COLLEGE OF ENGINEERING

Electrical Engineering and Computer Science

FULCRUM NFL

INSPIRATION

Fantasy football was said to be created in 1962 by a minority owner of the Oakland Raiders and some colleagues. In 1985, the first online sports service was made avialable through Q-Link (America Online). The game has seen tremendous growth. In 2017 alone, fantasy football generated an estimated $7 billion in revenue. Online daily fantasy sports leagues, such as FanDuel and DraftKings, have emerged that allow users to wager on fantasy scoring on a weekly basis and these companies now sport billion dollar valuations.

Picking fantasy players and teams is largely an emotional decision for most. Many pick hometown favorites or a fellow player from their alma mater. As three fantasy sports enthusiasts and stat junkies, our goal was to create a predictive model strictly on hard data to provide an edge over the crowd.

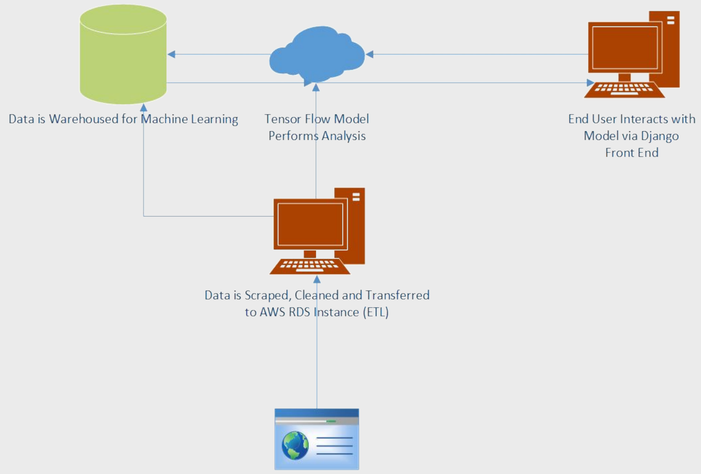
DESCRIPTION

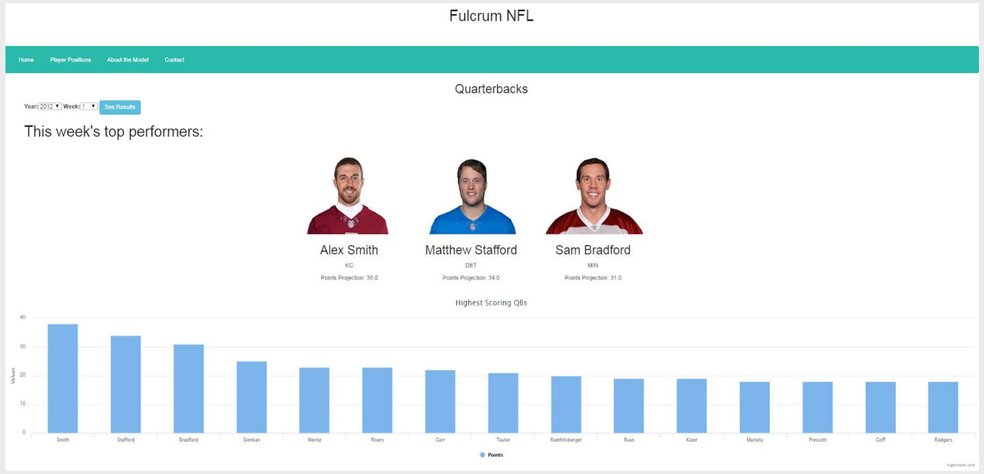
Fulcrum uses a client-server model. An AWS instance provides the server capabilities, providing the Postgres RDBMS and the Linux virtual machine used to host the machine learning model. 5 years of fantasy football statistics for various positions were collected using a custom made web scraper. The data was then scrubbed, formatted, and normalized before being sent to the model. A two layer neural network model was created using the Tensorflow library. The results of the regression model are placed back into the database and then retrieved by the front-end website to be displayed to users. The website contains all of the data the user needs to make informed fantasy football decisions, including predicted scores and historical graphs of the model's performance.

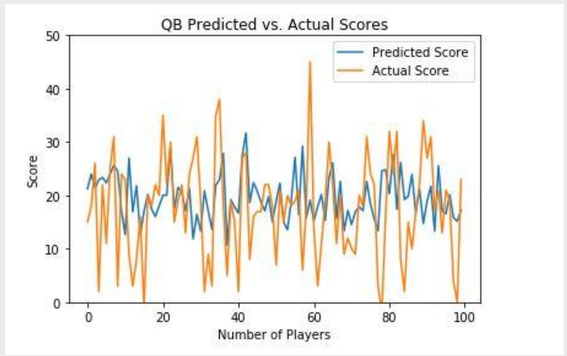
Oregon State University

MACHINE LEARNING FOR FANTASY FOOTBALL

Multi-layer neural network predicts weekly player scores







Model Performance for quaterbacks over the last 100 games

```
def scrape(pos1, yr, wk, filename):
    url = 'http://www.footballdb.com/fantasy-football/index.html?pos=' + pos1 + '&yr=' + yr + '&wk=' + wk + '&rules=1'
    headers = {
    'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/67.0.3396.99 Safari/537.36'
    }
    r = requests.get(url, headers=headers)
    data = r.text
    soup = BeautifulSoup(data, 'html')

    list = []

    table = soup.find_all('table')

    for row in table:
        col = row.find_all('td')
        col = [ele.text.strip() for ele in col]
        list.append([ele for ele in col if ele])
    y = 0
    name = ""
    pos = 0
    size = len(list[0])
    i=0
    file = open(filename, "a")
```

Snapshot of the web scraper code

FEATURES

- Web scraper, built using the Beautiful Soup library, collects data from sports statistics websites
- Postgres database on AWS contains over 10,000 rows of scraped data. covering a myriad of football statistics from 2013-2017
- Runs on an Amazon Web Service's Ubuntu EC2 instance that is optimized for machine learning
- Datastack contains 26 features, including defensive rank of the opponent and if the game is home or away
- Utilizes TensorFlow's powerful machine learning library to create a neural network with two hidden layers and 50 neurons per layer
- Predicted scores returned from the model are paired with the appropriate player and inserted back into the database using SQLAlchemy
- Front-end website, created using Django, retrieves the data and provides users with easy access to the score predictions in a clear visual format.
- Graphs on the website, via the HighCharts.js package, provide an in-depth look at the historical performance of the model

```
# Split into training and test sets
train_samples = int(0.8*samples_count)
test_samples = int(samples_count - train_samples)

train_inputs = scaled_inputs[:train_samples]
train_targets = pts_target[:train_samples]

test_inputs = scaled_inputs[train_samples:]
test_targets = pts_target[train_samples:]
```

```
print("Training set: {}".format(train_inputs.shape))  # 1052 examples, 26 features
print("Testing set:  {}".format(test_inputs.shape))   # 263 examples, 26 features
```

```
# Create 2 Layer Neural Network
def build_model():
  model = keras.Sequential([
    keras.layers.Dense(64, activation=tf.nn.relu,
                       input_shape=(train_inputs.shape[1],)),
    keras.layers.Dense(64, activation=tf.nn.relu),
    keras.layers.Dense(1)
  ])

  # AdamOptimizer
  optimizer = tf.train.AdamOptimizer(0.001)

  # Reduce mean squared loss
  model.compile(loss='mse',
                optimizer=optimizer,
                metrics=['mae'])
  return model

model = build_model()
model.summary()
```

Snapshot of the model code. The data is being split into training and test sets and the neural network is being built.

FULCRUM NFL DEVELOPED BY:
Eric Durboraw James Meehan Anthony Quach

Github: https://github.com/jmeehan2003/MachineLearning-FantasyFootball **Hosted at:** http://cepheus.pythonanywhere.com/