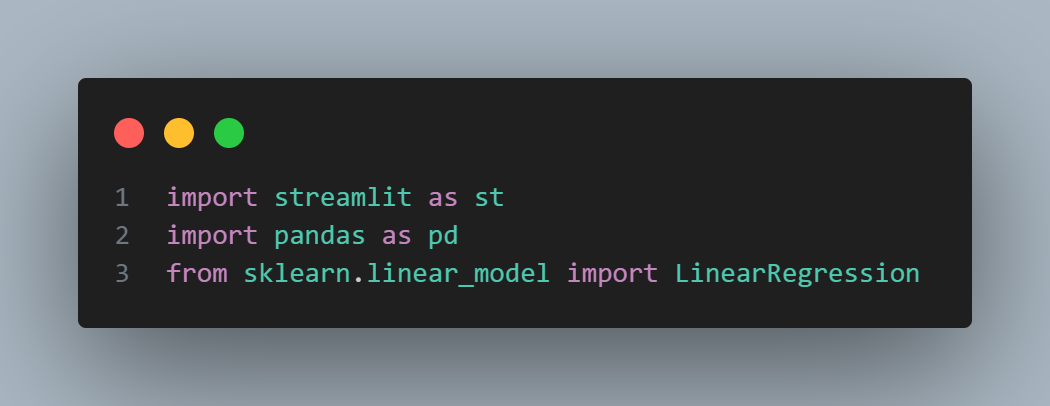


No início, importamos as bibliotecas que serão usadas no código:

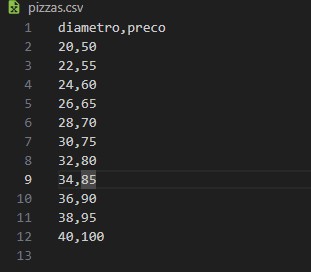
streamlit: responsável por criar uma interface web interativa para a IA

pandas: serve para analisar, limpar, organizar e manipular dados. no código ela serve para ler o arquivo de texto pizzas.csv, que a IA utiliza para calcular o preço.

scikit-learn: biblioteca focada em machine learning. importamos a Regressão Linear, que é um modelo matemático que tenta encontrar uma relação linear entre uma variável independente (entrada) e uma variável dependente (saída).



O comando pd.read\_csv lê o arquivo pizzas.csv e cria um DataFrame chamado df, que é uma tabela na memória. Este arquivo contem os dados históricos com duas colunas: diametro (característica de entrada) e preco (valor que queremos prever).

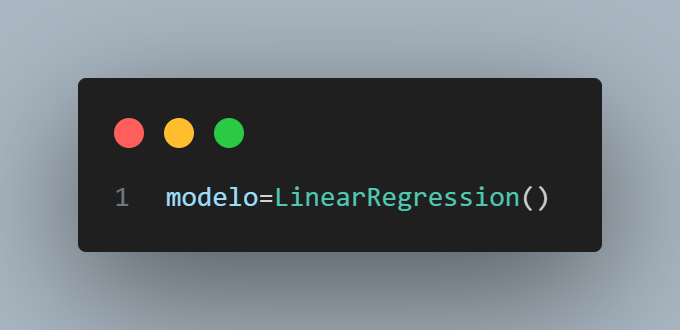


Após isso, vamos criar o modelo da IA usando modelo = LinearRegression()

Ele cria uma instância do modelo de regressão linear do scikit-learn. Este modelo irá aprender uma equação do tipo:

preco = β0 + β1 \* diametro

onde β0 é o valor inicial e β1 é a inclinação da reta. O modelo ajusta esses valores usando o método dos mínimos quadrados para minimizar o erro entre os preços reais e os preços previstos.



Também precisamos separar a features (X) e o alvo (y), o que significa dividir os dados em duas partes:  
X (features) são as variáveis de entrada, usadas para prever algo. No caso, o diâmetro da pizza.  
y (alvo) é a variável de saída, o valor que queremos prever. No caso, o preço da pizza.

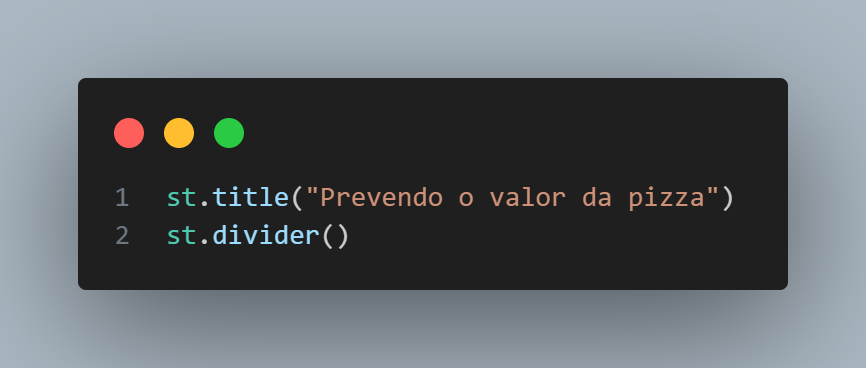


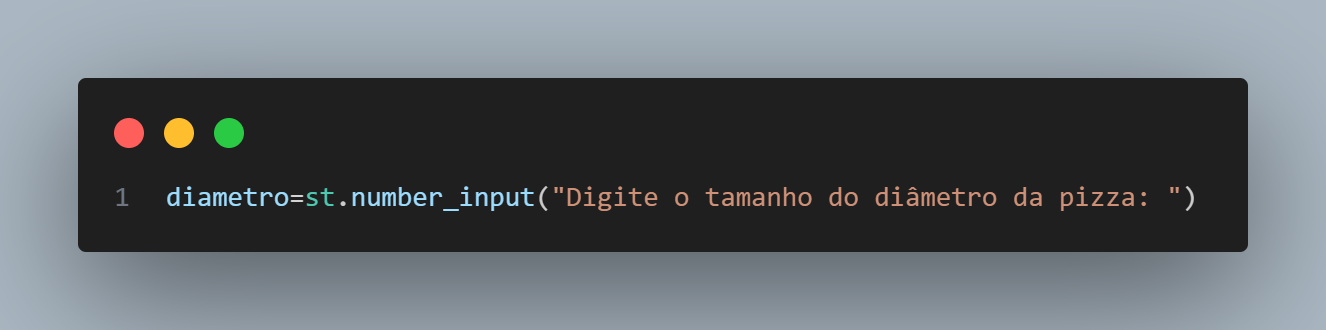
Para finalizar a criação do modelo, treinamos ele usando modelo.fit(x, y)

Aqui o modelo aprende os valores de β0 e β1 a partir dos dados carregados no DataFrame.



Após terminar o modelo de IA, temos que criar a interface onde ela vai funcionar. Vamos usar st.title("Prevendo o valor da pizza") para criar um título e st.divider() para criar uma divisória



Além disso, precisamos criar o local onde o usuário entrará com os dados. diametro = st.number\_input("Digite o tamanho do diâmetro da pizza: ") cria um campo numérico para que o usuário insira o diâmetro da pizza que deseja prever o preço.  


A lógica para prever o resultado ocorre após o modelo já estar treinado com os dados fornecidos. Ela funciona assim:

O usuário informa um valor para o diâmetro da pizza no campo numérico criado por st.number\_input.

O código verifica se algum valor foi realmente fornecido:

if diametro:

Caso um valor válido seja inserido, o modelo calcula o preço previsto:

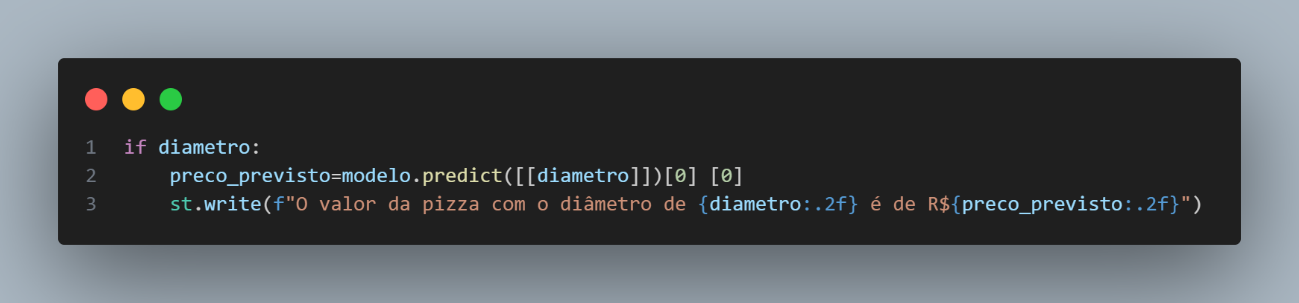
preco\_previsto = modelo.predict([[diametro]])[0][0]

modelo.predict faz a previsão usando a equação aprendida na fase de treino.

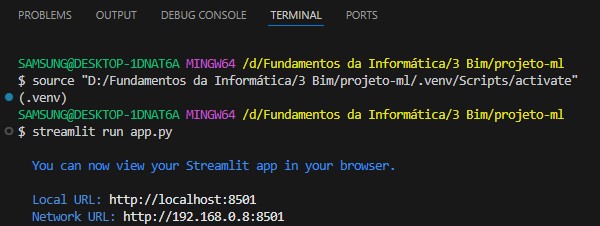
O valor [[diametro]] é passado dentro de duas listas para que o formato seja bidimensional (1 amostra, 1 característica), que é o formato esperado pelo scikit-learn.

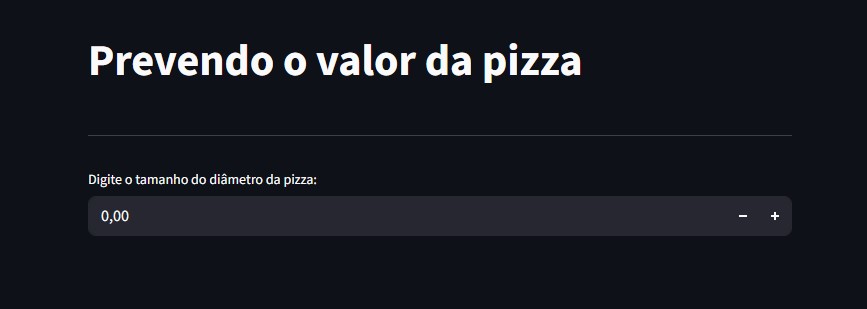
O resultado vem como um array 2D, por exemplo [[56.2]].

Por isso, é usado [0][0] para pegar apenas o valor numérico 56.2.



Concluindo o código, executamos ele e no terminal escrevemos streamlit run app.py para criar a interface web no navegador



O terminal vai abrir automaticamente no navegador a seguinte página:

Ao colocarmos qualquer valor, a IA vai calcular o valor da pizza automaticamente, exibindo o resultado da seguinte forma:

