Chapter 6 - Solving Regression Problems with Machine Learning

6.1. Preparing Data for Regression Problems

```
import pandas as pd
import numpy as np
import seaborn as sns
sns.get_dataset_names()
```

```
['anagrams',
 'anscombe',
 'attention',
 'brain_networks',
 'car_crashes',
 'diamonds',
 'dots',
 'dowjones',
 'exercise',
 'flights',
 'fmri',
 'geyser',
 'glue',
 'healthexp',
 'iris',
 'mpg',
 'penguins',
 'planets',
 'seaice',
 'taxis',
 'tips',
 'titanic']
```

```
tips_df = sns.load_dataset("tips")
tips_df.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

```
diamond_df = sns.load_dataset("diamonds")
diamond_df.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

6.1.1. Dividing Data into Features and Labels

```
X = tips_df.drop(['tip'], axis=1)
y = tips_df["tip"]
```

```
X.head()
```

	total_bill	sex	smoker	day	time	size
0	16.99	Female	No	Sun	Dinner	2
1	10.34	Male	No	Sun	Dinner	3
2	21.01	Male	No	Sun	Dinner	3
3	23.68	Male	No	Sun	Dinner	2
4	24.59	Female	No	Sun	Dinner	4

```
y.head()
```

```
0    1.01
1    1.66
2    3.50
3    3.31
4    3.61
Name: tip, dtype: float64
```

6.1.2. Converting Categorical Data to Numbers

```
numerical = X.drop(['sex', 'smoker', 'day', 'time'], axis = 1)
```

```
numerical.head()
```

	total_bill	size
0	16.99	2
1	10.34	3
2	21.01	3
3	23.68	2
4	24.59	4

```
categorical = X.filter(['sex', 'smoker', 'day', 'time'])
categorical.head()
```

	sex	smoker	day	time
0	Female	No	Sun	Dinner
1	Male	No	Sun	Dinner
2	Male	No	Sun	Dinner
3	Male	No	Sun	Dinner
4	Female	No	Sun	Dinner

```
import pandas as pd
cat_numerical = pd.get_dummies(categorical,drop_first=True)
cat_numerical.head()
```

	sex_Female	smoker_No	day_Fri	day_Sat	day_Sun	time_Dinner
0	True	True	False	False	True	True
1	False	True	False	False	True	True
2	False	True	False	False	True	True
3	False	True	False	False	True	True
4	True	True	False	False	True	True

```
X = pd.concat([numerical, cat_numerical], axis = 1)
X.head()
```

	total_bill	size	sex_Female	smoker_No	day_Fri	day_Sat	day_Sun	time_Dinner
0	16.99	2	True	True	False	False	True	True
1	10.34	3	False	True	False	False	True	True
2	21.01	3	False	True	False	False	True	True
3	23.68	2	False	True	False	False	True	True
4	24.59	4	True	True	False	False	True	True

6.1.3. Divide Data into Training and Test Sets

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,  test_size=0.20, random_state=0)
```

6.1.4. Data Scaling/Normalization

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform (X_test)
```

6.2. Linear Regression

```
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()
regressor = lin_reg.fit(X_train, y_train)

y_pred = regressor.predict(X_test)


from sklearn import metrics

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
Mean Absolute Error: 0.7080218832979825
Mean Squared Error: 0.8939195221609609
Root Mean Squared Error: 0.9454731736865731
```