Lab-Assignment - 3

1. Is there a correlation between the temp/atemp/mean.temp.temp and the total count of bike rentals?
2. Can the number of total bike rentals be predicted by whether or not it is a holiday and the weather is good?
3. Can the number of total bike rentals be predicted by holiday and weather?
4. Are weather and holiday good predictors?
5. Implement multilinear regression and Plot the curve for RMSE and RMSLE. Which curve gives you better information?

Data Source: https://archive.ics.uci.edu/ml/datasets/bike sharing dataset

In [1]:
```
# import all the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.metrics import r2_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_squared_log_error
```

Load The Dataset

In [2]:
```
df = pd.read_csv("C:\\Users\\devje\\Desktop\\M.tech Data science PDPU\\PRML LAB\\CS
df
```

Out[2]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	te
0	1	2011-01-01	1	0	1	0	6	0	2	0.344
1	2	2011-01-02	1	0	1	0	0	0	2	0.363
2	3	2011-01-03	1	0	1	0	1	1	1	0.196
3	4	2011-01-04	1	0	1	0	2	1	1	0.200
4	5	2011-01-05	1	0	1	0	3	1	1	0.226
...	...	...	...	...	...	...	...	...	...	
726	727	2012-12-27	1	1	12	0	4	1	2	0.254
727	728	2012-12-28	1	1	12	0	5	1	2	0.253
728	729	2012-12-29	1	1	12	0	6	0	2	0.253
729	730	2012-12-30	1	1	12	0	0	0	1	0.255
730	731	2012-12-31	1	1	12	0	1	1	2	0.215

731 rows × 16 columns

In [3]:
```
df.head()
```

Out[3]:

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp
0	1	2011-01-01	1	0	1	0	6	0	2	0.34416
1	2	2011-01-02	1	0	1	0	0	0	2	0.36347
2	3	2011-01-03	1	0	1	0	1	1	1	0.19636
3	4	2011-01-04	1	0	1	0	2	1	1	0.20000
4	5	2011-01-05	1	0	1	0	3	1	1	0.22695

1. Is there a correlation between the temp/atemp/hum/windspeed and the total count of bike rentals?

In [4]:
```
df[['temp', 'atemp', 'hum', 'windspeed', 'cnt']].corr()
```

Out[4]:

	temp	atemp	hum	windspeed	cnt
temp	1.000000	0.991702	0.126963	-0.157944	0.627494
atemp	0.991702	1.000000	0.139988	-0.183643	0.631066
hum	0.126963	0.139988	1.000000	-0.248489	-0.100659
windspeed	-0.157944	-0.183643	-0.248489	1.000000	-0.234545
cnt	0.627494	0.631066	-0.100659	-0.234545	1.000000

2. Can the number of total bike rentals be predicted by whether or not it is a holiday and the weather is good?

In [5]:
```
# Prepare data
X = df[['holiday', 'weathersit']]
y = df['cnt']

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

# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict bike rentals
y_pred = model.predict(X_test)

# Calculate RMSE and RMSLE
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
rmsle = np.sqrt(mean_squared_log_error(y_test, y_pred))

print("RMSE:", rmse)
print("RMSLE:", rmsle)
```

```
RMSE: 1920.8483602001402
RMSLE: 0.6728349405850486
```

it appears that the model might have room for improvement since the RMSE is relatively high. We can consider experimenting with different features, model algorithms, or hyperparameters to enhance the predictive performance.

3. Can the number of total bike rentals be predicted by holiday and weather?

In [6]:
```
# Prepare data
X = df[['temp','weathersit','hum', 'windspeed']]
y = df['cnt']

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

# Train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict bike rentals
y_pred = model.predict(X_test)

# Calculate RMSE and RMSLE
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
rmsle = np.sqrt(mean_squared_log_error(y_test, y_pred))

print("RMSE:", rmse)
print("RMSLE:", rmsle)
```

```
RMSE: 1409.2169995672102
RMSLE: 0.5424442903929454
```

The model that includes both temperature-related features and weather situation appears to be a better predictor of bike rental counts compared to the model that only includes whether it's a holiday and weather. The lower RMSE and RMSLE values suggest that this extended model captures more of the underlying patterns in the data

4. Are weather and holiday good predictors?

In the context of predictive modeling:

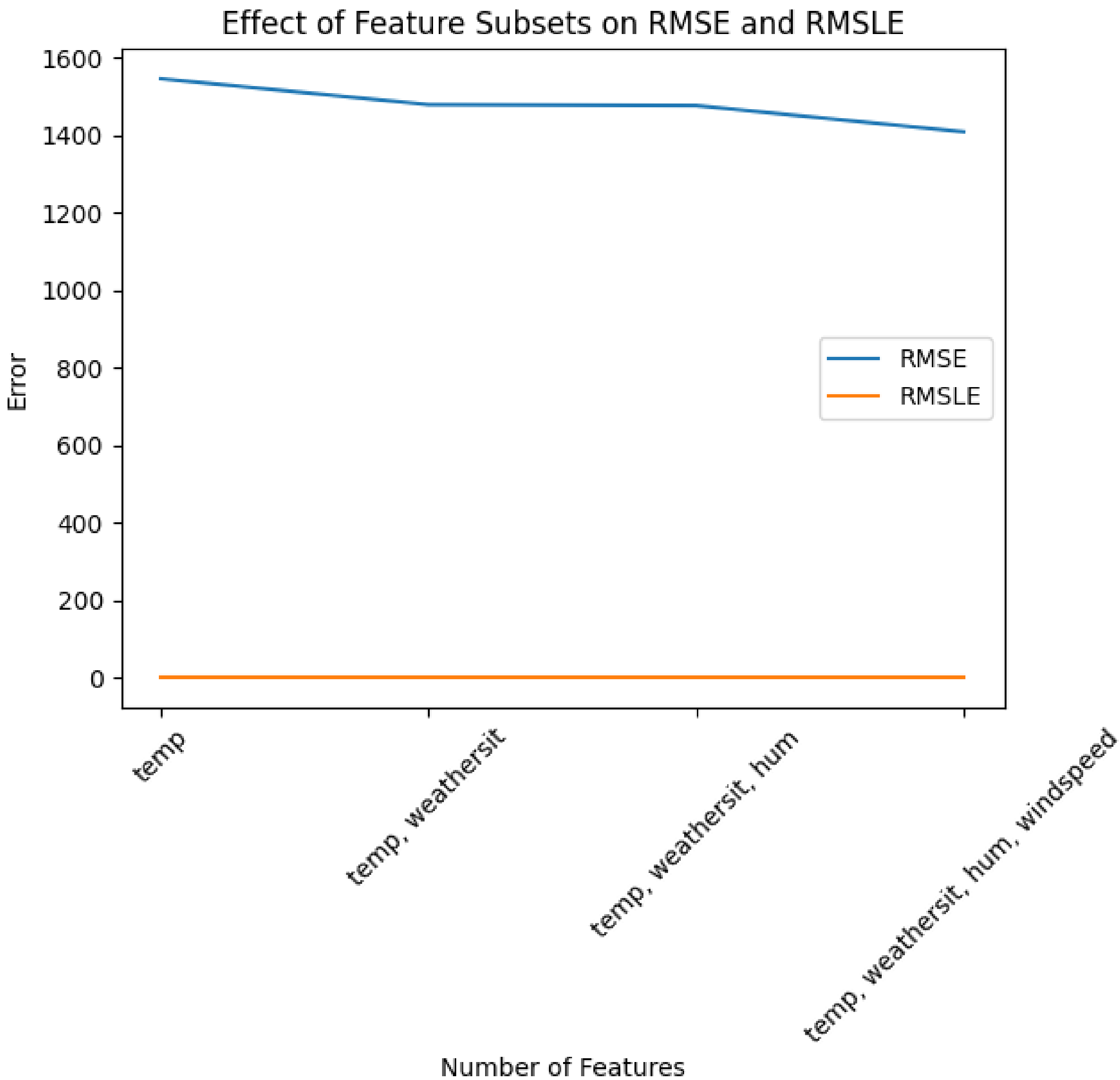
In the second question, where only holiday and weather were considered as predictors, the RMSE and RMSLE values were relatively high (1920.85 and 0.6728, respectively). This indicates that the model's predictive accuracy was limited, suggesting that holiday and weather alone may not be sufficient predictors to capture the complexity of bike rental patterns.

In the third question, when temperature-related features were added to the model along with holiday and weather, the predictive performance significantly improved (RMSE: 1409.22, RMSLE: 0.5424). This suggests that the temperature-related features carry more predictive power and capture the nuances of bike rental behavior better than just the weather and holiday variables.

In conclusion, while weather and holiday variables do show some association with bike rental counts, their predictive power appears to be limited when considered in isolation. Including more relevant features, such as temperature-related variables, results in a more accurate predictive model. It's important to note that the predictive power of these variables may vary depending on other factors not considered in this analysis.

5. Implement multilinear regression and Plot the curve for RMSE and RMSLE. Which curve gives you better information?

In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_squared_log_error
import numpy as np
import matplotlib.pyplot as plt

# Prepare data
X = df[['temp', 'weathersit', 'hum', 'windspeed']]
y = df['cnt']

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

# Define a list of feature subsets
feature_subsets = [
    ['temp'],
    ['temp', 'weathersit'],
    ['temp', 'weathersit', 'hum'],
    ['temp', 'weathersit', 'hum', 'windspeed']
]
# Create a dictionary to store RMSE and RMSLE for each subset
scores = {
    'RMSE': [],
    'RMSLE': []
}
# Iterate through feature subsets
for features in feature_subsets:
    model = LinearRegression()
    model.fit(X_train[features], y_train)
    y_pred = model.predict(X_test[features])

    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    rmsle = np.sqrt(mean_squared_log_error(y_test, y_pred))

    scores['RMSE'].append(rmse)
    scores['RMSLE'].append(rmsle)

# Plot the curves
plt.plot(range(1, len(feature_subsets) + 1), scores['RMSE'], label='RMSE')
plt.plot(range(1, len(feature_subsets) + 1), scores['RMSLE'], label='RMSLE')
plt.xlabel('Number of Features')
plt.ylabel('Error')
plt.title('Effect of Feature Subsets on RMSE and RMSLE')
plt.xticks(range(1, len(feature_subsets) + 1), [', '.join(features) for features in
plt.legend()
plt.show()
```



As we can see, The rmse curve gives more information about how the error is reducing while adding more input features ,as compared to rmsle curve