Catching The Star

Analyzing Yelp Coffee Shop Dataset

Veronica Hui

Feb 2017

Goal: Predicting the customer sentiment for coffee shops on Yelp

Motivation: Understand the driver of user sentiment of a coffee shop on Yelp

The Dataset

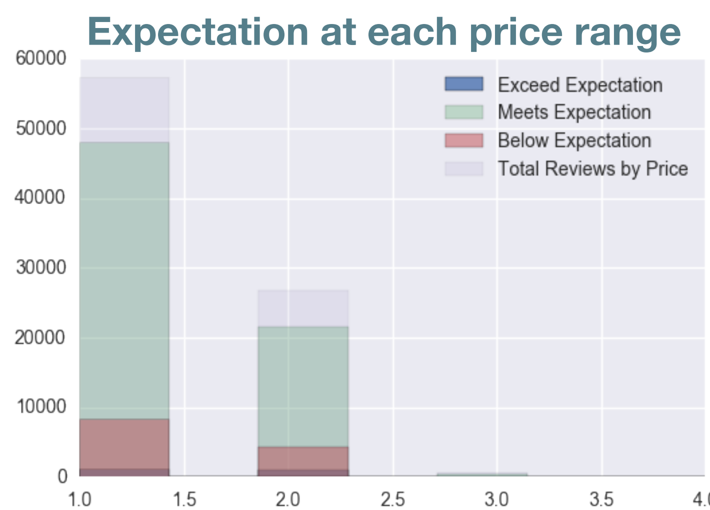
- 5 Files on business, user, checkins, tips, and review
- 100k+ Rows
- 100+ Columns

First Attempt - Price and Bias

Hypothesis 1: Based on the data set for coffee shops between 2004-10-12 and 2016-07-19, the higher the price, the higher expectation (star of the business - star by the user) compared to its average rating

Observation:

- At a lower price range, there's a wider variation of user expectation; the higher the range is, the narrower the variation, and the expectation is concentrated on the positive side, where individual rating is lower than the average business rating
- There are way more reviews for coffee shops in the $ - $$ price range, and most meets the expectation

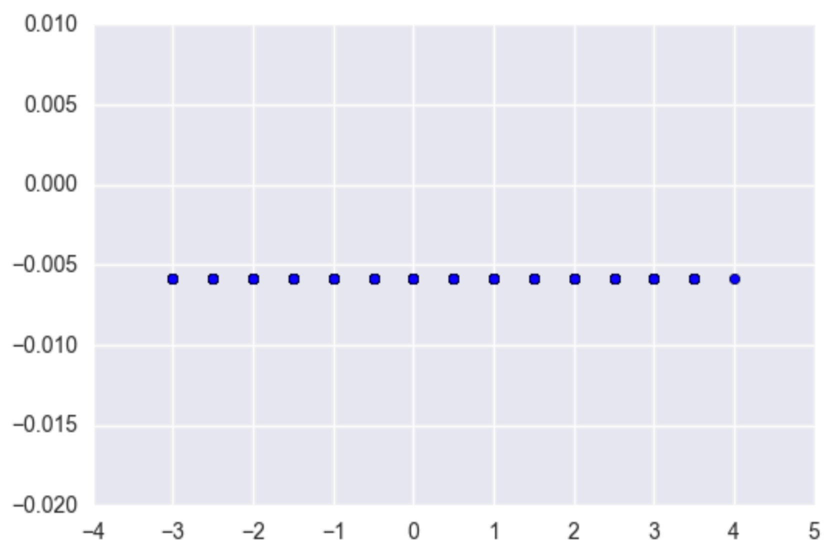


Price and Bias - Analysis

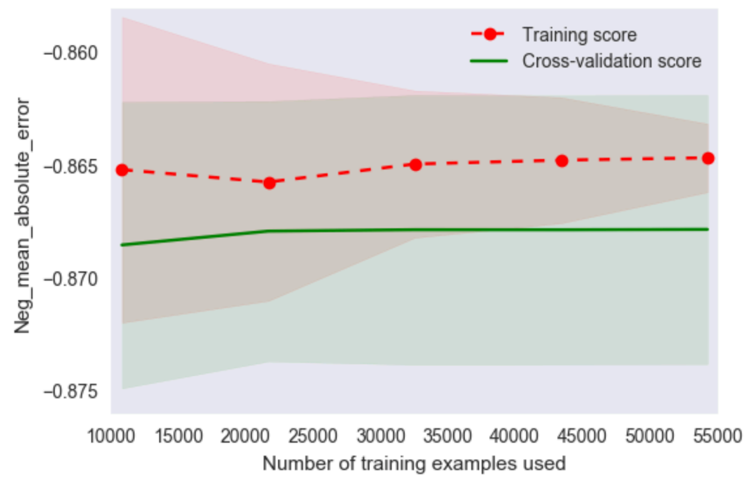
```
In [57]: from sklearn.metrics import mean_absolute_error
         mean_absolute_error(y_true = y_test, y_pred=y_pred)
Out[57]: 0.8743576860566995
```

```
In [54]: clf.best_estimator_.coef_
Out[54]: array([-0.])
```

```
In [118]: %matplotlib inline
          plt.scatter(x = y_test, y=y_pred)
Out[118]: <matplotlib.collections.PathCollection at 0x13145d250>
```




```
In [99]: draw_learning_curve(clf, X_train, y_train, cv=cv,
                             scoring='neg_mean_absolute_error');
```

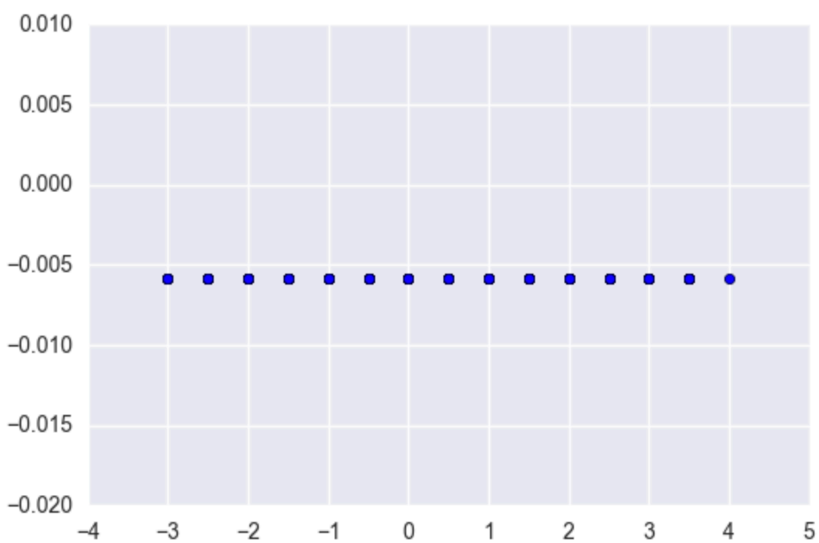


Price and Bias - Analysis

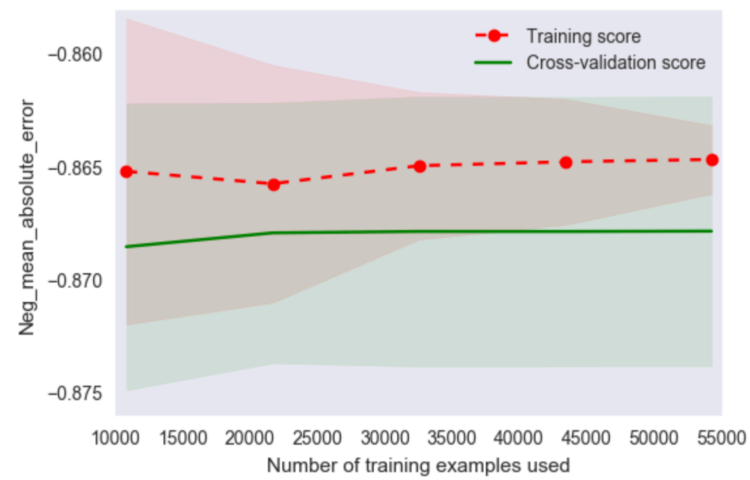
In [57]:
```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(y_true = y_test, y_pred=y_pred)
```
Out[57]: 0.8743576860566995

In [54]:
```
clf.best_estimator_.coef_
```
Out[54]: array([-0.])

In [118]:
```
%matplotlib inline
plt.scatter(x = y_test, y=y_pred)
```
Out[118]: <matplotlib.collections.PathCollection at 0x13145d250>



In [99]:
```
draw_learning_curve(clf, X_train, y_train, cv=cv,
                    scoring='neg_mean_absolute_error');
```



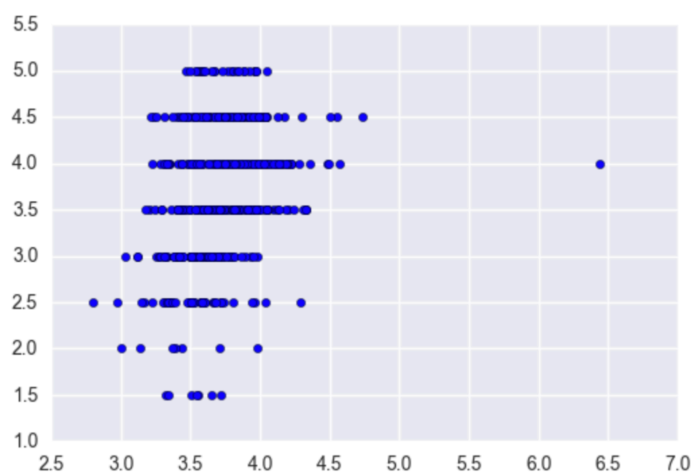
User Engagement and Amenities

Hypothesis 2: The more engaged the business keeps the customer, using review, checkin, etc, the higher rating it gets

Model: Linear regression

Evaluation: Accuracy

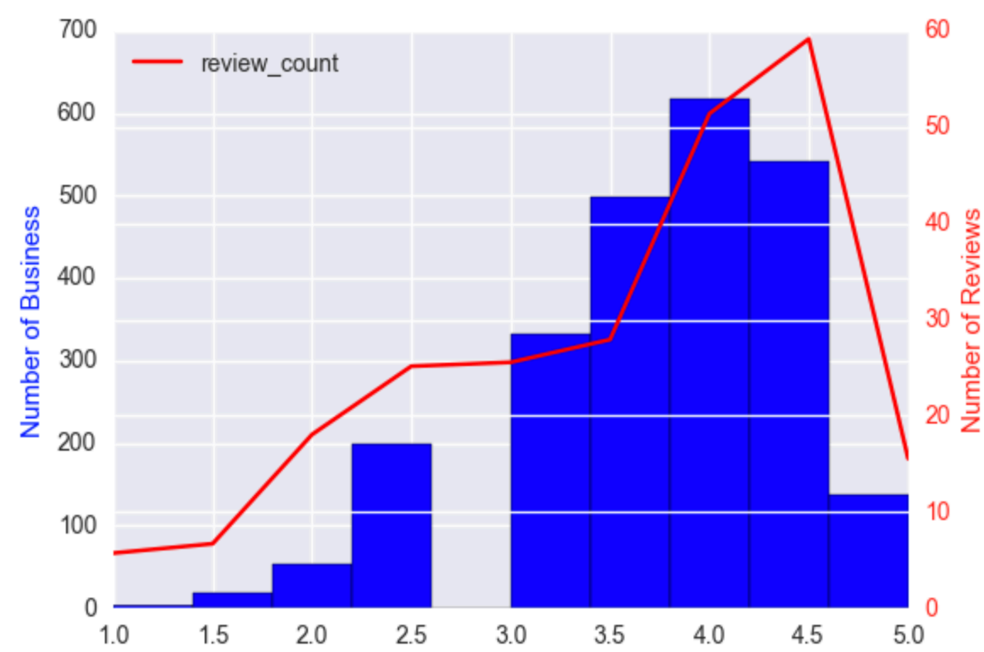
```
In [130]: %matplotlib inline
          plt.scatter(x = y_pred, y=y_test)
Out[130]: <matplotlib.collections.PathCollection at 0x1771512d0>
```



	Coefficient_LR	Feature
14	0.177949	Total_Votes
5	0.153703	review_count
13	0.115564	attributes.Free-Wi-Fi
2	0.095881	attributes.Parking.street
11	0.092031	longitude
1	0.062554	attributes.Parking.lot
10	0.046424	latitude
0	0.030276	attributes.Outdoor Seating
12	0.016218	attributes.Parking.validated
16	0.014489	Total_Likes
7	-0.005735	attributes.Parking.valet
6	-0.025628	attributes.Accepts Credit Cards
4	-0.028289	attributes.Parking.garage
9	-0.059142	open
17	-0.080591	Total_Checkin
3	-0.085925	attributes.Price Range
8	-0.164597	attributes.Wi-Fi
15	-0.183266	Total_Tips

User Engagement - Take Two

- Predicting sentiment instead of stars
- Models: Logistic regression & Random Forest
- Evaluation: accuracy, f1 and AUC



User Engagement - Take Two

Feature
Total_Votes
review_count
attributes.Free-Wi-Fi
attributes.Parking.street
longitude
latitude
attributes.Outdoor Seating
attributes.Parking.lot
Total_Likes
attributes.Parking.validated
attributes.Parking.valet
attributes.Accepts Credit Cards
attributes.Parking.garage
open
attributes.Price Range
attributes.Wi-Fi
Total_Checkin
Total_Tips

User Engagement - Take Two

Feature	Coefficient_LgReg
Total_Votes	1.132725
review_count	0.658793
attributes.Free-Wi-Fi	0.466701
attributes.Parking.street	0.246041
longitude	0.188361
latitude	0.145114
attributes.Outdoor Seating	0.087756
attributes.Parking.lot	0.082555
Total_Likes	0.077926
attributes.Parking.validated	0.029697
attributes.Parking.valet	-0.037266
attributes.Accepts Credit Cards	-0.054014
attributes.Parking.garage	-0.094747
open	-0.125446
attributes.Price Range	-0.183227
attributes.Wi-Fi	-0.548491
Total_Checkin	-0.587731
Total_Tips	-0.608664

User Engagement - Take Two

Feature	Coefficient_LgReg
Total_Votes	1.132725
review_count	0.658793
attributes.Free-Wi-Fi	0.466701
attributes.Parking.street	0.246041
longitude	0.188361
latitude	0.145114
attributes.Outdoor Seating	0.087756
attributes.Parking.lot	0.082555
Total_Likes	0.077926
attributes.Parking.validated	0.029697
attributes.Parking.valet	-0.037266
attributes.Accepts Credit Cards	-0.054014
attributes.Parking.garage	-0.094747
open	-0.125446
attributes.Price Range	-0.183227
attributes.Wi-Fi	-0.548491
Total_Checkin	-0.587731
Total_Tips	-0.608664

Feature	Importance Score_RF
longitude	0.166892
latitude	0.145369
Total_Checkin	0.144488
review_count	0.141140
Total_Votes	0.140113
Total_Tips	0.093265
attributes.Price Range	0.035448
open	0.022624
attributes.Outdoor Seating	0.019595
attributes.Parking.street	0.019550
Total_Likes	0.012985
attributes.Parking.lot	0.012613
attributes.Accepts Credit Cards	0.012320
attributes.Wi-Fi	0.011858
attributes.Free-Wi-Fi	0.011379
attributes.Parking.garage	0.008666
attributes.Parking.valet	0.001253
attributes.Parking.validated	0.000442

User Engagement - Take Two

In [171]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},index=np.arange(1))
f1
```

Out[171]:

	F1_LogReg	F1_RF
0	0.632231	0.674464

In [172]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_lgreg.mean()},index=np.arange(1))
auc
```

Out[172]:

	AUC_LogReg	AUC_RF
0	0.664576	0.687554

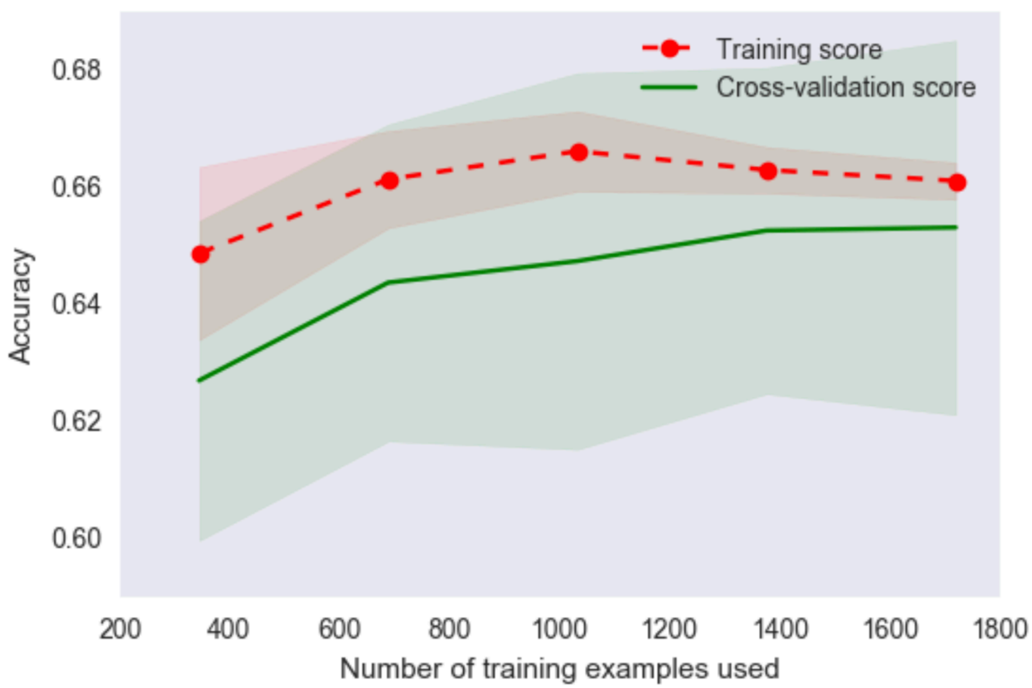
In [173]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy_LogReg':accuracy_lgreg},index=np.arange(1))
accuracy
```

Out[173]:

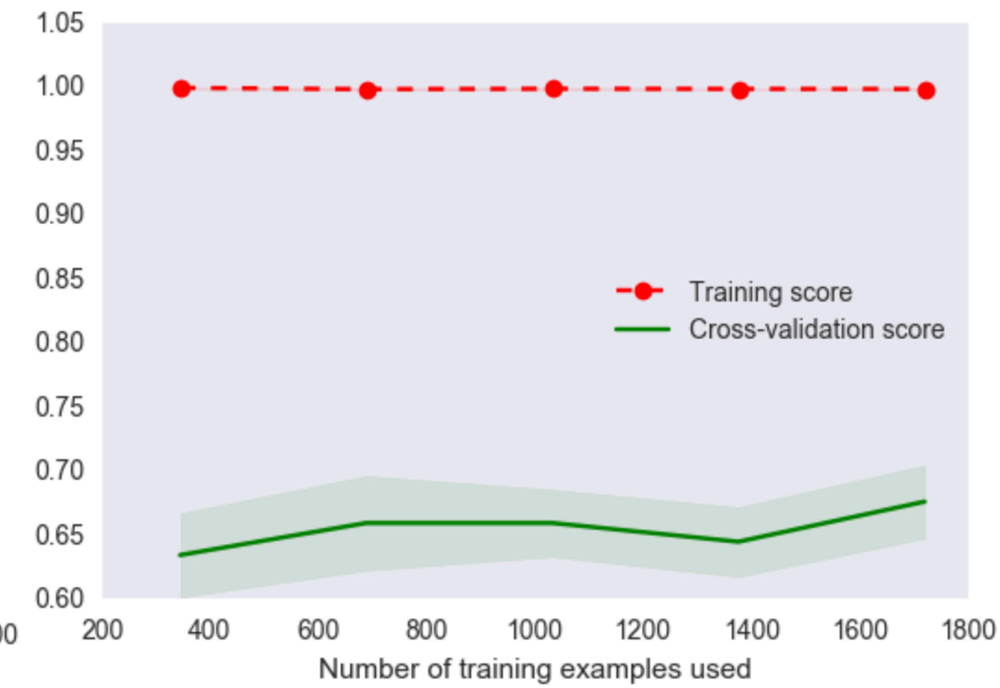
	Accuracy_LogReg	Accuracy_RF
0	0.627615	0.650628

User Engagement - Take Two

Logistic Regression

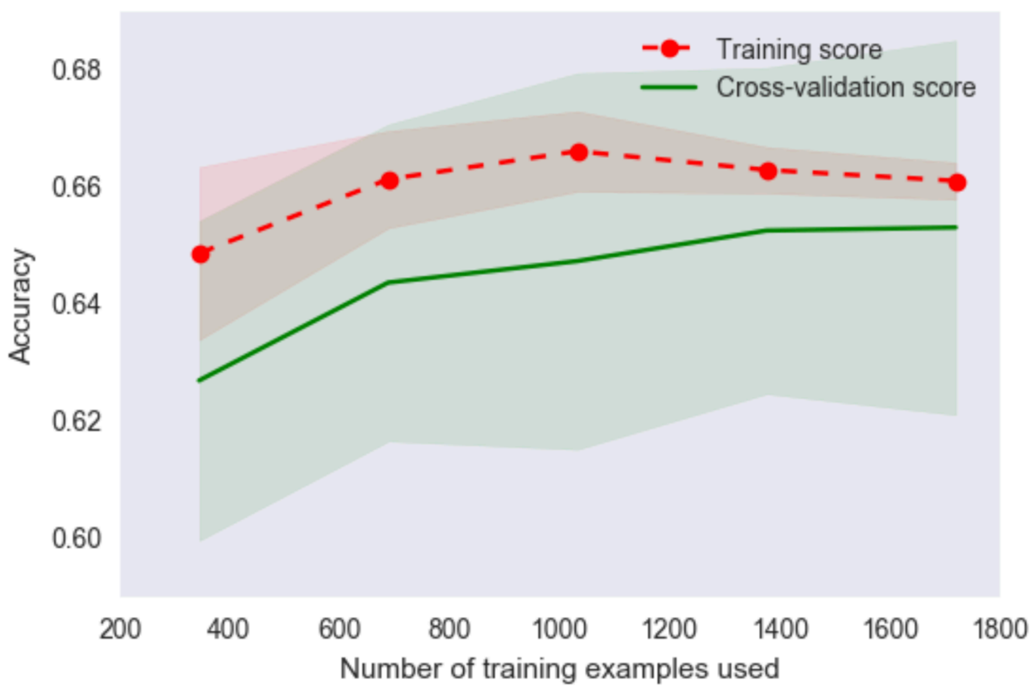


Random Forest

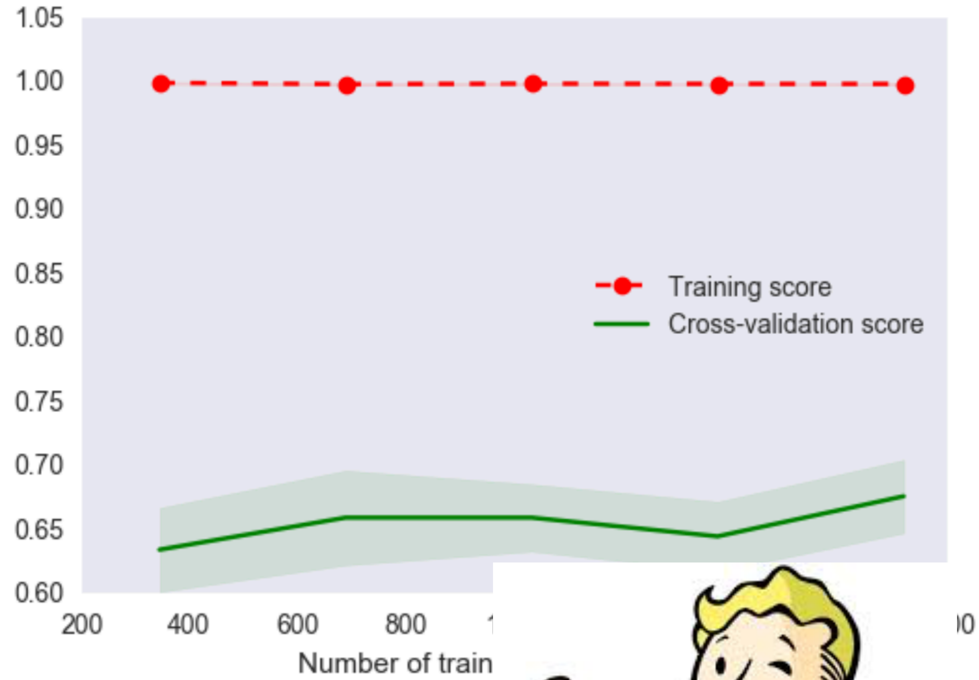


User Engagement - Take Two

Logistic Regression



Random Forest



User Engagement - Feature Selection

	Coefficient_LR	Feature	Coefficient_LgReg	Importance Score_RF
11	0.092031	longitude	0.188361	0.166892
10	0.046424	latitude	0.145114	0.145369
17	-0.080591	Total_Checkin	-0.587731	0.144488
5	0.153703	review_count	0.658793	0.141140
14	0.177949	Total_Votes	1.132725	0.140113
15	-0.183266	Total_Tips	-0.608664	0.093265
3	-0.085925	attributes.Price Range	-0.183227	0.035448
9	-0.059142	open	-0.125446	0.022624
0	0.030276	attributes.Outdoor Seating	0.087756	0.019595
2	0.095881	attributes.Parking.street	0.246041	0.019550
16	0.014489	Total_Likes	0.077926	0.012985
1	0.062554	attributes.Parking.lot	0.082555	0.012613
6	-0.025628	attributes.Accepts Credit Cards	-0.054014	0.012320
8	-0.164597	attributes.Wi-Fi	-0.548491	0.011858
13	0.115564	attributes.Free-Wi-Fi	0.466701	0.011379
4	-0.028289	attributes.Parking.garage	-0.094747	0.008666
7	-0.005735	attributes.Parking.valet	-0.037266	0.001253
12	0.016218	attributes.Parking.validated	0.029697	0.000442

User Engagement - Feature Selection

	Feature	Importance Score_RF	Coefficient_LgReg
1	Total_Votes	0.263134	1.108487
2	review_count	0.222015	0.873157
5	attributes.Free-Wi-Fi	0.029270	-0.083926
4	attributes.Price Range	0.047260	-0.163040
0	Total_Checkin	0.293045	-0.628683
3	Total_Tips	0.145275	-0.778291

In [223]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},index=np.arange(1))
f1
```
Out[223]:

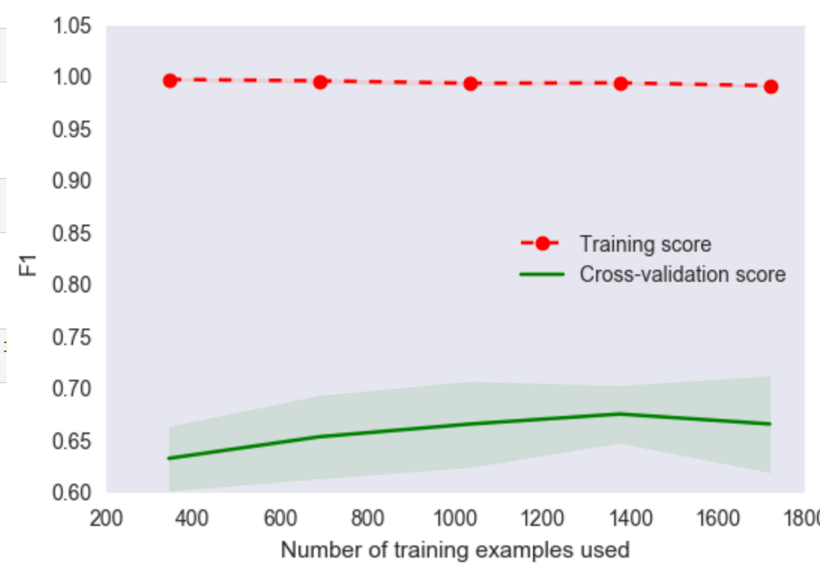
	F1_LogReg	F1_RF
0	0.640145	0.579365

In [224]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_lgreg.mean()},index=np.arange(1))
auc
```
Out[224]:

	AUC_LogReg	AUC_RF
0	0.640489	0.635434

In [225]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy_LogReg':accuracy_lgreg},index=np.arange(
accuracy
```
Out[225]:

	Accuracy_LogReg	Accuracy_RF
0	0.583682	0.556485



User Engagement - Feature Selection

In [171]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},index=np.arange(1))
f1
```

Out[171]:

	F1_LogReg	F1_RF
0	0.632231	0.674464

In [172]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_lgreg.mean()},index=np.arange(1))
auc
```

Out[172]:

	AUC_LogReg	AUC_RF
0	0.664576	0.687554

In [173]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy_LogReg':accuracy_lgreg},index=np.arange(1))
accuracy
```

Out[173]:

	Accuracy_LogReg	Accuracy_RF
0	0.627615	0.650628

In [223]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},index=np.arange(1))
f1
```

Out[223]:

	F1_LogReg	F1_RF
0	0.640145	0.579365

In [224]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_lgreg.mean()},index=np.arange(1))
auc
```

Out[224]:

	AUC_LogReg	AUC_RF
0	0.640489	0.635434

In [225]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy_LogReg':accuracy_lgreg},index=np.arange(1))
accuracy
```

Out[225]:

	Accuracy_LogReg	Accuracy_RF
0	0.583682	0.556485

User Engagement - Feature Selection

In [171]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},index=np.arange(1))
f1
```

Out[171]:

	F1_LogReg	F1_RF
0	0.632231	0.674464

In [172]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_lgreg.mean()},index=np.arange(1))
auc
```

Out[172]:

	AUC_LogReg	AUC_RF
0	0.664576	0.687554

In [173]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy_LogReg':accuracy_lgreg},index=np.arange(1))
accuracy
```

Out[173]:

	Accuracy_LogReg	Accuracy_RF
0	0.627615	0.650628

In [223]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},ind
f1
```

Out[223]:

	F1_LogReg	F1_RF
0	0.640145	0.579365

In [224]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_
auc
```

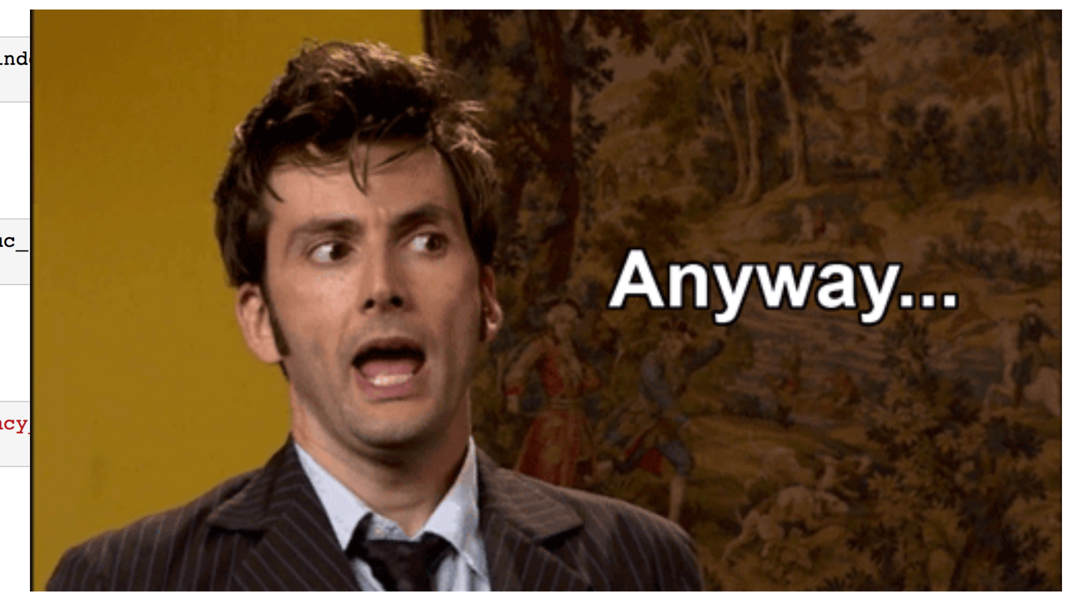
Out[224]:

	AUC_LogReg	AUC_RF
0	0.640489	0.635434

In [225]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy
accuracy
```

Out[225]:

	Accuracy_LogReg	Accuracy_RF
0	0.583682	0.556485



User Engagement - Feature Engineering

Interaction: Price and Star

In [227]:
```
## star/price - higher, the better, i.e. for each dollar spent how much do you like this place?
data['value'] = data['stars'].divide(data['attributes.Price Range'])
```

Ratios: Vote, Tip and Checkin taking into account the total number of review

In [228]:
```
## vote/review: the higher the better
data['vote_review'] = data.Total_Votes.divide(data.review_count)
## tip/review
data['tip_review'] = data.Total_Tips.divide(data.review_count)
## checkin/review
data['checkin_review'] = data.Total_Checkin.divide(data.review_count)
```

	Feature	Importance Score_RF	Coefficient_LgReg
0	value	0.634440	0.004822
2	review_count	0.085757	0.001492
3	vote_review	0.077127	0.000521
5	attributes.Free-Wi-Fi	0.013213	-0.000393
4	tip_review	0.065663	-0.001033
1	checkin_review	0.123800	-0.001738

User Engagement - Feature Engineering

In [254]:
```
f1 = pd.DataFrame({'F1_RF': f1_rf, 'F1_LogReg': f1_lgreg},index=np.arange(1))
f1
```

Out[254]:

	F1_LogReg	F1_RF
0	0.662474	0.936508

In [255]:
```
auc = pd.DataFrame({'AUC_RF':auc_rf.mean(), 'AUC_LogReg':auc_lgreg.mean()},index=np.arange(1))
auc
```

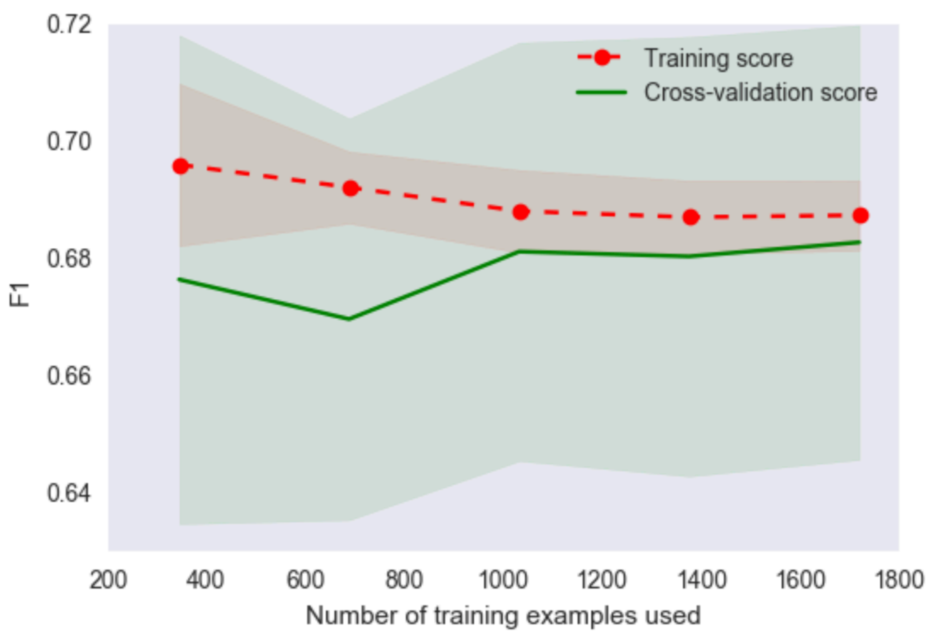
Out[255]:

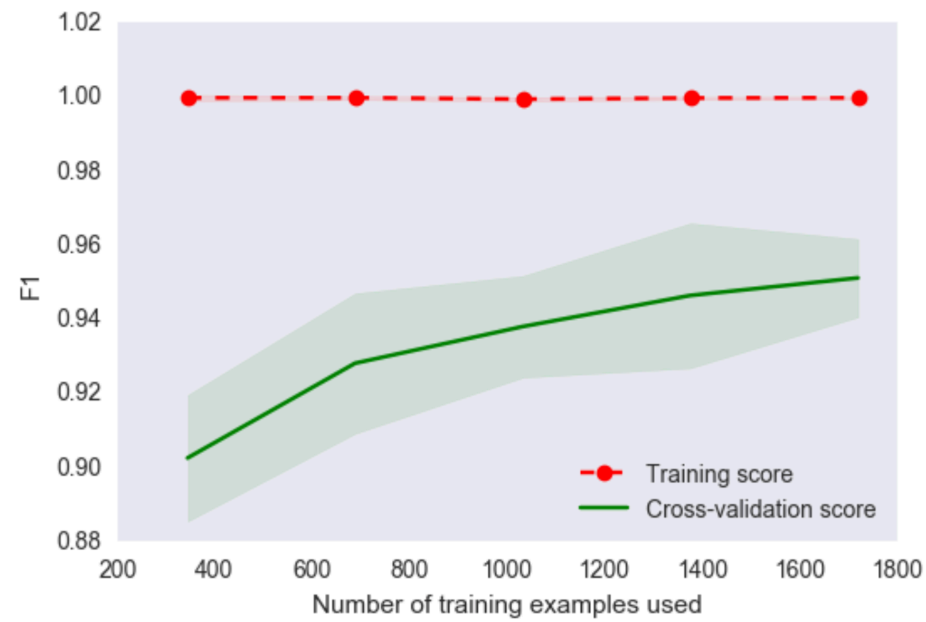
	AUC_LogReg	AUC_RF
0	0.794147	0.969855

In [256]:
```
accuracy = pd.DataFrame({'Accuracy_RF':accuracy_rf, 'Accuracy_LogReg':accuracy_lgreg},index=np.arange(1))
accuracy
```

Out[256]:

	Accuracy_LogReg	Accuracy_RF
0	0.66318	0.933054





Conclusion

- Value plays a critical role in consumer sentiment for coffee shops
- The more review, the more favorable the coffee shops are
- If the reviews are in high quality, and voted by other users, it’s more likely to be a favorable coffee shop

Action

- Coffee Shop: Encourage customers to “rate us on Yelp” and provide such reward
- Yelp: Establish a more robust social network to engage users and provide more meaningful activities

Further Studies

Clustering and KNN: Customer clustering, predicting customer's rating for a specific business

NLP: employ NLP techniques to further digest consumer sentiment