In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
import matplotlib
```

In [2]:

```
data = pd.read_csv("C:\\Users\\AMIT\\Downloads\\50_Startups (2).csv")
data
```

Out[2]:

	R&D Spend	Administration	Marketing Spend	Profit
0	165349.20	136897.80	471784.10	192261.83
1	162597.70	151377.59	443898.53	191792.06
2	153441.51	101145.55	407934.54	191050.39
3	144372.41	118671.85	383199.62	182901.99
4	142107.34	91391.77	366168.42	166187.94
5	131876.90	99814.71	362861.36	156991.12
6	134615.46	147198.87	127716.82	156122.51
7	130298.13	145530.06	323876.68	155752.60
8	120542.52	148718.95	311613.29	152211.77
9	123334.88	108679.17	304981.62	149759.96
10	101913.08	110594.11	229160.95	146121.95
11	100671.96	91790.61	249744.55	144259.40
12	93863.75	127320.38	249839.44	141585.52
13	91992.39	135495.07	252664.93	134307.35
14	119943.24	156547.42	256512.92	132602.65
15	114523.61	122616.84	261776.23	129917.04
16	78013.11	121597.55	264346.06	126992.93
17	94657.16	145077.58	282574.31	125370.37
18	91749.16	114175.79	294919.57	124266.90
19	86419.70	153514.11	0.00	122776.86
20	76253.86	113867.30	298664.47	118474.03
21	78389.47	153773.43	299737.29	111313.02
22	73994.56	122782.75	303319.26	110352.25
23	67532.53	105751.03	304768.73	108733.99
24	77044.01	99281.34	140574.81	108552.04
25	64664.71	139553.16	137962.62	107404.34
26	75328.87	144135.98	134050.07	105733.54
27	72107.60	127864.55	353183.81	105008.31
28	66051.52	182645.56	118148.20	103282.38

	R&D Spend	Administration	Marketing Spend	Profit
29	65605.48	153032.06	107138.38	101004.64
30	61994.48	115641.28	91131.24	99937.59
31	61136.38	152701.92	88218.23	97483.56
32	63408.86	129219.61	46085.25	97427.84
33	55493.95	103057.49	214634.81	96778.92
34	46426.07	157693.92	210797.67	96712.80
35	46014.02	85047.44	205517.64	96479.51
36	28663.76	127056.21	201126.82	90708.19
37	44069.95	51283.14	197029.42	89949.14
38	20229.59	65947.93	185265.10	81229.06
39	38558.51	82982.09	174999.30	81005.76
40	28754.33	118546.05	172795.67	78239.91
41	27892.92	84710.77	164470.71	77798.83
42	23640.93	96189.63	148001.11	71498.49
43	15505.73	127382.30	35534.17	69758.98
44	22177.74	154806.14	28334.72	65200.33
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75
47	0.00	135426.92	0.00	42559.73
48	542.05	51743.15	0.00	35673.41
49	0.00	116983.80	45173.06	14681.40

In [3]:

```
data.head() # To get the first 5 rows
```

Out[3]:

	R&D Spend	Administration	Marketing Spend	Profit
0	165349.20	136897.80	471784.10	192261.83
1	162597.70	151377.59	443898.53	191792.06
2	153441.51	101145.55	407934.54	191050.39
3	144372.41	118671.85	383199.62	182901.99
4	142107.34	91391.77	366168.42	166187.94

In [4]:

```
data.shape # shows number of rows and columns in a dataset
```

Out[4]:

```
(50, 4)
```

In [6]:

```
data
```

Out[6]:

	R&D Spend	Administration	Marketing Spend	Profit
0	165349.2000	136897.80	471784.1000	192261.83
1	162597.7000	151377.59	443898.5300	191792.06
2	153441.5100	101145.55	407934.5400	191050.39
3	144372.4100	118671.85	383199.6200	182901.99
4	142107.3400	91391.77	366168.4200	166187.94
5	131876.9000	99814.71	362861.3600	156991.12
6	134615.4600	147198.87	127716.8200	156122.51
7	130298.1300	145530.06	323876.6800	155752.60
8	120542.5200	148718.95	311613.2900	152211.77
9	123334.8800	108679.17	304981.6200	149759.96
10	101913.0800	110594.11	229160.9500	146121.95
11	100671.9600	91790.61	249744.5500	144259.40
12	93863.7500	127320.38	249839.4400	141585.52
13	91992.3900	135495.07	252664.9300	134307.35
14	119943.2400	156547.42	256512.9200	132602.65
15	114523.6100	122616.84	261776.2300	129917.04
16	78013.1100	121597.55	264346.0600	126992.93
17	94657.1600	145077.58	282574.3100	125370.37
18	91749.1600	114175.79	294919.5700	124266.90
19	86419.7000	153514.11	211025.0978	122776.86
20	76253.8600	113867.30	298664.4700	118474.03
21	78389.4700	153773.43	299737.2900	111313.02
22	73994.5600	122782.75	303319.2600	110352.25
23	67532.5300	105751.03	304768.7300	108733.99
24	77044.0100	99281.34	140574.8100	108552.04
25	64664.7100	139553.16	137962.6200	107404.34
26	75328.8700	144135.98	134050.0700	105733.54
27	72107.6000	127864.55	353183.8100	105008.31
28	66051.5200	182645.56	118148.2000	103282.38
29	65605.4800	153032.06	107138.3800	101004.64
30	61994.4800	115641.28	91131.2400	99937.59
31	61136.3800	152701.92	88218.2300	97483.56
32	63408.8600	129219.61	46085.2500	97427.84
33	55493.9500	103057.49	214634.8100	96778.92
34	46426.0700	157693.92	210797.6700	96712.80
35	46014.0200	85047.44	205517.6400	96479.51

	R&D Spend	Administration	Marketing Spend	Profit
36	28663.7600	127056.21	201126.8200	90708.19
37	44069.9500	51283.14	197029.4200	89949.14
38	20229.5900	65947.93	185265.1000	81229.06
39	38558.5100	82982.09	174999.3000	81005.76
40	28754.3300	118546.05	172795.6700	78239.91
41	27892.9200	84710.77	164470.7100	77798.83
42	23640.9300	96189.63	148001.1100	71498.49
43	15505.7300	127382.30	35534.1700	69758.98
44	22177.7400	154806.14	28334.7200	65200.33
45	1000.2300	124153.04	1903.9300	64926.08
46	1315.4600	115816.21	297114.4600	49490.75
47	73721.6156	135426.92	211025.0978	42559.73
48	542.0500	51743.15	211025.0978	35673.41
49	73721.6156	116983.80	45173.0600	14681.40

In [11]:

```
# Performing Data Preprocessing
from sklearn.preprocessing import StandardScaler
# Replacing the zero values with column means
for column in ['R&D Spend', 'Administration', 'Marketing Spend', 'Profit']:
    column_mean = data[column].mean()
    data[column] = data[column].replace(0, column_mean)

# Performing Feature Scaling (Standardization)
scaler = StandardScaler()
scaled_data = pd.DataFrame(scaler.fit_transform(data), columns=data.columns)

# Displaying  the first few rows of the preprocessed data
print(scaled_data.head())
```

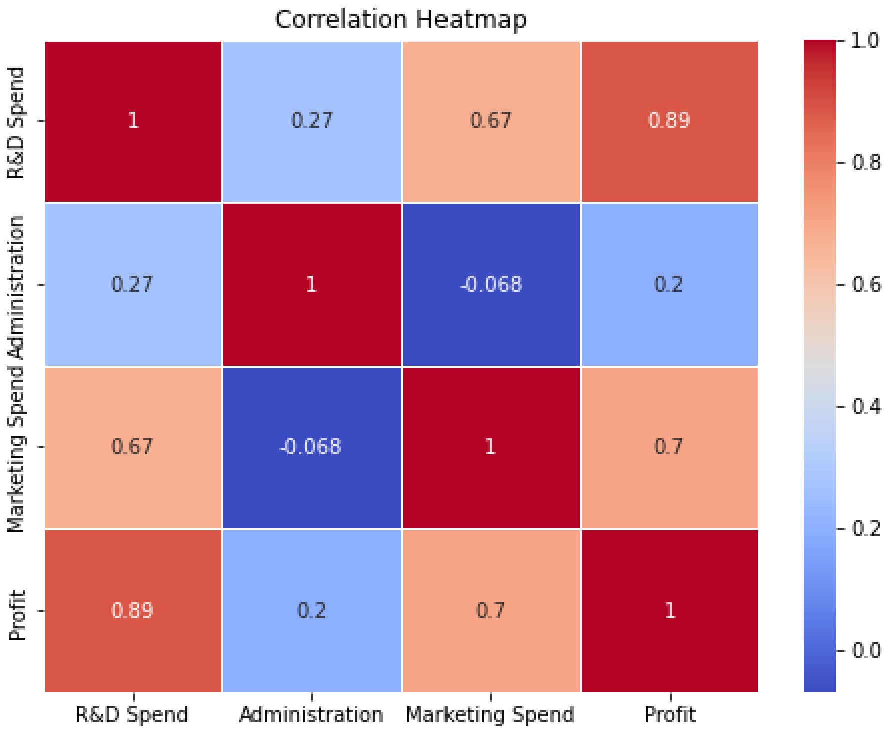
```
   R&D Spend  Administration  Marketing Spend    Profit
0   2.068016        0.560753         2.281641  2.011203
1   2.003850        1.082807         2.025190  1.999430
2   1.790325       -0.728257         1.694445  1.980842
3   1.578830       -0.096365         1.466969  1.776627
4   1.526008       -1.079919         1.310341  1.357740
```

In [7]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [8]:

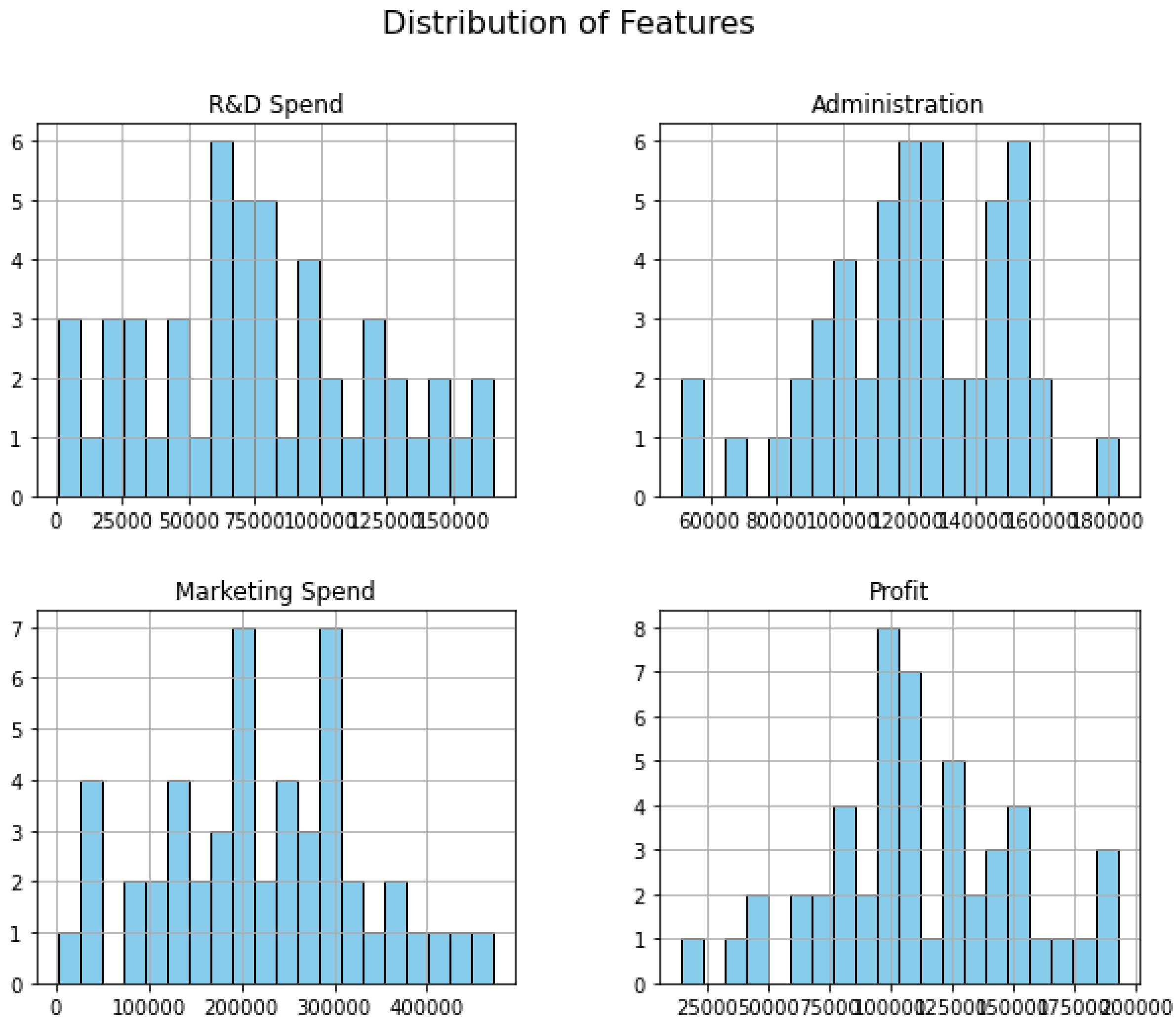
```
# using  Heatmap to Visualize the  correlation between the  variables and Displays c
#, Administration, Marketing Spend, and Profit.
plt.figure(figsize=(8, 6))
sns.heatmap(data.corr(), annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Heatmap")
plt.show()
```



In [9]:

```
# Using Matplotlib Histogram for the Distribution of features that Shows how data is
plt.figure(figsize=(10, 6))
data.hist(bins=20, figsize=(10, 8), color='skyblue', edgecolor='black')
plt.suptitle("Distribution of Features", size=16)
plt.show()
```

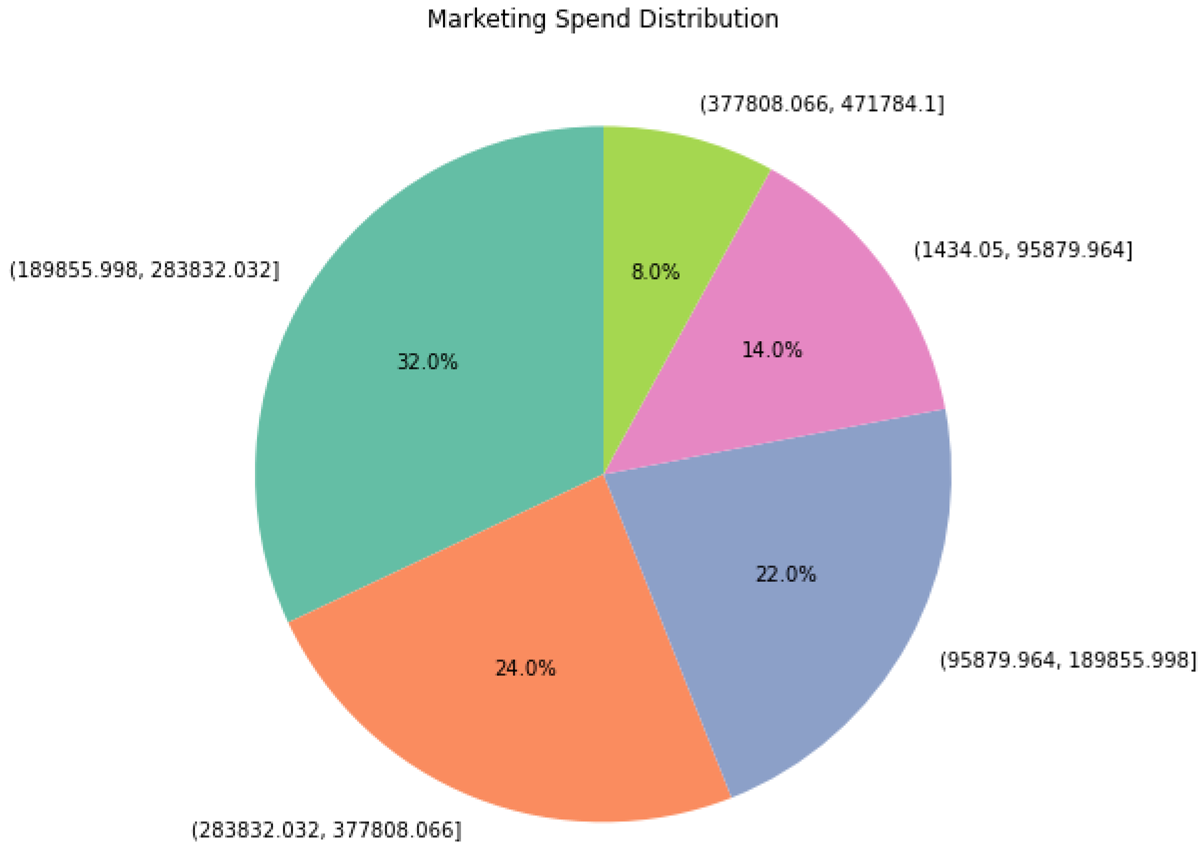
```
<Figure size 720x432 with 0 Axes>
```



In [10]:

```
# using the Pie Chart to see the  Marketing Spend Distribution
marketing_spend_ranges = pd.cut(data['Marketing Spend'], bins=5)
marketing_spend_distribution = marketing_spend_ranges.value_counts()

plt.figure(figsize=(8, 8))
plt.pie(marketing_spend_distribution, labels=marketing_spend_distribution.index, aut
        startangle=90, colors=sns.color_palette("Set2"))
plt.title("Marketing Spend Distribution")
plt.show()
```



In [24]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
```

In [25]:

```
# Generate synthetic data for demonstration
np.random.seed(42)
X = np.random.rand(100, 3) * 100  # 100 samples and 3 features
y = 2.5 * X[:, 0] + 0.5 * X[:, 1]**2 - 1.5 * X[:, 2] + np.random.randn(100) * 10
```

In [26]:

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat
```

In [27]:

```
X_train
```

Out[27]:

```
array([[83.53024956, 32.0780065 , 18.65185104],
       [87.73730719, 74.07686178, 69.7015741 ],
       [35.84657285, 11.58690595, 86.31034259],
       [81.80147659, 86.07305833,  0.69521305],
       [66.35017691,  0.50615838, 16.08080514],
       [66.25222844, 31.17110761, 52.00680212],
       [80.74401552, 89.60912999, 31.8003475 ],
       [28.65412521, 59.08332606,  3.05002499],
       [51.42344384, 59.24145689,  4.64504127],
```

```
[32.53996982, 74.64914051, 64.9632899 ],
[94.88855373, 96.56320331, 80.83973481],
[25.17822958, 49.72485059, 30.08783098],
[55.68012625, 93.61547742, 69.60297967],
[31.09823217, 32.5183322 , 72.96061783],
[79.57926694, 89.00053418, 33.79951569],
[18.34045099, 30.4242243 , 52.47564316],
[88.70864243, 77.98755459, 64.20316462],
[34.92095746, 72.59556789, 89.711026  ],
[24.92922291, 41.0382923 , 75.55511385],
[54.67102793, 18.48544555, 96.95846278],
[50.26790232,  5.14787512, 27.86464642],
[31.43559811, 50.85706912, 90.75664739],
[13.94938607, 29.21446485, 36.63618433],
[ 3.59422738, 46.55980181, 54.26446347],
[62.32981268, 33.08980249,  6.35583503],
[19.59828624,  4.52272889, 32.53303308],
[37.01587003,  1.54566165, 92.83185626],
[72.9007168 , 77.12703467,  7.40446517],
[81.72222002, 55.52008116, 52.96505784],
[44.01524937, 12.20382348, 49.51769101],
[ 0.55221171, 81.54614285, 70.68573438],
[70.80725778,  2.05844943, 96.99098522],
[77.51328234, 93.94989416, 89.48273504],
[63.34037565, 87.14605902, 80.36720769],
[45.60699842, 78.51759614, 19.96737822],
[19.52429878, 72.24521153, 28.07723624],
[43.19450186, 29.12291402, 61.18528947],
[90.04180572, 63.31014573, 33.9029791 ],
[22.87981655,  7.69799098, 28.97514529],
[70.2484084 , 35.94911512, 29.35918443],
[ 4.07751416, 59.08929432, 67.75643618],
[21.58210275, 62.28904758,  8.5347465 ],
[63.35297108, 53.57746841,  9.02897701],
[51.07473026, 41.74110031, 22.21078105],
[90.8265886 , 23.95618907, 14.48948721],
[ 8.4139965 , 16.16287141, 89.85541885],
[36.36296024, 97.17820827, 96.24472949],
[60.64290597,  0.91970516, 10.14715429],
[87.73393534, 25.79416277, 65.9984046 ],
[ 3.73481887, 82.26005607, 36.01906414],
[ 2.43159664, 64.54722959, 17.71106794],
[11.00519245, 22.79351625, 42.71077886],
[64.51727904, 17.4366429 , 69.09377381],
[28.48404944,  3.68869474, 60.9564334 ],
[12.70605127, 52.22432601, 76.99935531],
[ 1.65878289, 51.20930583, 22.64957752],
[24.39896434, 97.30105548, 39.30977247],
[52.27328294, 42.75410184,  2.54191267],
[37.55829526,  9.39819398, 57.8280141 ],
[38.67353463, 93.67299887, 13.75209441],
[24.18522909,  9.31027678, 89.7215758 ],
[85.11366715, 31.69220052, 16.94927467],
[16.12212873, 92.96976523, 80.81203796],
[63.75574714, 88.72127426, 47.22149252],
[59.86584842, 15.60186404, 15.59945203],
[67.21355474, 76.16196153, 23.7637544 ],
[35.67533267, 28.09345097, 54.26960832],
[ 5.80836122, 86.61761458, 60.11150117],
[98.68869366, 77.22447693, 19.87156815],
[99.00538501, 14.00840152, 51.83296524],
[91.32405526, 51.13423989, 50.15162947],
[ 9.36747678, 36.77158031, 26.52023677],
[57.00611701,  9.71764938, 61.50072267],
```

```
       [42.81841483, 96.6654819 , 96.36199771],
       [38.86772897, 27.13490318, 82.87375092],
       [34.10663511, 11.34735212, 92.46936183],
       [22.42693095, 71.21792213, 23.72490875],
       [ 3.43885211, 90.93204021, 25.87799816],
       [79.8295179 , 64.99639308, 70.19668773],
       [48.94527603, 98.56504541, 24.20552715]])
```

In [28]:

```
y_train, y_test
```

Out[28]:

```
(array([ 701.23026982, 2859.45159173,   31.71730731, 3897.93147015,
         136.28052232,  566.90605037, 4180.68924406, 1818.72118268,
        1875.21879028, 2769.94306332, 4786.84545054, 1250.24107458,
        4413.62198447,  487.75631358, 4096.65455002,  431.81891787,
        3174.62481547, 2590.86708829,  776.75969032,  179.75026671,
         103.74440207, 1251.99887678,  415.18600036, 1019.41307263,
         701.50463396,   19.60232377,  -37.37847799, 3138.35813798,
        1669.66782416,  106.88322838, 3232.90774568,   39.98946387,
        4476.90022639, 3849.43066174, 3158.64767309, 2614.20337478,
         433.66251631, 2187.3333204 ,   38.96591116,  783.70325364,
        1657.14140134, 1972.14331774, 1578.59367171,  970.15141656,
         508.14190614,   23.16735749, 4669.04040869,  128.51951217,
         461.50266895, 3344.95031806, 2073.67258214,  223.32133506,
         207.5895722 ,  -12.28611862, 1279.83374083, 1275.14209755,
        4729.86473655, 1038.30126872,   62.89785083, 4458.4410446 ,
         -37.70776005,  696.37757235, 4242.08329451, 4023.69412845,
         241.45852035, 3054.04078127,  412.72956547, 3697.09866972,
        3193.88195972,  266.58024075, 1481.36181106,  672.48954486,
         100.72222675, 4647.66552449,  362.23174324,    5.05009757,
        2562.57988349, 4099.34864811, 2196.4841459 , 4931.2108328 ]),
 array([ 589.1984932 ,  744.00949813, 2440.45701742, 1315.06426644,
         460.44050301, 3934.69038585, 3226.45473969, 4656.03780116,
         292.57073061, 4503.5801189 , 4372.85280585, 2426.9587633 ,
        2290.50641655,  -76.0275682 , 3345.51579544,  386.02497887,
        2100.9407432 , 1713.85015316,    9.21473352, 3027.95804636]))
```

In [29]:

```
# Standardize the feature set
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [30]:

```
X_train_scaled
```

Out[30]:

```
array([[ 1.30326424, -0.56944729, -1.01640827],
       [ 1.45175529,  0.77984283,  0.75000824],
       [-0.37976463, -1.22776141,  1.3247029 ],
       [ 1.24224598,  1.16524261, -1.63774175],
       [ 0.6968815 , -1.58375071, -1.1053713 ],
       [ 0.69342434, -0.59858308,  0.13773582],
       [ 1.20492215,  1.27884538, -0.56144555],
       [-0.63362712,  0.29814804, -1.55626082],
       [ 0.17003197,  0.30322829, -1.50107026],
       [-0.49647351,  0.79822833,  0.58605469],
       [ 1.70416356,  1.50225772,  1.13540957],
       [-0.7563113 , -0.00251011, -0.62070184],
       [ 0.32027457,  1.40755663,  0.74659669],
       [-0.54736065, -0.55530102,  0.86277729],
       [ 1.16381153,  1.25929313, -0.49227056],
       [-0.9976555 , -0.62257807,  0.1539586 ],
       [ 1.48603925,  0.905481  ,  0.55975292],
       [-0.41243488,  0.73225355,  1.44237294],
```

```
       [-0.76510017, -0.2815817 ,  0.95255174],
       [ 0.28465773, -1.0061332 ,  1.6931479 ],
       [ 0.12924632, -1.43462706, -0.69762821],
       [-0.53545308,  0.03386448,  1.47855341],
       [-1.15264122, -0.66144381, -0.39411649],
       [-1.51813382, -0.1041931 ,  0.21585515],
       [ 0.55497992, -0.53694149, -1.44187357],
       [-0.95325932, -1.45471103, -0.53609325],
       [-0.33849347, -1.55035477,  1.55035955],
       [ 0.92808741,  0.87783522, -1.405589  ],
       [ 1.23944856,  0.18367228,  0.17089327],
       [-0.09144546, -1.2079418 ,  0.05160789],
       [-1.6255039 ,  1.01980716,  0.78406204],
       [ 0.8541973 , -1.53388052,  1.69427324],
       [ 1.09089115,  1.41830038,  1.43447365],
       [ 0.59064845,  1.19971472,  1.11905924],
       [-0.03526355,  0.92250955, -0.97088856],
       [-0.95587076,  0.72099771, -0.6902722 ],
       [-0.12041432, -0.66438505,  0.45532876],
       [ 1.53309419,  0.43394236, -0.48869053],
       [-0.8374354 , -1.35269992, -0.65920286],
       [ 0.83447232, -0.44508084, -0.64591438],
       [-1.50107588,  0.29833979,  0.68270281],
       [-0.88323913,  0.40113772, -1.36647914],
       [ 0.59109302,  0.12126229, -1.34937783],
       [ 0.15772388, -0.25900271, -0.8932626 ],
       [ 1.56079365, -0.83037554, -1.1604339 ],
       [-1.34801669, -1.08075014,  1.44736921],
       [-0.36153838,  1.52201588,  1.66845138],
       [ 0.49543937, -1.57046477, -1.31068679],
       [ 1.45163628, -0.77132727,  0.62187161],
       [-1.51317155,  1.04274293, -0.41547001],
       [-1.55916969,  0.47368595, -1.04896114],
       [-1.2565586 , -0.86772854, -0.18392408],
       [ 0.63218808, -1.0398279 ,  0.72897721],
       [-0.63963007, -1.48150591,  0.44740991],
       [-1.1965256 ,  0.07779011,  1.00252521],
       [-1.5864467 ,  0.04518073, -0.87807941],
       [-0.78381602,  1.52596257, -0.30160531],
       [ 0.20002764, -0.22645818, -1.57384247],
       [-0.31934819, -1.29807779,  0.33916071],
       [-0.27998503,  1.40940461, -1.18594908],
       [-0.79135995, -1.30090229,  1.44273799],
       [ 1.35915207, -0.58184201, -1.07532062],
       [-1.07595274,  1.38681194,  1.13445121],
       [ 0.60530928,  1.25032139, -0.0278449 ],
       [ 0.46801257, -1.09877353, -1.12202703],
       [ 0.72735503,  0.84683048, -0.8395268 ],
       [-0.38580867, -0.69745843,  0.21603318],
       [-1.43998441,  1.18273747,  0.41817367],
       [ 1.83829217,  0.88096573, -0.97420377],
       [ 1.84947001, -1.14996643,  0.13172074],
       [ 1.57835208,  0.0427691 ,  0.07354336],
       [-1.31436293, -0.41865764, -0.74414732],
       [ 0.36707634, -1.2878147 ,  0.46624334],
       [-0.13368857,  1.50554361,  1.67250908],
       [-0.2731308 , -0.72825353,  1.20579035],
       [-0.44117696, -1.23545752,  1.53781656],
       [-0.85342032,  0.68799416, -0.84087093],
       [-1.5236179 ,  1.3213463 , -0.76636998],
       [ 1.17264431,  0.48811615,  0.7671401 ],
       [ 0.0825633 ,  1.56657056, -0.82424063]])
```

In [31]:

```
X_test_scaled
```

Out[31]:
```
array([[ 1.3657622 , -0.654041  , -0.3292868 ],
       [ 0.92529616, -0.41844133,  0.52609977],
       [ 0.29179986,  0.62282802,  0.59411125],
       [-0.50422715,  0.06669788,  0.77078051],
       [-1.22192137, -0.51536143,  1.60084764],
       [-0.98648247,  1.26749722,  0.20442825],
       [-1.14759265,  0.97719269, -1.40383817],
       [ 1.67442004,  1.46465822,  1.50380558],
       [ 0.49937763, -1.05217201, -1.43670667],
       [-0.32302894,  1.45433179,  0.8710394 ],
       [ 0.46533538,  1.36167778, -1.35559672],
       [-1.22287831,  0.69141748,  0.97066205],
       [ 1.35239908,  0.51268984,  0.30465732],
       [-1.26418423, -1.49903996,  0.54030239],
       [ 1.21170245,  1.00262562,  1.33843601],
       [ 1.29317018, -0.91783371, -1.03264882],
       [ 1.50354638,  0.42763641,  1.0883993 ],
       [ 0.12909833,  0.25339737,  0.0424065 ],
       [-0.56983891, -1.28622243,  0.70577764],
       [ 0.3360727 ,  0.87686124,  0.04682829]])
```

In [32]:
```
# Initialize a DataFrame to store metrics
metrics_df = pd.DataFrame(columns=['Model', 'MAE', 'MSE', 'RMSE', 'R²'])
```

In [33]:
```
#  Linear Regression
linear_regressor = LinearRegression()
linear_regressor.fit(X_train_scaled, y_train)
y_pred_linear = linear_regressor.predict(X_test_scaled)
```

In [34]:
```
y_pred_linear
```

Out[34]:
```
array([ 875.69788101, 1246.43315835, 2807.80322477, 1867.41773083,
        932.58292107, 3656.71872576, 3094.14371877, 4317.96159309,
        100.0804251 , 4056.09572476, 3865.41369998, 2777.9192008 ,
       2730.38031038, -667.13688696, 3539.43996886,  416.06087492,
       2658.89355797, 2182.76533954, -252.98423419, 3176.93809866])
```

In [35]:
```
# Calculate metrics
metrics_df = metrics_df.append({
    'Model': 'Linear Regression',
    'MAE': mean_absolute_error(y_test, y_pred_linear),
    'MSE': mean_squared_error(y_test, y_pred_linear),
    'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_linear)),
    'R²': r2_score(y_test, y_pred_linear)
}, ignore_index=True)
metrics_df
```

Out[35]:

	Model	MAE	MSE	RMSE	R^2
0	Linear Regression	356.024429	151402.204487	389.104362	0.936473

In [36]:
```
# Polynomial Regression
poly_features = PolynomialFeatures(degree=2)
X_train_poly = poly_features.fit_transform(X_train_scaled)
X_test_poly = poly_features.transform(X_test_scaled)
```

In [37]:
```
X_train_poly
```

Out[37]:
```
array([[ 1.00000000e+00,  1.30326424e+00, -5.69447289e-01,
        -1.01640827e+00,  1.69849767e+00, -7.42140286e-01,
        -1.32464855e+00,  3.24270215e-01,  5.78790936e-01,
         1.03308578e+00],
       [ 1.00000000e+00,  1.45175529e+00,  7.79842829e-01,
         7.50008241e-01,  2.10759342e+00,  1.13214095e+00,
         1.08882843e+00,  6.08154839e-01,  5.84888549e-01,
         5.62512362e-01],
       [ 1.00000000e+00, -3.79764632e-01, -1.22776141e+00,
         1.32470290e+00,  1.44221175e-01,  4.66260360e-01,
        -5.03075307e-01,  1.50739808e+00, -1.62641910e+00,
         1.75483776e+00],
       [ 1.00000000e+00,  1.24224598e+00,  1.16524261e+00,
        -1.63774175e+00,  1.54317507e+00,  1.44751794e+00,
        -2.03447811e+00,  1.35779033e+00, -1.90836647e+00,
         2.68219804e+00],
       [ 1.00000000e+00,  6.96881503e-01, -1.58375071e+00,
        -1.10537130e+00,  4.85643829e-01, -1.10368658e+00,
        -7.70312814e-01,  2.50826632e+00,  1.75063259e+00,
         1.22184572e+00],
       [ 1.00000000e+00,  6.93424343e-01, -5.98583076e-01,
         1.37735822e-01,  4.80837319e-01, -4.15072076e-01,
         9.55093715e-02,  3.58301699e-01, -8.24463318e-02,
         1.89711565e-02],
       [ 1.00000000e+00,  1.20492215e+00,  1.27884538e+00,
        -5.61445546e-01,  1.45183738e+00,  1.54090913e+00,
        -6.76498173e-01,  1.63544551e+00, -7.18002044e-01,
         3.15221101e-01],
       [ 1.00000000e+00, -6.33627124e-01,  2.98148044e-01,
        -1.55626082e+00,  4.01483332e-01, -1.88914688e-01,
         9.86089071e-01,  8.88922560e-02, -4.63996121e-01,
         2.42194775e+00],
       [ 1.00000000e+00,  1.70031973e-01,  3.03228286e-01,
        -1.50107026e+00,  2.89108718e-02,  5.15585037e-02,
        -2.55229937e-01,  9.19473936e-02, -4.55166962e-01,
         2.25321192e+00],
       [ 1.00000000e+00, -4.96473509e-01,  7.98228332e-01,
         5.86054689e-01,  2.46485945e-01, -3.96299221e-01,
        -2.90960628e-01,  6.37168469e-01,  4.67805456e-01,
         3.43460098e-01],
       [ 1.00000000e+00,  1.70416356e+00,  1.50225772e+00,
         1.13540957e+00,  2.90417344e+00,  2.56009287e+00,
         1.93492361e+00,  2.25677826e+00,  1.70567779e+00,
         1.28915489e+00],
       [ 1.00000000e+00, -7.56311302e-01, -2.51011436e-03,
        -6.20701841e-01,  5.72006786e-01,  1.89842786e-03,
         4.69443817e-01,  6.30067409e-06,  1.55803260e-03,
         3.85270775e-01],
       [ 1.00000000e+00,  3.20274569e-01,  1.40755663e+00,
         7.46596688e-01,  1.02575799e-01,  4.50804592e-01,
         2.39115932e-01,  1.98121566e+00,  1.05087712e+00,
         5.57406615e-01],
       [ 1.00000000e+00, -5.47360652e-01, -5.55301020e-01,
         8.62777289e-01,  2.99603683e-01,  3.03949928e-01,
        -4.72250340e-01,  3.08359223e-01, -4.79101109e-01,
         7.44384651e-01],
       [ 1.00000000e+00,  1.16381153e+00,  1.25929313e+00,
        -4.92270563e-01,  1.35445728e+00,  1.46557986e+00,
        -5.72910157e-01,  1.58581918e+00, -6.19912937e-01,
         2.42330307e-01],
       [ 1.00000000e+00, -9.97655501e-01, -6.22578070e-01,
```

```
   1.53958604e-01,  9.95316499e-01,  6.21118437e-01,
  -1.53597649e-01,  3.87603454e-01, -9.58512508e-02,
   2.37032518e-02],
 [ 1.00000000e+00,  1.48603925e+00,  9.05480996e-01,
   5.59752923e-01,  2.20831265e+00,  1.34558030e+00,
   8.31814814e-01,  8.19895834e-01,  5.06845634e-01,
   3.13323335e-01],
 [ 1.00000000e+00, -4.12434877e-01,  7.32253551e-01,
   1.44237294e+00,  1.70102528e-01, -3.02006904e-01,
  -5.94884908e-01,  5.36195263e-01,  1.05618271e+00,
   2.08043971e+00],
 [ 1.00000000e+00, -7.65100167e-01, -2.81581703e-01,
   9.52551736e-01,  5.85378266e-01,  2.15438208e-01,
  -7.28797492e-01,  7.92882554e-02, -2.68221140e-01,
   9.07354809e-01],
 [ 1.00000000e+00,  2.84657735e-01, -1.00613320e+00,
   1.69314790e+00,  8.10300260e-02, -2.86403596e-01,
   4.81967646e-01,  1.01230401e+00, -1.70353231e+00,
   2.86674981e+00],
 [ 1.00000000e+00,  1.29246323e-01, -1.43462706e+00,
  -6.97628211e-01,  1.67046120e-02, -1.85420273e-01,
  -9.01658812e-02,  2.05815481e+00,  1.00083631e+00,
   4.86685121e-01],
 [ 1.00000000e+00, -5.35453084e-01,  3.38644787e-02,
   1.47855341e+00,  2.86710005e-01, -1.81328396e-02,
  -7.91695984e-01,  1.14680292e-03,  5.00704406e-02,
   2.18612019e+00],
 [ 1.00000000e+00, -1.15264122e+00, -6.61443807e-01,
  -3.94116488e-01,  1.32858179e+00,  7.62407399e-01,
   4.54274911e-01,  4.37507910e-01,  2.60685910e-01,
   1.55327806e-01],
 [ 1.00000000e+00, -1.51813382e+00, -1.04193101e-01,
   2.15855154e-01,  2.30473030e+00,  1.58179071e-01,
  -3.27697011e-01,  1.08562023e-02, -2.24906179e-02,
   4.65934476e-02],
 [ 1.00000000e+00,  5.54979923e-01, -5.36941491e-01,
  -1.44187357e+00,  3.08002715e-01, -2.97991747e-01,
  -8.00210885e-01,  2.88306164e-01,  7.74201746e-01,
   2.07899940e+00],
 [ 1.00000000e+00, -9.53259322e-01, -1.45471103e+00,
  -5.36093245e-01,  9.08703336e-01,  1.38671685e+00,
   5.11035884e-01,  2.11618418e+00,  7.79860758e-01,
   2.87395968e-01],
 [ 1.00000000e+00, -3.38493467e-01, -1.55035477e+00,
   1.55035955e+00,  1.14577827e-01,  5.24784959e-01,
  -5.24786577e-01,  2.40359990e+00, -2.40360731e+00,
   2.40361472e+00],
 [ 1.00000000e+00,  9.28087410e-01,  8.77835218e-01,
  -1.40558900e+00,  8.61346241e-01,  8.14707814e-01,
  -1.30450945e+00,  7.70594670e-01, -1.23387552e+00,
   1.97568043e+00],
 [ 1.00000000e+00,  1.23944856e+00,  1.83672278e-01,
   1.70893273e-01,  1.53623274e+00,  2.27652341e-01,
   2.11813422e-01,  3.37355056e-02,  3.13883568e-02,
   2.92045108e-02],
 [ 1.00000000e+00, -9.14454553e-02, -1.20794180e+00,
   5.16078917e-02,  8.36227129e-03,  1.10460788e-01,
  -4.71930715e-03,  1.45912340e+00, -6.23393298e-02,
   2.66337448e-03],
 [ 1.00000000e+00, -1.62550390e+00,  1.01980716e+00,
   7.84062040e-01,  2.64226294e+00, -1.65770052e+00,
  -1.27449591e+00,  1.04000664e+00,  7.99592081e-01,
   6.14753282e-01],
 [ 1.00000000e+00,  8.54197303e-01, -1.53388052e+00,
```

```
         1.69427324e+00,  7.29653033e-01, -1.31023660e+00,
         1.44724363e+00,  2.35278945e+00, -2.59881271e+00,
         2.87056180e+00],
       [ 1.00000000e+00,  1.09089115e+00,  1.41830038e+00,
         1.43447365e+00,  1.19004350e+00,  1.54721133e+00,
         1.56485461e+00,  2.01157596e+00,  2.03451452e+00,
         2.05771464e+00],
       [ 1.00000000e+00,  5.90648453e-01,  1.19971472e+00,
         1.11905924e+00,  3.48865595e-01,  7.08609642e-01,
         6.60970609e-01,  1.43931541e+00,  1.34255184e+00,
         1.25229358e+00],
       [ 1.00000000e+00, -3.52635544e-02,  9.22509551e-01,
        -9.70888557e-01,  1.24351827e-03, -3.25309657e-02,
         3.42369815e-02,  8.51023872e-01, -8.95653967e-01,
         9.42624591e-01],
       [ 1.00000000e+00, -9.55870762e-01,  7.20997712e-01,
        -6.90272203e-01,  9.13688913e-01, -6.89180632e-01,
         6.59811017e-01,  5.19837700e-01, -4.97684679e-01,
         4.76475715e-01],
       [ 1.00000000e+00, -1.20414315e-01, -6.64385045e-01,
         4.55328762e-01,  1.44996073e-02,  8.00014703e-02,
        -5.48281011e-02,  4.41407488e-01, -3.02513620e-01,
         2.07324281e-01],
       [ 1.00000000e+00,  1.53309419e+00,  4.33942364e-01,
        -4.88690534e-01,  2.35037778e+00,  6.65274515e-01,
        -7.49208616e-01,  1.88305976e-01, -2.12063526e-01,
         2.38818438e-01],
       [ 1.00000000e+00, -8.37435405e-01, -1.35269992e+00,
        -6.59202860e-01,  7.01298057e-01,  1.13279881e+00,
         5.52039814e-01,  1.82979708e+00,  8.91703658e-01,
         4.34548411e-01],
       [ 1.00000000e+00,  8.34472322e-01, -4.45080837e-01,
        -6.45914383e-01,  6.96344056e-01, -3.71407640e-01,
        -5.38997675e-01,  1.98096952e-01,  2.87484114e-01,
         4.17205390e-01],
       [ 1.00000000e+00, -1.50107588e+00,  2.98339785e-01,
         6.82702808e-01,  2.25322879e+00, -4.47830655e-01,
        -1.02478872e+00,  8.90066274e-02,  2.03677409e-01,
         4.66083124e-01],
       [ 1.00000000e+00, -8.83239125e-01,  4.01137717e-01,
        -1.36647914e+00,  7.80111352e-01, -3.54300526e-01,
         1.20692784e+00,  1.60911468e-01, -5.48146321e-01,
         1.86726523e+00],
       [ 1.00000000e+00,  5.91093017e-01,  1.21262287e-01,
        -1.34937783e+00,  3.49390955e-01,  7.16772908e-02,
        -7.97607813e-01,  1.47045421e-02, -1.63628641e-01,
         1.82082053e+00],
       [ 1.00000000e+00,  1.57723882e-01, -2.59002709e-01,
        -8.93262596e-01,  2.48768229e-02, -4.08509127e-02,
        -1.40888844e-01,  6.70824032e-02,  2.31357432e-01,
         7.97918066e-01],
       [ 1.00000000e+00,  1.56079365e+00, -8.30375544e-01,
        -1.16043390e+00,  2.43607681e+00, -1.29604487e+00,
        -1.81119786e+00,  6.89523544e-01,  9.63595929e-01,
         1.34660683e+00],
       [ 1.00000000e+00, -1.34801669e+00, -1.08075014e+00,
         1.44736921e+00,  1.81714899e+00,  1.45686922e+00,
        -1.95107784e+00,  1.16802087e+00, -1.56424448e+00,
         2.09487762e+00],
       [ 1.00000000e+00, -3.61538376e-01,  1.52201588e+00,
         1.66845138e+00,  1.30709997e-01, -5.50267150e-01,
        -6.03209204e-01,  2.31653234e+00,  2.53940950e+00,
         2.78373002e+00],
       [ 1.00000000e+00,  4.95439365e-01, -1.57046477e+00,
```

```
        -1.31068679e+00,  2.45460165e-01, -7.78070067e-01,
        -6.49365832e-01,  2.46635958e+00,  2.05838742e+00,
         1.71789986e+00],
       [ 1.00000000e+00,  1.45163628e+00, -7.71327274e-01,
         6.21871606e-01,  2.10724789e+00, -1.11968665e+00,
         9.02731384e-01,  5.94945763e-01, -4.79666530e-01,
         3.86724294e-01],
       [ 1.00000000e+00, -1.51317155e+00,  1.04274293e+00,
        -4.15470008e-01,  2.28968813e+00, -1.57784893e+00,
         6.28677395e-01,  1.08731281e+00, -4.33228413e-01,
         1.72615328e-01],
       [ 1.00000000e+00, -1.55916969e+00,  4.73685948e-01,
        -1.04896114e+00,  2.43101013e+00, -7.38556774e-01,
         1.63550842e+00,  2.24378378e-01, -4.96878153e-01,
         1.10031947e+00],
       [ 1.00000000e+00, -1.25655860e+00, -8.67728537e-01,
        -1.83924082e-01,  1.57893953e+00,  1.09035176e+00,
         2.31111388e-01,  7.52952813e-01,  1.59596175e-01,
         3.38280680e-02],
       [ 1.00000000e+00,  6.32188085e-01, -1.03982790e+00,
         7.28977207e-01,  3.99661775e-01, -6.57366809e-01,
         4.60850704e-01,  1.08124206e+00, -7.58010839e-01,
         5.31407769e-01],
       [ 1.00000000e+00, -6.39630068e-01, -1.48150591e+00,
         4.47409911e-01,  4.09126624e-01,  9.47615723e-01,
        -2.86176832e-01,  2.19485975e+00, -6.62840425e-01,
         2.00175628e-01],
       [ 1.00000000e+00, -1.19652560e+00,  7.77901095e-02,
         1.00252521e+00,  1.43167351e+00, -9.30778573e-02,
        -1.19954707e+00,  6.05130114e-03,  7.79865458e-02,
         1.00505679e+00],
       [ 1.00000000e+00, -1.58644670e+00,  4.51807280e-02,
        -8.78079412e-01,  2.51681312e+00, -7.16768167e-02,
         1.39302618e+00,  2.04129819e-03, -3.96722671e-02,
         7.71023455e-01],
       [ 1.00000000e+00, -7.83816016e-01,  1.52596257e+00,
        -3.01605306e-01,  6.14367548e-01, -1.19607390e+00,
         2.36403069e-01,  2.32856177e+00, -4.60238408e-01,
         9.09657604e-02],
       [ 1.00000000e+00,  2.00027641e-01, -2.26458180e-01,
        -1.57384247e+00,  4.00110571e-02, -4.52978955e-02,
        -3.14811996e-01,  5.12833074e-02,  3.56409501e-01,
         2.47698011e+00],
       [ 1.00000000e+00, -3.19348187e-01, -1.29807779e+00,
         3.39160714e-01,  1.01983264e-01,  4.14538788e-01,
        -1.08310359e-01,  1.68500595e+00, -4.40256990e-01,
         1.15029990e-01],
       [ 1.00000000e+00, -2.79985030e-01,  1.40940461e+00,
        -1.18594908e+00,  7.83916169e-02, -3.94612192e-01,
         3.32047988e-01,  1.98642136e+00, -1.67148210e+00,
         1.40647521e+00],
       [ 1.00000000e+00, -7.91359952e-01, -1.30090229e+00,
         1.44273799e+00,  6.26250574e-01,  1.02948197e+00,
        -1.14172506e+00,  1.69234677e+00, -1.87686115e+00,
         2.08149290e+00],
       [ 1.00000000e+00,  1.35915207e+00, -5.81842013e-01,
        -1.07532062e+00,  1.84729436e+00, -7.90811777e-01,
        -1.46152425e+00,  3.38540128e-01,  6.25666715e-01,
         1.15631444e+00],
       [ 1.00000000e+00, -1.07595274e+00,  1.38681194e+00,
         1.13445121e+00,  1.15767431e+00, -1.49214412e+00,
        -1.22061589e+00,  1.92324736e+00,  1.57327048e+00,
         1.28697954e+00],
       [ 1.00000000e+00,  6.05309281e-01,  1.25032139e+00,
```

```
        -2.78448987e-02,  3.66399326e-01,  7.56831143e-01,
        -1.68547756e-02,  1.56330358e+00, -3.48150725e-02,
         7.75338382e-04],
       [ 1.00000000e+00,  4.68012573e-01, -1.09877353e+00,
        -1.12202703e+00,  2.19035769e-01, -5.14239828e-01,
        -5.25122756e-01,  1.20730327e+00,  1.23285360e+00,
         1.25894465e+00],
       [ 1.00000000e+00,  7.27355030e-01,  8.46830477e-01,
        -8.39526795e-01,  5.29045339e-01,  6.15946407e-01,
        -6.10634037e-01,  7.17121856e-01, -7.10936876e-01,
         7.04805240e-01],
       [ 1.00000000e+00, -3.85808674e-01, -6.97458431e-01,
         2.16033176e-01,  1.48848333e-01,  2.69085512e-01,
        -8.33474731e-02,  4.86448262e-01, -1.50674160e-01,
         4.66703330e-02],
       [ 1.00000000e+00, -1.43998441e+00,  1.18273747e+00,
         4.18173665e-01,  2.07355511e+00, -1.70312353e+00,
        -6.02163560e-01,  1.39886793e+00,  4.94589664e-01,
         1.74869214e-01],
       [ 1.00000000e+00,  1.83829217e+00,  8.80965729e-01,
        -9.74203766e-01,  3.37931810e+00,  1.61947240e+00,
        -1.79087115e+00,  7.76100616e-01, -8.58240131e-01,
         9.49072978e-01],
       [ 1.00000000e+00,  1.84947001e+00, -1.14996643e+00,
         1.31720738e-01,  3.42053933e+00, -2.12682843e+00,
         2.43613555e-01,  1.32242279e+00, -1.51474427e-01,
         1.73503529e-02],
       [ 1.00000000e+00,  1.57835208e+00,  4.27690972e-02,
         7.35433557e-02,  2.49119530e+00,  6.75046936e-02,
         1.16077309e-01,  1.82919567e-03,  3.14538292e-03,
         5.40862516e-03],
       [ 1.00000000e+00, -1.31436293e+00, -4.18657637e-01,
        -7.44147316e-01,  1.72754991e+00,  5.50268079e-01,
         9.78079647e-01,  1.75274217e-01,  3.11542957e-01,
         5.53755228e-01],
       [ 1.00000000e+00,  3.67076344e-01, -1.28781470e+00,
         4.66243344e-01,  1.34745042e-01, -4.72726312e-01,
         1.71146902e-01,  1.65846670e+00, -6.00435032e-01,
         2.17382856e-01],
       [ 1.00000000e+00, -1.33688571e-01,  1.50554361e+00,
         1.67250908e+00,  1.78726341e-02, -2.01273974e-01,
        -2.23595350e-01,  2.26666155e+00,  2.51803536e+00,
         2.79728664e+00],
       [ 1.00000000e+00, -2.73130804e-01, -7.28253533e-01,
         1.20579035e+00,  7.46004362e-02,  1.98908473e-01,
        -3.29338489e-01,  5.30353209e-01, -8.78121086e-01,
         1.45393038e+00],
       [ 1.00000000e+00, -4.41176957e-01, -1.23545752e+00,
         1.53781656e+00,  1.94637107e-01,  5.45055387e-01,
        -6.78449228e-01,  1.52635527e+00, -1.89990702e+00,
         2.36487976e+00],
       [ 1.00000000e+00, -8.53420320e-01,  6.87994159e-01,
        -8.40870928e-01,  7.28326243e-01, -5.87148196e-01,
         7.17616336e-01,  4.73335963e-01, -5.78514287e-01,
         7.07063917e-01],
       [ 1.00000000e+00, -1.52361790e+00,  1.32134630e+00,
        -7.66369981e-01,  2.32141151e+00, -2.01322687e+00,
         1.16765502e+00,  1.74595603e+00, -1.01264014e+00,
         5.87322948e-01],
       [ 1.00000000e+00,  1.17264431e+00,  4.88116148e-01,
         7.67140105e-01,  1.37509468e+00,  5.72386624e-01,
         8.99582481e-01,  2.38257374e-01,  3.74453473e-01,
         5.88503940e-01],
       [ 1.00000000e+00,  8.25632989e-02,  1.56657056e+00,
```

```
        -8.24240626e-01,  6.81669832e-03,  1.29341234e-01,
        -6.80520252e-02,  2.45414333e+00, -1.29123110e+00,
         6.79372610e-01]])
```

In [38]:

```
X_test_poly
```

Out[38]:

```
array([[ 1.00000000e+00,  1.36576220e+00, -6.54041003e-01,
        -3.29286798e-01,  1.86530637e+00, -8.93264476e-01,
        -4.49727460e-01,  4.27769634e-01,  2.15367068e-01,
         1.08429795e-01],
       [ 1.00000000e+00,  9.25296157e-01, -4.18441329e-01,
         5.26099770e-01,  8.56172979e-01, -3.87182153e-01,
         4.86798096e-01,  1.75093145e-01, -2.20141887e-01,
         2.76780968e-01],
       [ 1.00000000e+00,  2.91799856e-01,  6.22828020e-01,
         5.94111254e-01,  8.51471558e-02,  1.81741126e-01,
         1.73361578e-01,  3.87914742e-01,  3.70029136e-01,
         3.52968182e-01],
       [ 1.00000000e+00, -5.04227149e-01,  6.66978768e-02,
         7.70780505e-01,  2.54245018e-01, -3.36308803e-02,
        -3.88648457e-01,  4.44860677e-03,  5.14094232e-02,
         5.94102587e-01],
       [ 1.00000000e+00, -1.22192137e+00, -5.15361428e-01,
         1.60084764e+00,  1.49309183e+00,  6.29731141e-01,
        -1.95610994e+00,  2.65597401e-01, -8.25015126e-01,
         2.56271317e+00],
       [ 1.00000000e+00, -9.86482474e-01,  1.26749722e+00,
         2.04428254e-01,  9.73147671e-01, -1.25036379e+00,
        -2.01664889e-01,  1.60654920e+00,  2.59112243e-01,
         4.17909108e-02],
       [ 1.00000000e+00, -1.14759265e+00,  9.77192692e-01,
        -1.40383817e+00,  1.31696889e+00, -1.12141915e+00,
         1.61103437e+00,  9.54905557e-01, -1.37182040e+00,
         1.97076161e+00],
       [ 1.00000000e+00,  1.67442004e+00,  1.46465822e+00,
         1.50380558e+00,  2.80368247e+00,  2.45245308e+00,
         2.51800219e+00,  2.14522371e+00,  2.20256121e+00,
         2.26143121e+00],
       [ 1.00000000e+00,  4.99377632e-01, -1.05217201e+00,
        -1.43670667e+00,  2.49378020e-01, -5.25431167e-01,
        -7.17459177e-01,  1.10706594e+00,  1.51166255e+00,
         2.06412607e+00],
       [ 1.00000000e+00, -3.23028942e-01,  1.45433179e+00,
         8.71039401e-01,  1.04347697e-01, -4.69791260e-01,
        -2.81370936e-01,  2.11508096e+00,  1.26678029e+00,
         7.58709637e-01],
       [ 1.00000000e+00,  4.65335375e-01,  1.36167778e+00,
        -1.35559672e+00,  2.16537012e-01,  6.33636842e-01,
        -6.30807106e-01,  1.85416638e+00, -1.84588593e+00,
         1.83764246e+00],
       [ 1.00000000e+00, -1.22287831e+00,  6.91417485e-01,
         9.70662050e-01,  1.49543136e+00, -8.45519446e-01,
        -1.18700157e+00,  4.78058138e-01,  6.71132713e-01,
         9.42184815e-01],
       [ 1.00000000e+00,  1.35239908e+00,  5.12689837e-01,
         3.04657317e-01,  1.82898328e+00,  6.93361266e-01,
         4.12018276e-01,  2.62850869e-01,  1.56194710e-01,
         9.28160807e-02],
       [ 1.00000000e+00, -1.26418423e+00, -1.49903996e+00,
         5.40302392e-01,  1.59816177e+00,  1.89506269e+00,
        -6.83041764e-01,  2.24712081e+00, -8.09934878e-01,
         2.91926675e-01],
       [ 1.00000000e+00,  1.21170245e+00,  1.00262562e+00,
```

```
                 1.33843601e+00,  1.46822284e+00,  1.21488393e+00,
                 1.62178620e+00,  1.00525814e+00,  1.34195024e+00,
                 1.79141096e+00],
               [ 1.00000000e+00,  1.29317018e+00, -9.17833709e-01,
                -1.03264882e+00,  1.67228911e+00, -1.18691518e+00,
                -1.33539065e+00,  8.42418717e-01,  9.47799893e-01,
                 1.06636358e+00],
               [ 1.00000000e+00,  1.50354638e+00,  4.27636410e-01,
                 1.08839930e+00,  2.26065172e+00,  6.42971176e-01,
                 1.63645882e+00,  1.82872899e-01,  4.65439168e-01,
                 1.18461303e+00],
               [ 1.00000000e+00,  1.29098328e-01,  2.53397373e-01,
                 4.24065027e-02,  1.66663782e-02,  3.27131772e-02,
                 5.47460859e-03,  6.42102288e-02,  1.07456964e-02,
                 1.79831147e-03],
               [ 1.00000000e+00, -5.69838905e-01, -1.28622243e+00,
                 7.05777643e-01,  3.24716378e-01,  7.32939581e-01,
                -4.02179560e-01,  1.65436814e+00, -9.07787034e-01,
                 4.98122081e-01],
               [ 1.00000000e+00,  3.36072696e-01,  8.76861239e-01,
                 4.68282861e-02,  1.12944857e-01,  2.94689121e-01,
                 1.57377084e-02,  7.68885633e-01,  4.10619090e-02,
                 2.19288838e-03]])
```

In [39]:

```
poly_regressor = LinearRegression()
poly_regressor.fit(X_train_poly, y_train)
y_pred_poly = poly_regressor.predict(X_test_poly)
```

In [40]:

```
y_pred_poly
```

Out[40]:

```
array([ 594.30466217,  768.72445078, 2435.82136751, 1325.25364233,
        466.19628761, 3950.28248561, 3241.12327764, 4657.62056776,
        288.75422355, 4504.60085412, 4378.96697334, 2466.1206863 ,
       2294.65804252,  -62.46205755, 3361.84687331,  410.47227271,
       2105.31378629, 1717.43419977,   22.1034541 , 3038.23703586])
```

In [41]:

```
# Calculate metrics
metrics_df = metrics_df.append({
    'Model': 'Polynomial Regression',
    'MAE': mean_absolute_error(y_test, y_pred_poly),
    'MSE': mean_squared_error(y_test, y_pred_poly),
    'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_poly)),
    'R²': r2_score(y_test, y_pred_poly)
}, ignore_index=True)
```

In [42]:

```
metrics_df
```

Out[42]:

	Model	MAE	MSE	RMSE	R^2
0	Linear Regression	356.024429	151402.204487	389.104362	0.936473
1	Polynomial Regression	11.098949	210.607707	14.512329	0.999912

In [43]:

```
#  Decision Tree Regression
decision_tree_regressor = DecisionTreeRegressor(random_state=42)
decision_tree_regressor.fit(X_train_scaled, y_train)
y_pred_decision_tree = decision_tree_regressor.predict(X_test_scaled)
```

In [44]:
```
y_pred_decision_tree
```

Out[44]:
```
array([ 508.14190614,  783.70325364, 2562.57988349, 1279.83374083,
        776.75969032, 4023.69412845, 3232.90774568, 4669.04040869,
        207.5895722 , 4476.90022639, 4242.08329451, 2562.57988349,
       2196.4841459 ,  -12.28611862, 3232.90774568,  223.32133506,
       2196.4841459 , 1669.66782416,   31.71730731, 3158.64767309])
```

In [45]:
```
# Calculate metrics
metrics_df = metrics_df.append({
    'Model': 'Decision Tree Regression',
    'MAE': mean_absolute_error(y_test, y_pred_decision_tree),
    'MSE': mean_squared_error(y_test, y_pred_decision_tree),
    'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_decision_tree)),
    'R²': r2_score(y_test, y_pred_decision_tree)
}, ignore_index=True)
```

In [46]:
```
metrics_df
```

Out[46]:

	Model	MAE	MSE	RMSE	R^2
0	Linear Regression	356.024429	151402.204487	389.104362	0.936473
1	Polynomial Regression	11.098949	210.607707	14.512329	0.999912
2	Decision Tree Regression	90.346365	12832.074622	113.278747	0.994616

In [47]:
```
# Random Forest Regression
random_forest_regressor = RandomForestRegressor(n_estimators=100, random_state=42)
random_forest_regressor.fit(X_train_scaled, y_train)
y_pred_random_forest = random_forest_regressor.predict(X_test_scaled)
```

In [48]:
```
# Calculate metrics
metrics_df = metrics_df.append({
    'Model': 'Random Forest Regression',
    'MAE': mean_absolute_error(y_test, y_pred_random_forest),
    'MSE': mean_squared_error(y_test, y_pred_random_forest),
    'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_random_forest)),
    'R²': r2_score(y_test, y_pred_random_forest)
}, ignore_index=True)
```

In [49]:
```
metrics_df
```

Out[49]:

	Model	MAE	MSE	RMSE	R^2
0	Linear Regression	356.024429	151402.204487	389.104362	0.936473
1	Polynomial Regression	11.098949	210.607707	14.512329	0.999912
2	Decision Tree Regression	90.346365	12832.074622	113.278747	0.994616
3	Random Forest Regression	91.127631	13059.691346	114.279007	0.994520

In [50]:
```
# Support Vector Regression (SVR)
svr_regressor = SVR(kernel='rbf')
```

```python
svr_regressor.fit(X_train_scaled, y_train)
y_pred_svr = svr_regressor.predict(X_test_scaled)
```

In [51]:

```python
# Calculate metrics
metrics_df = metrics_df.append({
    'Model': 'Support Vector Regression',
    'MAE': mean_absolute_error(y_test, y_pred_svr),
    'MSE': mean_squared_error(y_test, y_pred_svr),
    'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_svr)),
    'R²': r2_score(y_test, y_pred_svr)
}, ignore_index=True)
```

In [52]:

```python
metrics_df
```

Out[52]:

	Model	MAE	MSE	RMSE	R^2
0	Linear Regression	356.024429	1.514022e+05	389.104362	0.936473
1	Polynomial Regression	11.098949	2.106077e+02	14.512329	0.999912
2	Decision Tree Regression	90.346365	1.283207e+04	113.278747	0.994616
3	Random Forest Regression	91.127631	1.305969e+04	114.279007	0.994520
4	Support Vector Regression	1460.631351	3.035555e+06	1742.284512	-0.273687

In [53]:

```python
# 6. K-Nearest Neighbors Regression (KNN)
knn_regressor = KNeighborsRegressor(n_neighbors=5)
knn_regressor.fit(X_train_scaled, y_train)
y_pred_knn = knn_regressor.predict(X_test_scaled)
```

In [54]:

```python
y_pred_knn
```

Out[54]:

```
array([ 602.95947739,  877.40558535, 3450.63479677, 1313.25833697,
        658.81038107, 3554.95290667, 2664.38956125, 3829.45054917,
        449.2128915 , 4148.47085509, 3748.13969395, 2821.81297042,
       2276.31803766,  145.78386853, 3633.82241526,  586.18230446,
       2276.31803766, 1030.67852241,  106.47551861, 3671.59090151])
```

In [55]:

```python
# Calculate metrics
metrics_df = metrics_df.append({
    'Model': 'K-Nearest Neighbors Regression',
    'MAE': mean_absolute_error(y_test, y_pred_knn),
    'MSE': mean_squared_error(y_test, y_pred_knn),
    'RMSE': np.sqrt(mean_squared_error(y_test, y_pred_knn)),
    'R²': r2_score(y_test, y_pred_knn)
}, ignore_index=True)
```

In [56]:

```python
metrics_df
```

Out[56]:

	Model	MAE	MSE	RMSE	R^2
0	Linear Regression	356.024429	1.514022e+05	389.104362	0.936473
1	Polynomial Regression	11.098949	2.106077e+02	14.512329	0.999912
2	Decision Tree Regression	90.346365	1.283207e+04	113.278747	0.994616

	Model	MAE	MSE	RMSE	R^2
3	Random Forest Regression	91.127631	1.305969e+04	114.279007	0.994520
4	Support Vector Regression	1460.631351	3.035555e+06	1742.284512	-0.273687
5	K-Nearest Neighbors Regression	349.056282	2.005883e+05	447.870838	0.915835

In [57]:

```
# Print the metrics DataFrame
print("Model Comparison Metrics:")
print(metrics_df)
```

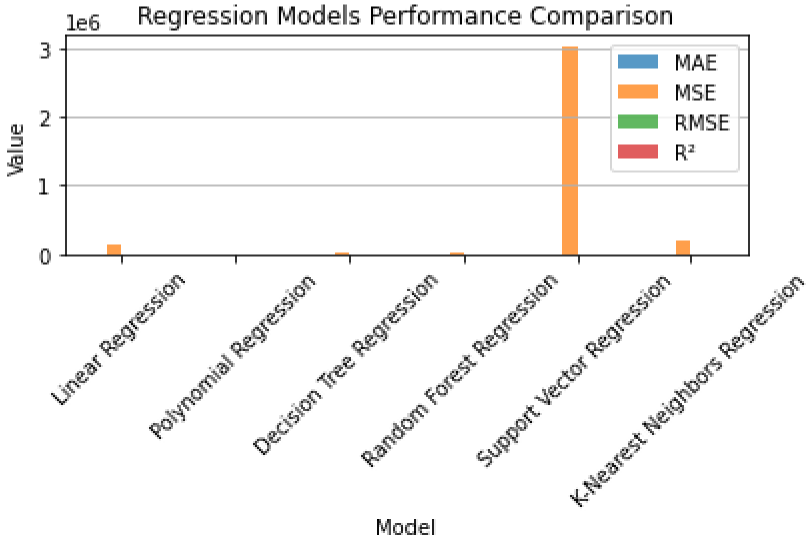
```
Model Comparison Metrics:
                            Model          MAE           MSE         RMSE  \
0               Linear Regression   356.024429  1.514022e+05   389.104362
1           Polynomial Regression    11.098949  2.106077e+02    14.512329
2        Decision Tree Regression    90.346365  1.283207e+04   113.278747
3        Random Forest Regression    91.127631  1.305969e+04   114.279007
4       Support Vector Regression  1460.631351  3.035555e+06  1742.284512
5  K-Nearest Neighbors Regression   349.056282  2.005883e+05   447.870838

         R²
0  0.936473
1  0.999912
2  0.994616
3  0.994520
4 -0.273687
5  0.915835
```

In [58]:

```
# Plot the metrics comparison
plt.figure(figsize=(14, 8))
metrics_df.set_index('Model').plot(kind='bar', alpha=0.75)
plt.title('Regression Models Performance Comparison')
plt.ylabel('Value')
plt.xticks(rotation=45)
plt.grid(axis='y')
plt.tight_layout()
plt.show()
```

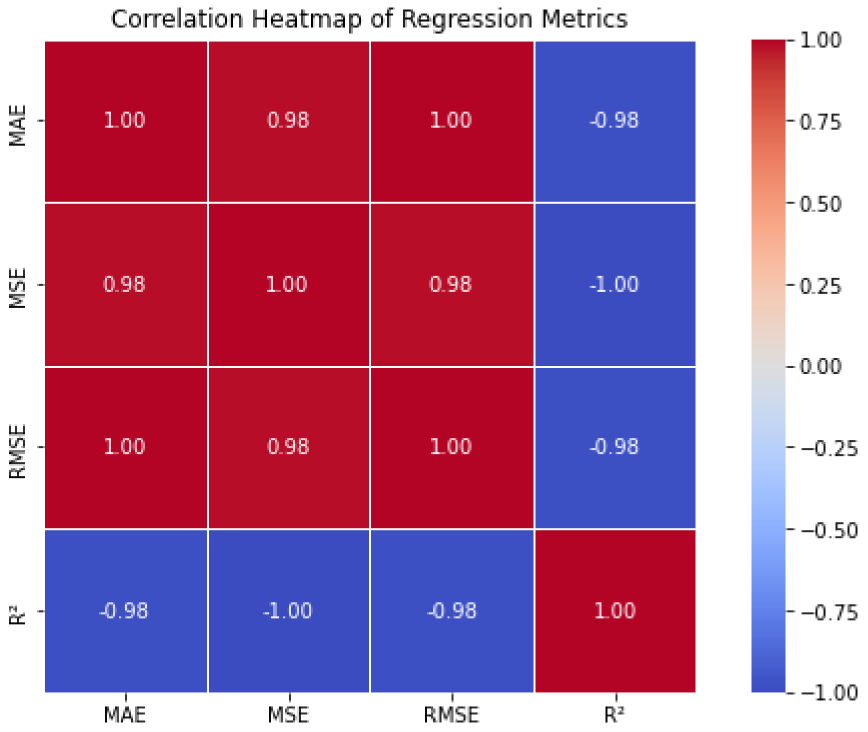
```
<Figure size 1008x576 with 0 Axes>
```



In [62]:

```
# Create a heatmap for the metrics
plt.figure(figsize=(10, 6))
sns.heatmap(metrics_df.set_index('Model').corr(), annot=True, cmap='coolwarm', fmt="
```

```
plt.title('Correlation Heatmap of Regression Metrics')
plt.show()
```



In [63]:

```
# Determine the best model based on R²
best_model_index = metrics_df['R²'].idxmax()
best_model = metrics_df.loc[best_model_index]
```

In [64]:

```
print("\nBest Model:")
print(best_model)
```

```
Best Model:
Model    Polynomial Regression
MAE                  11.098949
MSE                 210.607707
RMSE                 14.512329
R²                    0.999912
Name: 1, dtype: object
```

In [65]:

```
# Function to predict profit
def predict_profit(model, scaler, poly_features, r_and_d_spend, admin_cost, marketin
    input_data = np.array([[r_and_d_spend, admin_cost, marketing_spend]])
    input_data_scaled = scaler.transform(input_data)

    # If using polynomial features, transform the input data
    if poly_features is not None:
        input_data_scaled = poly_features.transform(input_data_scaled)

    predicted_profit = model.predict(input_data_scaled)
    return predicted_profit[0]
```

In [69]:

```
# Take user Input for prediction
r_and_d_spend = float(input("Enter the R&D Spend: "))
admin_cost = float(input("Enter the Administration Cost: "))
marketing_spend = float(input("Enter the Marketing Spend: "))
```

```
Enter the R&D Spend: 150000
Enter the Administration Cost: 50000
Enter the Marketing Spend: 60000
```

In [70]:

```python
# Use the best model for prediction
if best_model['Model'] == 'Linear Regression':
    predicted_profit = predict_profit(linear_regressor, scaler, None, r_and_d_spend,
elif best_model['Model'] == 'Polynomial Regression':
    predicted_profit = predict_profit(poly_regressor, scaler, poly_features, r_and_d
elif best_model['Model'] == 'Decision Tree Regression':
    predicted_profit = predict_profit(decision_tree_regressor, scaler, None, r_and_d
elif best_model['Model'] == 'Random Forest Regression':
    predicted_profit = predict_profit(random_forest_regressor, scaler, None, r_and_d
elif best_model['Model'] == 'Support Vector Regression':
    predicted_profit = predict_profit(svr_regressor, scaler, None, r_and_d_spend, ad
elif best_model['Model'] == 'K-Nearest Neighbors Regression':
    predicted_profit = predict_profit(knn_regressor, scaler, None, r_and_d_spend, ad
```

In [71]:

```python
#  predicted the profit value of the Company
print(f"\nPredicted Profit: ${predicted_profit:.2f}")
```

```
Predicted Profit: $1304876064.32
```