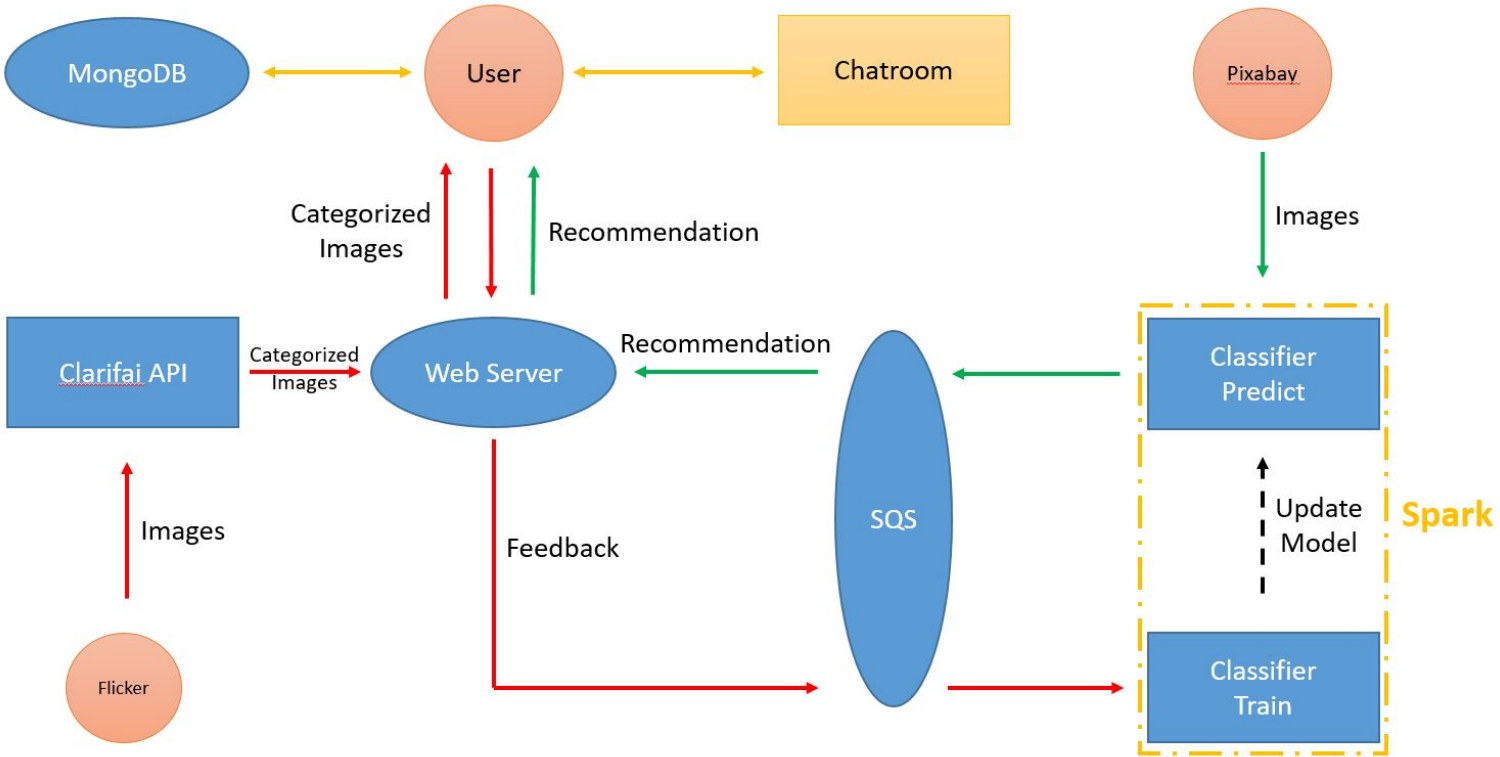
LightHouse

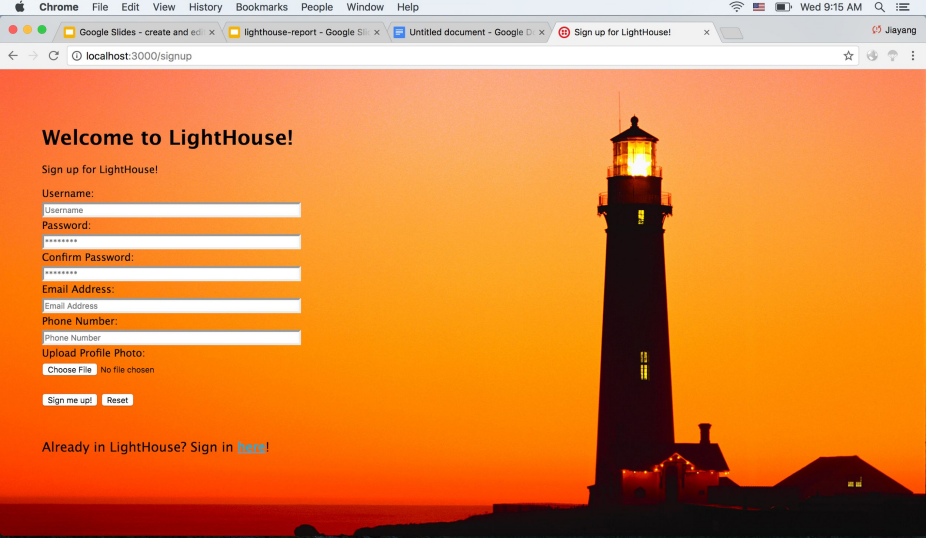
A smart way to explore the beauty of the world ♥

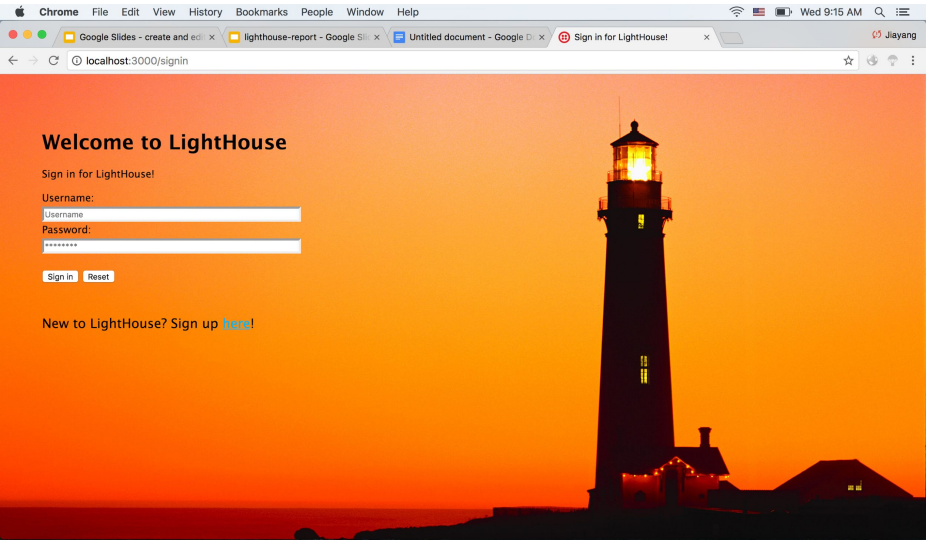
Application Infrastructure

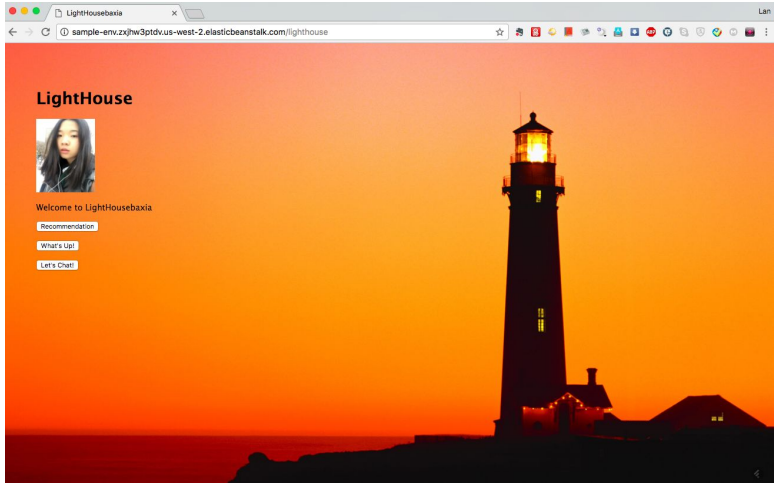


Sign in / Sign up

We utilize MongoDB to store users' information, such as his/ her username, password and email address. We also use express session to manage users' login sessions to improve security issues.

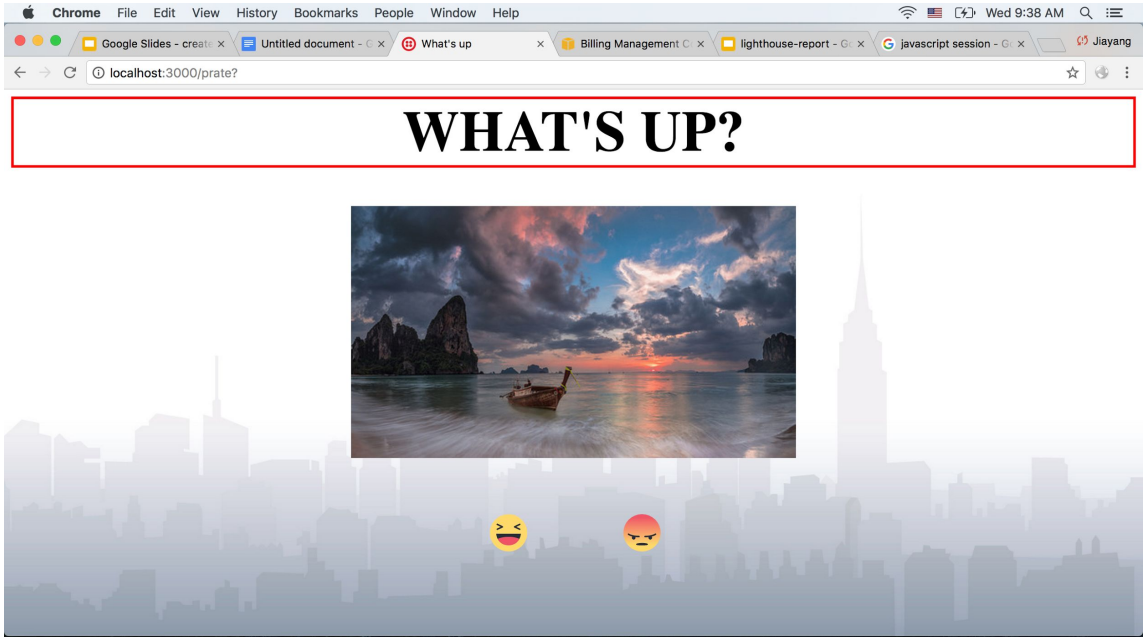






Machine Learning in LightHouse

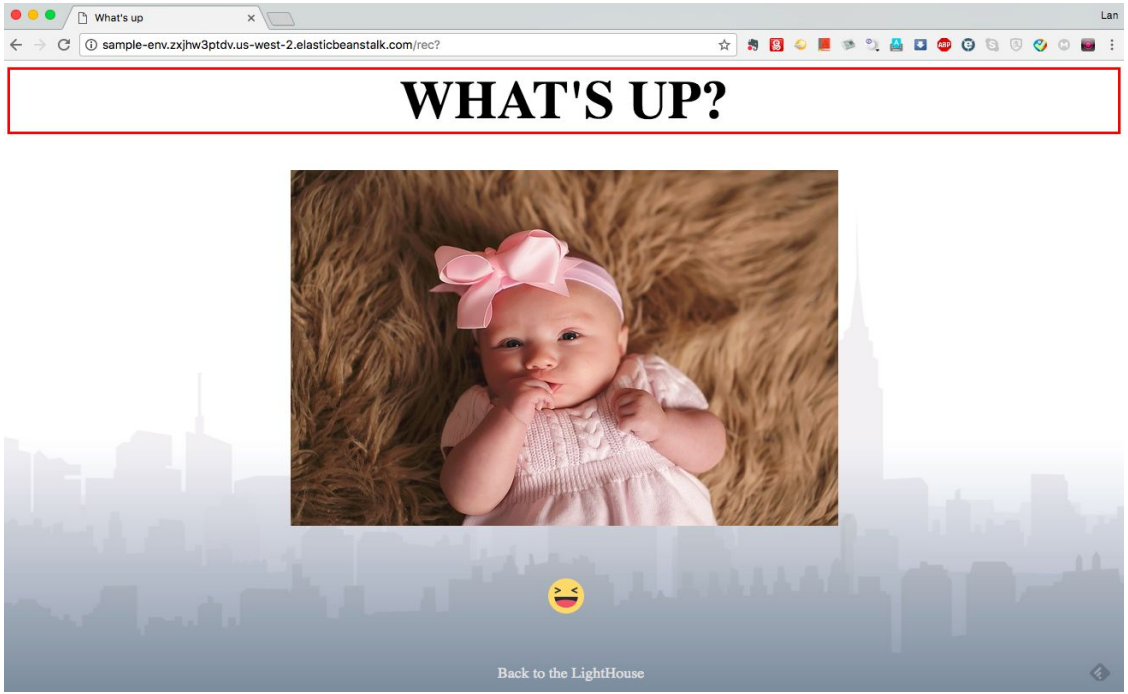
--Training model



We employ Flickr API to get some “interesting photos” randomly and present them to users. We then ask the users to tell us how they feel about them(like/ dislike) and try to learn their preference.

Machine Learning in LightHouse

--Predicting preference



-- We request 200 images with tags from Pixabay API;

-- Predict and filter pictures based on user preference by spark Average Perceptron model trained before.

-- Display recommendating images.

Machine Learning Online Classification

--Online&Average Perceptron

```
[Yilans-MacBook-Air:large_data_pj Jillian$ /usr/local/Cellar/spark-2.0.1-bin-hadoop2.7/bin/spark-submit tfidf]
_classify_spark.py
Test error rate of NaiveBayes Model:  0.159159159159
Test error rate of SVMWithSGD Model(iterations=100):  0.223223223223
Test error rate of LogisticRegressionWithLBFGS Model:  0.0
Test error rate of OnlinePerceptron(iterations=1):  0.206206206206
Test error rate of OnlinePerceptron(iterations=10):  0.206206206206
Test error rate of AveragePerceptron(iterations=1):  0.168168168168
Test error rate of AveragePerceptron(iterations=1) after single-pass OnlinePerceptron:  0.133133133133
Test error rate of AveragePerceptron(iterations=10) after 10 time pass OnlinePerceptron:  0.0930930930931
Test error rate of AveragePerceptron(iterations=5)::  0.115115115115
Test error rate of AveragePerceptron(iterations=10):  0.0930930930931
Test error rate of AveragePerceptron(iterations=100):  0.048048048048


Confusion Matrix of models:
+--------------------------+-------------------+-------------+-------------+------------+------------+
|Model                     |errorRate          |falseNegative|falsePositive|trueNegative|truePositive|
+--------------------------+-------------------+-------------+-------------+------------+------------+
|NaiveBayes                |0.15915915915915915|63           |96           |228         |612         |
|SVMWithSGDItr100          |0.22322322322322322|18           |205          |119         |657         |
|LogisticRegressionWithLBFGS|0.0                |0            |0            |324         |675         |
|OnlinePerceptronItr1      |0.2062062062062062 |160          |46           |278         |515         |
|OnlinePerceptronItr10     |0.2062062062062062 |30           |201          |123         |645         |
|AveragePerceptronItr1     |0.16816816816816818|58           |110          |214         |617         |
|AveragePerceptronOPItr1   |0.13313313313313313|47           |86           |238         |628         |
|AveragePerceptronOPItr10  |0.0930930930930931 |34           |59           |265         |641         |
|AveragePerceptronItr5     |0.11511511511511512|42           |73           |251         |633         |
|AveragePerceptronItr10    |0.0930930930930931 |32           |61           |263         |643         |
|AveragePerceptronItr100   |0.048048048048048045|22           |26           |298         |653         |
+--------------------------+-------------------+-------------+-------------+------------+------------+
```

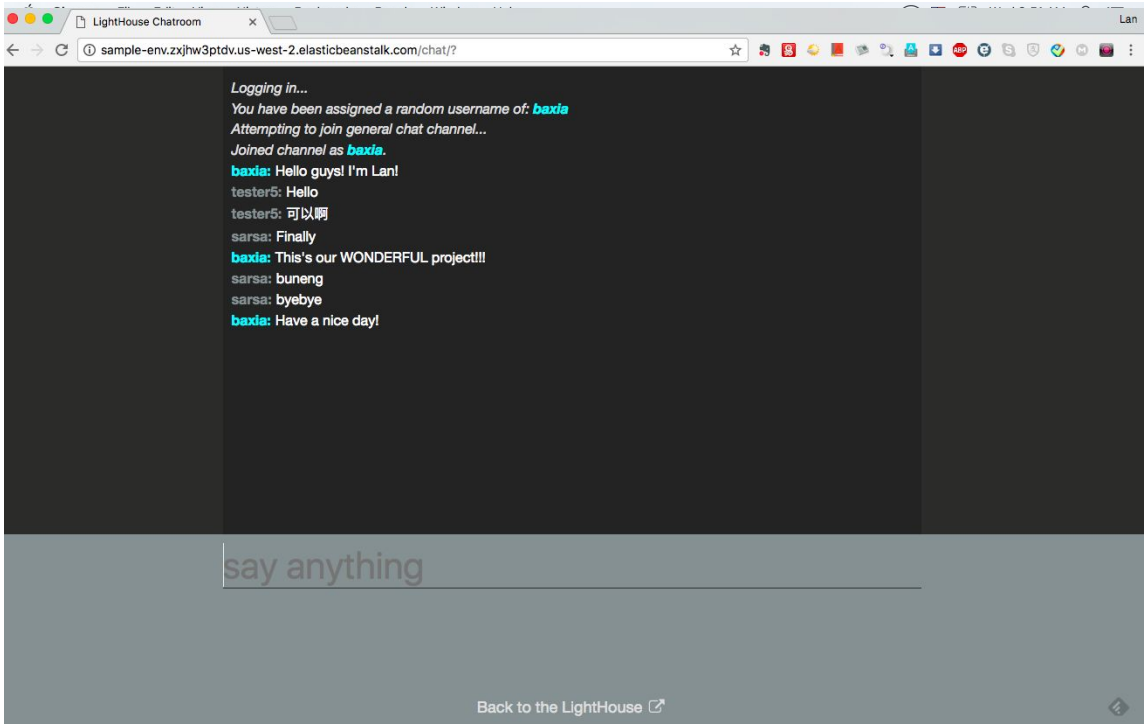
Using 5000 restuarant text reviews with labels: {+1(like),-1(dislike)} as training data. Using HashingTF function in MLlib to get tf of words.

Comparing Perceptron models with the classification models in pyspark MLlib. Show test error rate and confusion matrix.

Machine Learning On AWS Spark

- Get pictures from Flickr API and use machine learning API to predict its tags; display images on web and let user indicate preference.
- Deploy Spark on AWS EMR; Send user feedback {tags; preference} to SQS; consume and train by perceptron spark model
- If user requests recommendation, server will send request{username} to SQS; spark model then consumes message, requests latest images from Pixabay API, send positive predictions about specific user to SQS
- Server receive metadata and show recemondations for specific user on the web.

Let’s Chat



We implement a general chatroom for users to communicate with each other or share their ideas about the photos they see.

The chatroom is supported by Twilio’s API. All users of our app will be in a general channel.

ChatRoom Implementation

- Twilio SDK
- User login: fetch an AccessToken from the server via Ajax, pass a nser name
- Initialize the Chat JavaScript SDK
- Handle UI events to send messages or display new incoming messages