

```
In [1]: #Importieren der erforderlichen Python-Bibliotheken und des Datensatzes

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

data = pd.read_csv("https://raw.githubusercontent.com/amankharwal/Website-data/master/advertising.csv")
print(data.head())

   TV  Radio  Newspaper  Sales
0  230.1   37.8     69.2   22.1
1   44.5   39.3     45.1   10.4
2   17.2   45.9     69.3   12.0
3  151.5   41.3     58.5   16.5
4  180.8   10.8     58.4   17.9
```

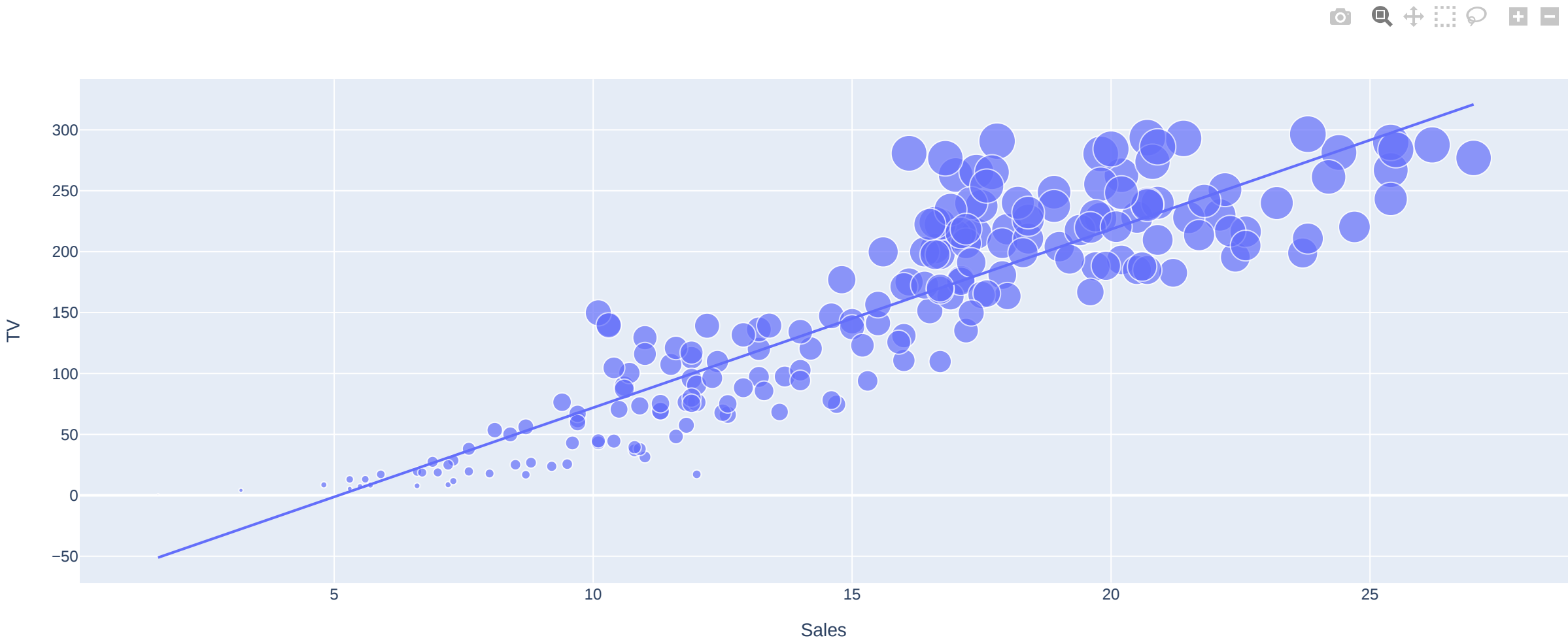
```
In [2]: #Schauen wir an, ob dieser Datensatz Nullwerte enthält oder nicht

print(data.isnull().sum())

TV          0
Radio       0
Newspaper   0
Sales       0
dtype: int64
```

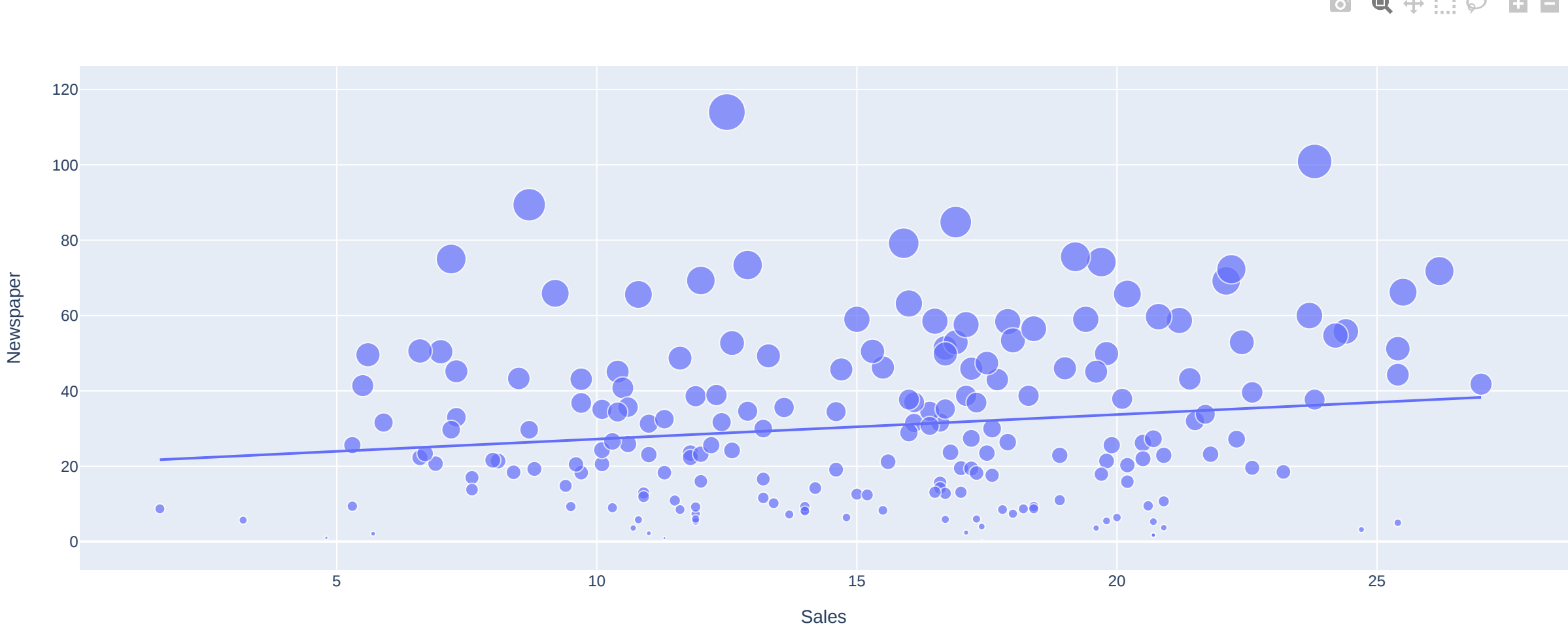
```
In [5]: #die Beziehung zwischen dem Betrag, der für Werbung im Fernsehen ausgegeben wird, und den verkauften Einheiten visualisieren

import plotly.express as px
import plotly.graph_objects as go
figure = px.scatter(data_frame = data, x="Sales", y="TV", size="TV", trendline="ols")
figure.show()
```



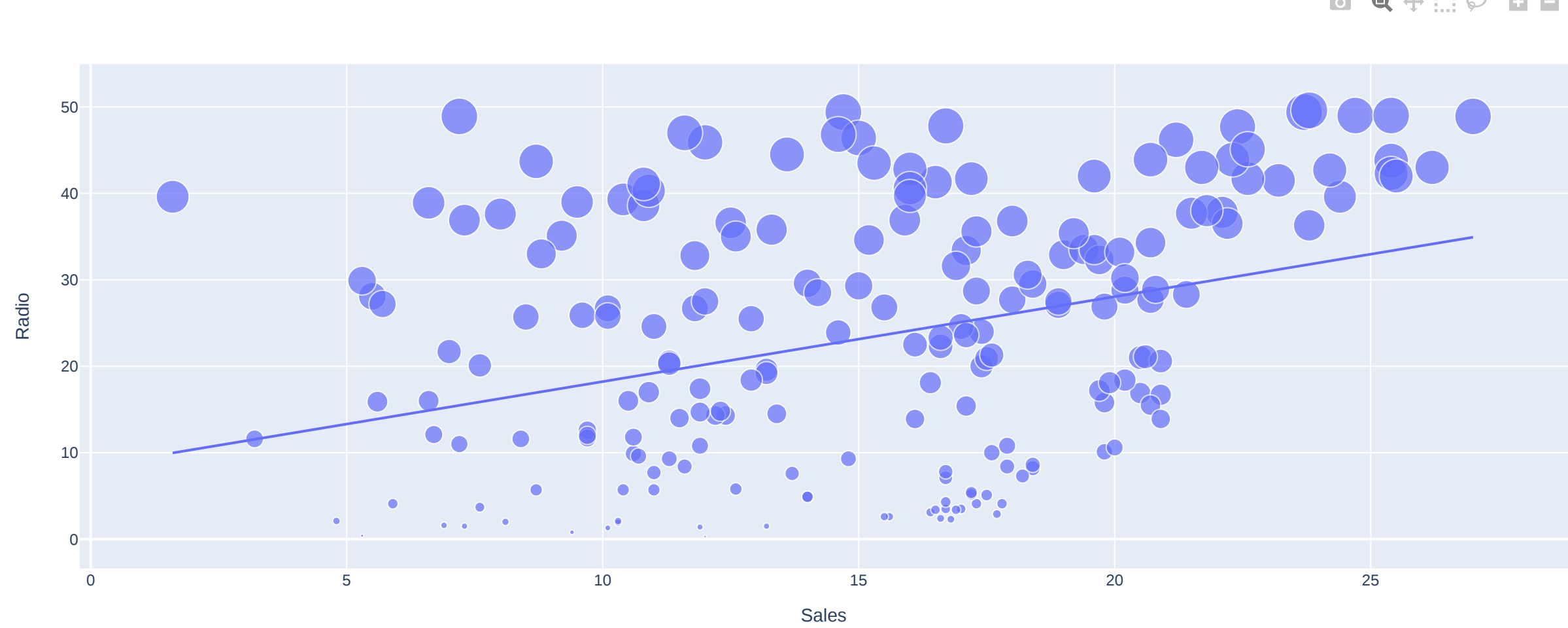
```
In [6]: #Visualisieren das Verhältnis zwischen den Werbeausgaben für Zeitungen und den verkauften Einheiten

figure = px.scatter(data_frame = data, x="Sales",
                    y="Newspaper", size="Newspaper", trendline="ols")
figure.show()
```



```
In [7]: #Visualisieren das Verhältnis zwischen den Ausgaben für Werbung im Radio und den verkauften Einheiten:

figure = px.scatter(data_frame = data, x="Sales",
                    y="Radio", size="Radio", trendline="ols")
figure.show()
```



```
In [8]: #die Korrelation aller Spalten mit der Verkaufsspalte

correlation = data.corr()
print(correlation["Sales"].sort_values(ascending=False))

Sales      1.000000
TV         0.901208
Radio      0.349631
Newspaper  0.157960
Name: Sales, dtype: float64
```

```
In [21]: #Trainieren ein maschinelles Lernmodell, um die zukünftigen Verkäufe eines Produkts vorherzusagen
# Assuming 'data' is a Pandas DataFrame

x = np.array(data.drop(["Sales"], axis=1))
y = np.array(data["Sales"])

# Splitting the dataset into the Training set and Test set
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size=0.2, random_state=42)
```

```
In [22]: model = LinearRegression()
model.fit(xtrain, ytrain)
print(model.score(xtest, ytest))

0.9059011844150825
```

```
In [26]: # Werte in das Modell entsprechend den Funktionen, mit denen wir es trainiert haben, und prognostizieren, wie viele Einheiten des Produktsbasierend auf dem für seine We
#features = [[TV, Radio, Newspaper]]

features = np.array([[230.1, 37.8, 69.2]])
print(model.predict(features))

[21.37254028]
```

In []:

In []: