# Train a model with bike rental data using XGBoost algorithm

```
Model is trained with XGBoost installed in notebook instance
In the later examples, we will train using SageMaker's XGBoost algorithm
```

In [1]:

```
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, mean_absolute_error

# XGBoost
import xgboost as xgb
```

In [6]:

```
column_list_file = 'bike_train_column_list.txt'
train_file = r'C:\Users\309962\Desktop\xgBoost\Bike Rental\bike_train.csv'
validation_file = r'C:\Users\309962\Desktop\xgBoost\Bike Rental\bike_validation.c
test_file = r'C:\Users\309962\Desktop\xgBoost\Bike Rental\bike_test.csv'
```

In [7]:

```

columns = ''
with open(column_list_file,'r') as f:
    columns = f.read().split(',')
```

In [8]:

```
columns
```

Out[8]:

```
['count',
 'season',
 'holiday',
 'workingday',
 'weather',
 'temp',
 'atemp',
 'humidity',
 'windspeed',
 'year',
 'month',
 'day',
 'dayofweek',
 'hour']
```

In [9]:

```
# Specify the column names as the file does not have column header
df_train = pd.read_csv(train_file,names=columns)
df_validation = pd.read_csv(validation_file,names=columns)
```

In [10]:
```
df_train.head()
```

Out[10]:

	count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	month
0	87	3	0	0	2	26.24	30.305	73	7.0015	2011	9
1	248	3	0	1	1	32.80	34.850	33	7.0015	2012	8
2	334	4	0	0	1	15.58	19.695	40	11.0014	2011	11
3	623	3	0	1	1	32.80	37.880	55	12.9980	2012	8
4	70	2	0	1	1	13.94	17.425	76	7.0015	2011	4

In [11]:
```
df_validation.head()
```

Out[11]:

	count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	month
0	443	3	0	1	2	28.70	33.335	79	12.9980	2011	7
1	387	2	0	0	1	32.80	37.880	55	12.9980	2011	6
2	2	1	0	1	1	14.76	16.665	40	19.9995	2011	2
3	48	1	0	1	1	9.02	9.090	47	36.9974	2011	2
4	55	4	0	0	1	10.66	15.150	87	0.0000	2011	12

In [12]:
```
X_train = df_train.iloc[:,1:] # Features: 1st column onwards
y_train = df_train.iloc[:,0].ravel() # Target: 0th column

X_validation = df_validation.iloc[:,1:]
y_validation = df_validation.iloc[:,0].ravel()
```

In [13]:
```
# XGBoost Training Parameter Reference:
#   https://github.com/dmlc/xgboost/blob/master/doc/parameter.md
#regressor = xgb.XGBRegressor(max_depth=5,eta=0.1,subsample=0.7,num_round=150)
regressor = xgb.XGBRegressor(max_depth=5,n_estimators=150)
```

In [14]:
```
regressor
```

Out[14]:
```
XGBRegressor(base_score=None, booster=None, colsample_bylevel=None,
             colsample_bynode=None, colsample_bytree=None, gamma=None,
             gpu_id=None, importance_type='gain', interaction_constraints=None,
             learning_rate=None, max_delta_step=None, max_depth=5,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=150, n_jobs=None, num_parallel_tree=None,
             objective='reg:squarederror', random_state=None, reg_alpha=None,
             reg_lambda=None, scale_pos_weight=None, subsample=None,
             tree_method=None, validate_parameters=False, verbosity=None)
```

In [15]:
```
regressor.fit(X_train,y_train, eval_set = [(X_train, y_train), (X_validation, y_v
```

```
[0]	validation_0-rmse:200.21249	validation_1-rmse:198.50751
[1]	validation_0-rmse:158.44942	validation_1-rmse:156.82239
[2]	validation_0-rmse:130.70631	validation_1-rmse:129.74683
[3]	validation_0-rmse:113.91985	validation_1-rmse:113.36164
[4]	validation_0-rmse:97.49926	validation_1-rmse:97.96390
[5]	validation_0-rmse:84.68191	validation_1-rmse:86.42600
[6]	validation_0-rmse:75.20274	validation_1-rmse:77.72004
[7]	validation_0-rmse:71.41853	validation_1-rmse:74.25261
[8]	validation_0-rmse:64.23005	validation_1-rmse:67.80524
[9]	validation_0-rmse:61.87000	validation_1-rmse:65.64181
[10]	validation_0-rmse:60.00385	validation_1-rmse:63.93544
[11]	validation_0-rmse:57.38251	validation_1-rmse:61.66849
[12]	validation_0-rmse:55.40469	validation_1-rmse:59.70897
[13]	validation_0-rmse:53.46252	validation_1-rmse:58.07007
[14]	validation_0-rmse:50.16571	validation_1-rmse:55.00782
[15]	validation_0-rmse:49.58626	validation_1-rmse:54.48380
[16]	validation_0-rmse:49.10568	validation_1-rmse:53.95739
[17]	validation_0-rmse:46.31593	validation_1-rmse:51.42872
[18]	validation_0-rmse:45.25658	validation_1-rmse:50.72695
[19]	validation_0-rmse:44.66081	validation_1-rmse:50.27314
```

In [16]:
```
eval_result = regressor.evals_result()
```

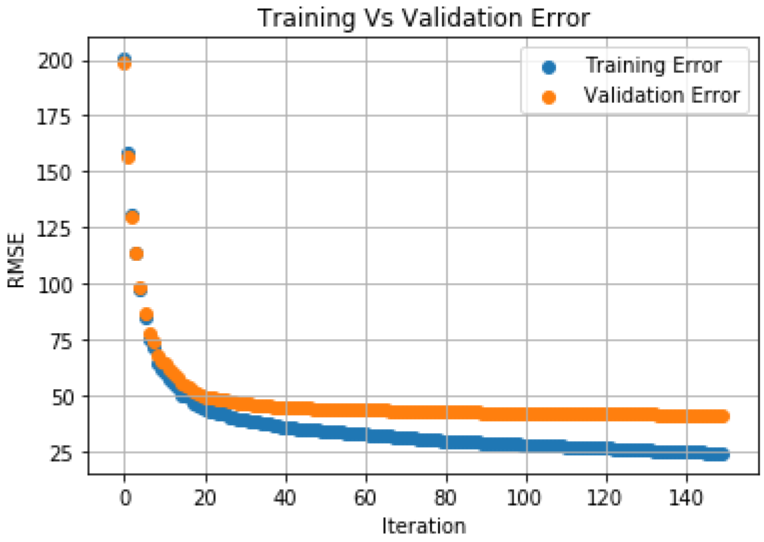
In [17]:
```
training_rounds = range(len(eval_result['validation_0']['rmse']))
```

In [18]:
```
print(training_rounds)
```

```
range(0, 150)
```

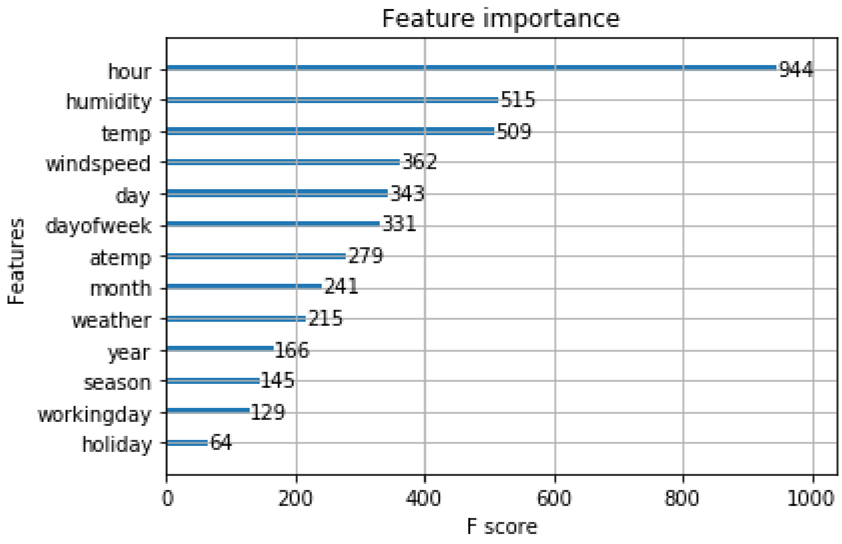
In [19]:

```
plt.scatter(x=training_rounds,y=eval_result['validation_0']['rmse'],label='Traini
plt.scatter(x=training_rounds,y=eval_result['validation_1']['rmse'],label='Valida
plt.grid(True)
plt.xlabel('Iteration')
plt.ylabel('RMSE')
plt.title('Training Vs Validation Error')
plt.legend()
plt.show()
```



In [20]:

```
xgb.plot_importance(regressor)
plt.show()
```



In [23]:

```
# Verify Quality using Validation dataset
# Compare actual vs predicted performance with dataset not seen by the model befo
df = pd.read_csv(validation_file,names=columns)
```

In [24]:

```
df.head()
```

Out[24]:

	count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	month
0	443	3	0	1	2	28.70	33.335	79	12.9980	2011	7
1	387	2	0	0	1	32.80	37.880	55	12.9980	2011	6
2	2	1	0	1	1	14.76	16.665	40	19.9995	2011	2
3	48	1	0	1	1	9.02	9.090	47	36.9974	2011	2
4	55	4	0	0	1	10.66	15.150	87	0.0000	2011	12

In [25]:

```
df.shape
```

Out[25]:

```
(3266, 14)
```

In [26]:

```
X_test = df.iloc[:,1:]
print(X_test[:5])
```

```
   season  holiday  workingday  weather   temp   atemp  humidity  windspeed  \
0       3        0           1        2  28.70  33.335        79    12.9980
1       2        0           0        1  32.80  37.880        55    12.9980
2       1        0           1        1  14.76  16.665        40    19.9995
3       1        0           1        1   9.02   9.090        47    36.9974
4       4        0           0        1  10.66  15.150        87     0.0000

   year  month  day  dayofweek  hour
0  2011      7    7          3     8
1  2011      6   11          5    13
2  2011      2   14          0     2
3  2011      2    8          1    10
4  2011     12    4          6     8
```

In [27]:

```
result = regressor.predict(X_test)
```

In [28]:

```
result[:5]
```

Out[28]:

```
array([452.154     , 373.7294    ,   0.75503814,  64.58523   ,
        83.32642   ], dtype=float32)
```

In [29]:

```
df['count_predicted'] = result
```

In [30]:

```
df.head()
```

Out[30]:

	count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	month
0	443	3	0	1	2	28.70	33.335	79	12.9980	2011	7
1	387	2	0	0	1	32.80	37.880	55	12.9980	2011	6
2	2	1	0	1	1	14.76	16.665	40	19.9995	2011	2
3	48	1	0	1	1	9.02	9.090	47	36.9974	2011	2
4	55	4	0	0	1	10.66	15.150	87	0.0000	2011	12

In [31]:

```
# Negative Values are predicted
df['count_predicted'].describe()
```

Out[31]:

```
count    3266.000000
mean      190.070694
std       174.655899
min       -95.306847
25%        43.720430
50%       150.537590
75%       284.134521
max       901.711853
Name: count_predicted, dtype: float64
```

In [32]:

```
df[df['count_predicted'] < 0]
```

Out[32]:

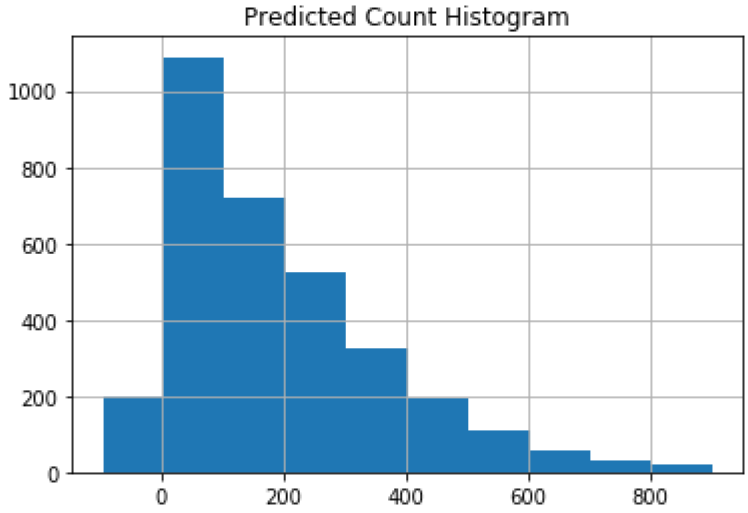
	count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	mo
99	71	2	0	1	3	22.96	26.515	88	7.0015	2012	
103	11	3	0	1	2	27.88	31.820	83	12.9980	2012	
117	2	4	0	1	1	8.20	12.880	80	0.0000	2011	
137	9	1	0	1	1	15.58	19.695	54	7.0015	2012	
158	45	1	0	0	2	12.30	13.635	100	19.9995	2011	
159	16	2	0	1	1	18.86	22.725	41	8.9981	2012	
174	4	4	0	1	2	18.04	21.970	100	15.0013	2011	
210	48	2	0	1	2	26.24	31.060	47	15.0013	2011	
236	2	1	0	1	2	13.12	16.665	93	6.0032	2011	
250	5	4	0	1	2	15.58	19.695	94	0.0000	2012	
373	13	4	1	0	1	12.30	14.395	49	16.9979	2011	
472	6	3	0	0	1	31.98	37.120	62	0.0000	2012	
490	6	3	0	0	2	26.24	28.030	94	7.0015	2011	
500	8	1	1	0	1	4.92	7.575	58	7.0015	2012	
522	133	2	1	0	1	21.32	25.000	83	11.0014	2012	
523	3	2	0	0	3	12.30	13.635	100	16.9979	2011	
526	7	4	0	0	1	9.84	11.365	70	12.9980	2012	
556	8	4	0	0	1	11.48	12.880	56	22.0028	2011	
581	3	1	0	1	2	6.56	7.575	55	12.9980	2011	
589	5	2	0	1	1	20.50	24.240	77	12.9980	2012	
606	1	1	0	1	1	8.20	11.365	55	6.0032	2012	
609	8	1	0	0	1	9.02	11.365	47	11.0014	2011	
659	5	2	0	0	2	12.30	14.395	93	15.0013	2011	
683	62	3	0	1	3	27.88	31.820	69	12.9980	2011	
684	8	3	0	1	2	24.60	28.030	83	11.0014	2011	
733	3	2	0	1	1	18.04	21.970	30	11.0014	2012	
742	1	1	0	0	1	10.66	12.880	60	15.0013	2011	
747	2	3	0	1	3	27.06	29.545	89	26.0027	2012	
759	18	3	0	0	1	26.24	30.305	73	7.0015	2011	
790	6	2	0	1	3	23.78	27.275	83	8.9981	2012	
...	...	...	...	...	...	...	...	...	...	...	
2478	2	1	0	1	2	6.56	11.365	69	0.0000	2011	
2492	4	3	0	1	1	28.70	33.335	79	11.0014	2011	
2616	4	3	0	1	1	16.40	20.455	71	19.0012	2011	

	count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	mo
2643	4	4	0	1	1	6.56	9.090	86	8.9981	2011	
2670	7	4	0	0	1	8.20	10.605	80	8.9981	2011	
2684	8	2	0	0	1	20.50	24.240	64	0.0000	2011	
2697	7	3	0	1	2	27.88	31.820	89	0.0000	2011	
2698	13	4	0	1	3	14.76	17.425	87	15.0013	2012	
2746	69	4	0	1	3	22.96	26.515	100	8.9981	2011	
2755	7	1	0	1	2	6.56	11.365	59	0.0000	2011	
2770	6	4	0	1	2	12.30	14.395	49	15.0013	2012	
2836	6	1	0	1	2	6.56	9.090	86	7.0015	2011	
2883	4	3	0	1	1	27.06	30.305	83	8.9981	2011	
2887	3	4	0	1	2	13.12	16.665	70	8.9981	2011	
2893	10	1	0	1	1	15.58	19.695	50	8.9981	2012	
2926	4	1	0	1	1	9.02	10.605	80	19.0012	2012	
2928	3	2	0	0	1	11.48	15.150	70	6.0032	2011	
2997	2	1	0	0	1	4.92	6.820	93	12.9980	2011	
3004	2	1	0	1	1	13.94	15.910	46	15.0013	2011	
3045	2	1	0	1	3	14.76	17.425	93	16.9979	2012	
3051	3	1	0	1	2	12.30	16.665	70	0.0000	2012	
3066	3	1	0	1	1	4.92	6.060	50	15.0013	2011	
3071	6	1	0	0	1	3.28	4.545	53	12.9980	2011	
3093	4	1	0	1	1	5.74	6.820	46	12.9980	2012	
3121	1	1	0	1	1	5.74	7.575	86	8.9981	2011	
3129	44	1	0	1	3	13.12	16.665	70	8.9981	2012	
3176	16	4	0	1	2	12.30	14.395	52	16.9979	2012	
3199	8	4	0	1	1	13.94	15.910	81	15.0013	2011	
3252	11	3	0	0	1	25.42	30.305	61	0.0000	2011	
3259	4	1	0	1	1	8.20	12.880	61	0.0000	2012	

127 rows × 15 columns

In [33]:

```
df['count_predicted'].hist()
plt.title('Predicted Count Histogram')
plt.show()
```



In [34]:

```
def adjust_count(x):
    if x < 0:
        return 0
    else:
        return x
```

In [35]:

```

df['count_predicted'] = df['count_predicted'].map(adjust_count)
```

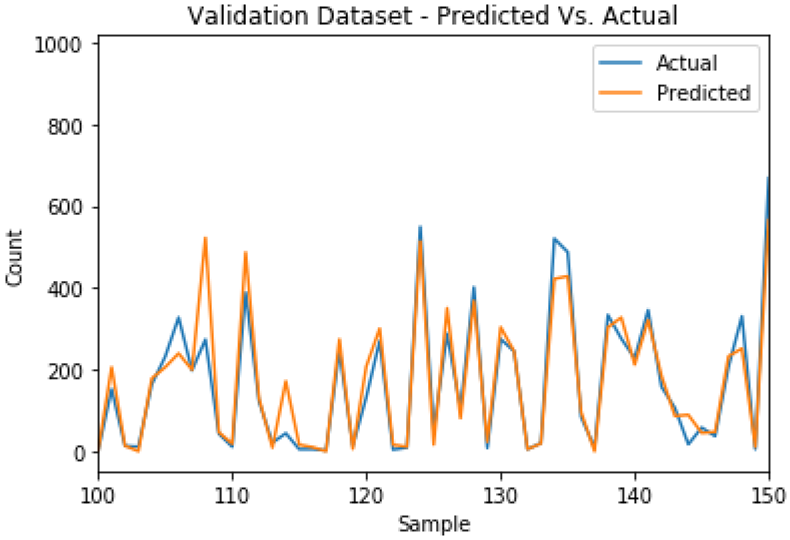
In [36]:

```
df[df['count_predicted'] < 0]
```

Out[36]:

count	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	month

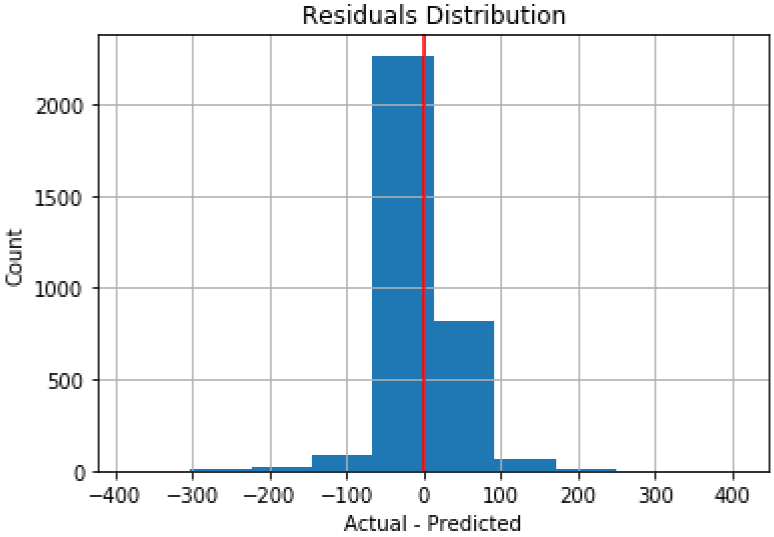
In [37]:
```
# Actual Vs Predicted
plt.plot(df['count'], label='Actual')
plt.plot(df['count_predicted'],label='Predicted')
plt.xlabel('Sample')
plt.ylabel('Count')
plt.xlim([100,150])
plt.title('Validation Dataset - Predicted Vs. Actual')
plt.legend()
plt.show()
```



In [38]:

```
# Over prediction and Under Prediction needs to be balanced
# Training Data Residuals
residuals = (df['count'] - df['count_predicted'])

plt.hist(residuals)
plt.grid(True)
plt.xlabel('Actual - Predicted')
plt.ylabel('Count')
plt.title('Residuals Distribution')
plt.axvline(color='r')
plt.show()
```



In [39]:

```

value_counts = (residuals > 0).value_counts(sort=False)
print(' Under Estimation: {0:0.2f}'.format(value_counts[True]/len(residuals)))
print(' Over  Estimation: {0:0.2f}'.format(value_counts[False]/len(residuals)))
```

```
 Under Estimation: 0.50
 Over  Estimation: 0.50
```

In [40]:

```
print("RMSE: {0:0.2f}".format(mean_squared_error(df['count'],df['count_predicted'
```

```
RMSE: 40.89
```

In [41]:

```
# RMSlE - Root Mean Squared Log Error
# RMSLE Metric is used by Kaggle for this competition

# RMSE Cost Function - Magnitude of difference matters

# RMSLE cost function - "Only Percentage difference matters"

# Reference:Katerina Malahova, Khor SoonHin
# https://www.slideshare.net/KhorSoonHin/rmsle-cost-function
def compute_rmsle(y_true, y_pred):
    if type(y_true) != np.ndarray:
        y_true = np.array(y_true)

    if type(y_pred) != np.ndarray:
        y_pred = np.array(y_pred)

    return(np.average((np.log1p(y_pred) - np.log1p(y_true))**2)**.5)
```

In [42]:

```
print('RMSLE')
print(compute_rmsle(100,50),
      compute_rmsle(1000,500),
      compute_rmsle(10000,5000))
```

```
RMSLE
0.683294884116934 0.6921486782303559 0.6930471955576127
```

In [43]:

```
print('RMSLE')
print(compute_rmsle(100,25),
      compute_rmsle(1000,250),
      compute_rmsle(10000,2500))
```

```
RMSLE
1.3570239788197775 1.383301840183437 1.3859944360988976
```

In [44]:

```
print('RMSE')
print(mean_squared_error([100],[50])**.5,
      mean_squared_error([1000],[500])**.5,
      mean_squared_error([10000],[5000])**.5)
```

```
RMSE
50.0 500.0 5000.0
```

In [45]:

```
print('RMSE')
print(mean_squared_error([100],[25])**.5,
      mean_squared_error([1000],[250])**.5,
      mean_squared_error([10000],[2500])**.5)
```

```
RMSE
75.0 750.0 7500.0
```

In [46]:
```
print("RMSLE: {0}".format(compute_rmsle(df['count'],df['count_predicted'])))
```

```
RMSLE: 0.5999730528617999
```

In [47]:
```
# Prepare Data for Submission to Kaggle
df_test = pd.read_csv(test_file,parse_dates=['datetime'])
```

In [48]:
```
df_test.head()
```

Out[48]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	mo
0	2011-01-20 00:00:00	1	0	1	1	10.66	11.365	56	26.0027	2011	
1	2011-01-20 01:00:00	1	0	1	1	10.66	13.635	56	0.0000	2011	
2	2011-01-20 02:00:00	1	0	1	1	10.66	13.635	56	0.0000	2011	
3	2011-01-20 03:00:00	1	0	1	1	10.66	12.880	56	11.0014	2011	
4	2011-01-20 04:00:00	1	0	1	1	10.66	12.880	56	11.0014	2011	

In [49]:
```
X_test =  df_test.iloc[:,1:] # Exclude datetime for prediction
```

In [50]:
```
X_test.head()
```

Out[50]:

	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	month	day
0	1	0	1	1	10.66	11.365	56	26.0027	2011	1	20
1	1	0	1	1	10.66	13.635	56	0.0000	2011	1	20
2	1	0	1	1	10.66	13.635	56	0.0000	2011	1	20
3	1	0	1	1	10.66	12.880	56	11.0014	2011	1	20
4	1	0	1	1	10.66	12.880	56	11.0014	2011	1	20

In [51]:
```
result = regressor.predict(X_test)
```

In [52]:
```
result[:5]
```

Out[52]:
```
array([ 12.3928995,  -3.7081814, -10.777084 ,  -4.4427557,  -4.4427557],
      dtype=float32)
```

In [53]:
```
df_test["count"] = result
```

In [54]:
```
df_test.head()
```

Out[54]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year	mo
0	2011-01-20 00:00:00	1	0	1	1	10.66	11.365	56	26.0027	2011	
1	2011-01-20 01:00:00	1	0	1	1	10.66	13.635	56	0.0000	2011	
2	2011-01-20 02:00:00	1	0	1	1	10.66	13.635	56	0.0000	2011	
3	2011-01-20 03:00:00	1	0	1	1	10.66	12.880	56	11.0014	2011	
4	2011-01-20 04:00:00	1	0	1	1	10.66	12.880	56	11.0014	2011	

In [55]:
```
df_test[df_test["count"] < 0]
```

Out[55]:

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year
1	2011-01-20 01:00:00	1	0	1	1	10.66	13.635	56	0.0000	2011
2	2011-01-20 02:00:00	1	0	1	1	10.66	13.635	56	0.0000	2011
3	2011-01-20 03:00:00	1	0	1	1	10.66	12.880	56	11.0014	2011
4	2011-01-20 04:00:00	1	0	1	1	10.66	12.880	56	11.0014	2011
25	2011-01-21 01:00:00	1	0	1	2	9.84	11.365	70	16.9979	2011
27	2011-01-21 03:00:00	1	0	1	3	9.02	10.605	80	19.9995	2011
28	2011-01-21 04:00:00	1	0	1	2	9.02	12.880	87	6.0032	2011
52	2011-01-22 04:00:00	1	0	0	2	0.82	0.760	48	19.9995	2011
53	2011-01-22 06:00:00	1	0	0	2	0.82	1.515	44	15.0013	2011
54	2011-01-22 07:00:00	1	0	0	1	0.82	0.760	44	19.0012	2011
73	2011-01-23 02:00:00	1	0	0	1	0.82	3.030	62	8.9981	2011
74	2011-01-23 03:00:00	1	0	0	1	0.82	3.030	62	8.9981	2011
77	2011-01-23 07:00:00	1	0	0	1	3.28	5.305	58	11.0014	2011
89	2011-01-23 19:00:00	1	0	0	1	4.92	6.060	30	19.0012	2011
97	2011-01-24 04:00:00	1	0	1	1	0.82	3.030	48	8.9981	2011
98	2011-01-24 05:00:00	1	0	1	1	0.82	3.030	48	8.9981	2011

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year
99	2011-01-24 06:00:00	1	0	1	1	0.82	3.790	48	6.0032	2011
117	2011-01-25 00:00:00	1	0	1	2	6.56	6.820	69	19.0012	2011
118	2011-01-25 01:00:00	1	0	1	2	6.56	8.335	69	11.0014	2011
119	2011-01-25 02:00:00	1	0	1	1	6.56	7.575	69	15.0013	2011
120	2011-01-25 04:00:00	1	0	1	1	5.74	8.335	74	7.0015	2011
141	2011-01-26 01:00:00	1	0	1	2	9.84	12.120	65	8.9981	2011
153	2011-01-26 15:00:00	1	0	1	3	9.02	9.090	93	31.0009	2011
191	2011-01-29 04:00:00	1	0	0	1	6.56	9.090	69	7.0015	2011
192	2011-01-29 06:00:00	1	0	0	1	6.56	9.090	64	8.9981	2011
193	2011-01-29 07:00:00	1	0	0	1	6.56	9.090	59	7.0015	2011
212	2011-01-30 02:00:00	1	0	0	1	6.56	11.365	80	0.0000	2011
213	2011-01-30 03:00:00	1	0	0	1	5.74	10.605	93	0.0000	2011
214	2011-01-30 04:00:00	1	0	0	1	5.74	10.605	93	0.0000	2011
215	2011-01-30 05:00:00	1	0	0	1	5.74	10.605	86	0.0000	2011
...	...	...	...	...	...	...	...	...	...	...
5290	2012-08-26 06:00:00	3	0	0	1	25.42	28.030	88	19.9995	2012
5310	2012-08-27 02:00:00	3	0	1	1	25.42	28.030	88	8.9981	2012
5311	2012-08-27 03:00:00	3	0	1	1	25.42	28.030	88	6.0032	2012

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year
5312	2012-08-27 04:00:00	3	0	1	1	25.42	28.790	83	0.0000	2012
5334	2012-08-28 02:00:00	3	0	1	1	27.88	31.820	83	16.9979	2012
5336	2012-08-28 04:00:00	3	0	1	1	27.06	30.305	83	8.9981	2012
5337	2012-08-28 05:00:00	3	0	1	2	27.06	30.305	83	6.0032	2012
5481	2012-09-22 05:00:00	3	0	0	1	22.96	26.515	83	22.0028	2012
5621	2012-09-28 01:00:00	4	0	1	3	24.60	27.275	88	0.0000	2012
5622	2012-09-28 02:00:00	4	0	1	3	24.60	27.275	88	19.9995	2012
5673	2012-09-30 05:00:00	4	0	0	1	18.04	21.970	72	0.0000	2012
5902	2012-10-28 18:00:00	4	0	0	3	17.22	21.210	94	32.9975	2012
5905	2012-10-28 21:00:00	4	0	0	3	18.04	21.970	88	27.9993	2012
5906	2012-10-28 22:00:00	4	0	0	3	18.04	21.970	88	23.9994	2012
5907	2012-10-28 23:00:00	4	0	0	3	17.22	21.210	94	23.9994	2012
6019	2012-11-23 03:00:00	4	0	1	1	10.66	15.150	70	0.0000	2012
6092	2012-11-26 04:00:00	4	0	1	1	9.02	13.635	69	0.0000	2012
6115	2012-11-27 03:00:00	4	0	1	3	13.12	16.665	70	8.9981	2012
6116	2012-11-27 04:00:00	4	0	1	3	12.30	14.395	81	12.9980	2012
6138	2012-11-28 02:00:00	4	0	1	2	10.66	12.880	75	15.0013	2012

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed	year
6139	2012-11-28 03:00:00	4	0	1	2	10.66	12.880	70	15.0013	2012
6140	2012-11-28 04:00:00	4	0	1	2	10.66	12.880	70	12.9980	2012
6185	2012-11-30 02:00:00	4	0	1	1	9.84	14.395	75	0.0000	2012
6186	2012-11-30 03:00:00	4	0	1	1	9.84	14.395	75	0.0000	2012
6187	2012-11-30 04:00:00	4	0	1	1	9.02	13.635	75	0.0000	2012
6210	2012-12-20 03:00:00	4	0	1	2	12.30	15.910	70	6.0032	2012
6211	2012-12-20 04:00:00	4	0	1	2	12.30	15.910	70	6.0032	2012
6235	2012-12-21 04:00:00	1	0	1	2	14.76	15.910	71	32.9975	2012
6284	2012-12-23 05:00:00	1	0	0	1	8.20	12.880	51	0.0000	2012
6451	2012-12-30 06:00:00	1	0	0	2	9.84	9.850	52	27.9993	2012

285 rows × 15 columns

In [56]:

```
df_test["count"] = df_test["count"].map(adjust_count)
```

In [57]:

```
df_test[['datetime','count']].to_csv('predicted_count.csv',index=False)
```

In [58]:

```
# RMSLE (Kaggle) Score
# Test 1: 0.62
```

In [ ]: