Predict FPPG on FanDuel: RMSE Regression, Aug. 2016

loadData

In [1]:

```
import graphlab
```

```
A newer version of GraphLab Create (v2.1) is available! Your current version is v2.0.1.
You can use pip to upgrade the graphlab-create package. For more information see https://turi.com/products/create/upgrade.
```

In [3]:

```
df = graphlab.SFrame('nfl.csv')
```

```
[INFO] graphlab.cython.cy_server: GraphLab Create v2.0.1 started. Logging: /tmp/graphlab_server_1470939727.log
INFO:graphlab.cython.cy_server:GraphLab Create v2.0.1 started. Logging: /tmp/graphlab_server_1470939727.log
Finished parsing file /home/ubuntu/nfl.csv

Parsing completed. Parsed 27 lines in 0.018136 secs.

This non-commercial license of GraphLab Create for academic use is assigned to crucker@mediacomcc.com and will expire on October 22, 2016.
------------------------------------------------------

Read 27 lines. Lines per second: 1918.29

Finished parsing file /home/ubuntu/nfl.csv

Parsing completed. Parsed 27 lines in 0.016764 secs.

Inferred types from first 100 line(s) of file as
column_type_hints=[str,str,float,int,float,int,int,int,int]
If parsing fails due to incorrect types, you can correct
the inferred type list above and pass it to read_csv in
the column_type_hints argument
------------------------------------------------------
```

In [4]:

```
df.head(1)
```

Out[4]:

Position	Player	AvgPointsPerGame	Salary	SalaryPerPointsPerGame	Yards	Touchdowns	Turnovers	Played
QB	A Rodgers	19.33	9000	465.5975168	3821	31	8	16

[1 rows x 9 columns]

buildModel

In [5]:

```
train_data, test_data = df.random_split(.8, seed=0)
```

In [11]:

```
reg_model = graphlab.linear_regression.create(train_data, target='AvgPointsPerGame', features=['Salary', 'SalaryPerPointsPerGame',
                                                                                                'Yards', 'Touchdowns', 'Turnovers',
                                                                                                'Played'])
```

```
Linear regression:
--------------------------------------------------------
Number of examples          : 24
Number of features          : 6
Number of unpacked features : 6
Number of coefficients    : 7
Starting Newton Method
--------------------------------------------------------
```

```
+-----------+----------+--------------+--------------------+---------------+
| Iteration | Passes   | Elapsed Time | Training-max_error | Training-rmse |
+-----------+----------+--------------+--------------------+---------------+
| 1         | 2        | 0.000129     | 1.455699           | 0.609566      |
+-----------+----------+--------------+--------------------+---------------+
SUCCESS: Optimal solution found.
```

In [12]:

```
reg_model.get('coefficients').print_rows(num_rows=18, num_columns=3)
```

```
+------------------------+-------+--------------------+-----+
|          name          | index |       value        | ... |
+------------------------+-------+--------------------+-----+
|      (intercept)       |  None |   12.6612906586    | ... |
|         Salary         |  None |  0.00170443363477  | ... |
| SalaryPerPointsPerGame |  None |  -0.018637542677   | ... |
|         Yards          |  None | 0.000586796024585  | ... |
|       Touchdowns       |  None |  0.0201662378571   | ... |
|       Turnovers        |  None |  0.00542014954634  | ... |
|         Played         |  None |  -0.134998408521   | ... |
+------------------------+-------+--------------------+-----+
[7 rows x 4 columns]
```

In [13]:

```
print reg_model.evaluate(test_data)
```

```
{'max_error': 0.8141961327413227, 'rmse': 0.521615273220494}
```

applyModel

In [14]:

```
def AvgPointsPerGame(x):
    AvgPointsPerGame = df[df['Player']==x]
    return reg_model.predict(AvgPointsPerGame)
```

In [15]:

```
AvgPointsPerGame('A Rodgers')
```

Out[15]:

```
dtype: float
Rows: 1
[20.074287425370894]
```

writeUp

This is a Root Mean Square Error (RMSE) regression model that results in predicted values close to the observed data. The RMSE value in the model results is an absolute measure of fit. One pitfall of RMSE is that it can incorporate too many variables, however this regression model has a Validation-rmse of 0.609566 and lower values indicate a better fit.