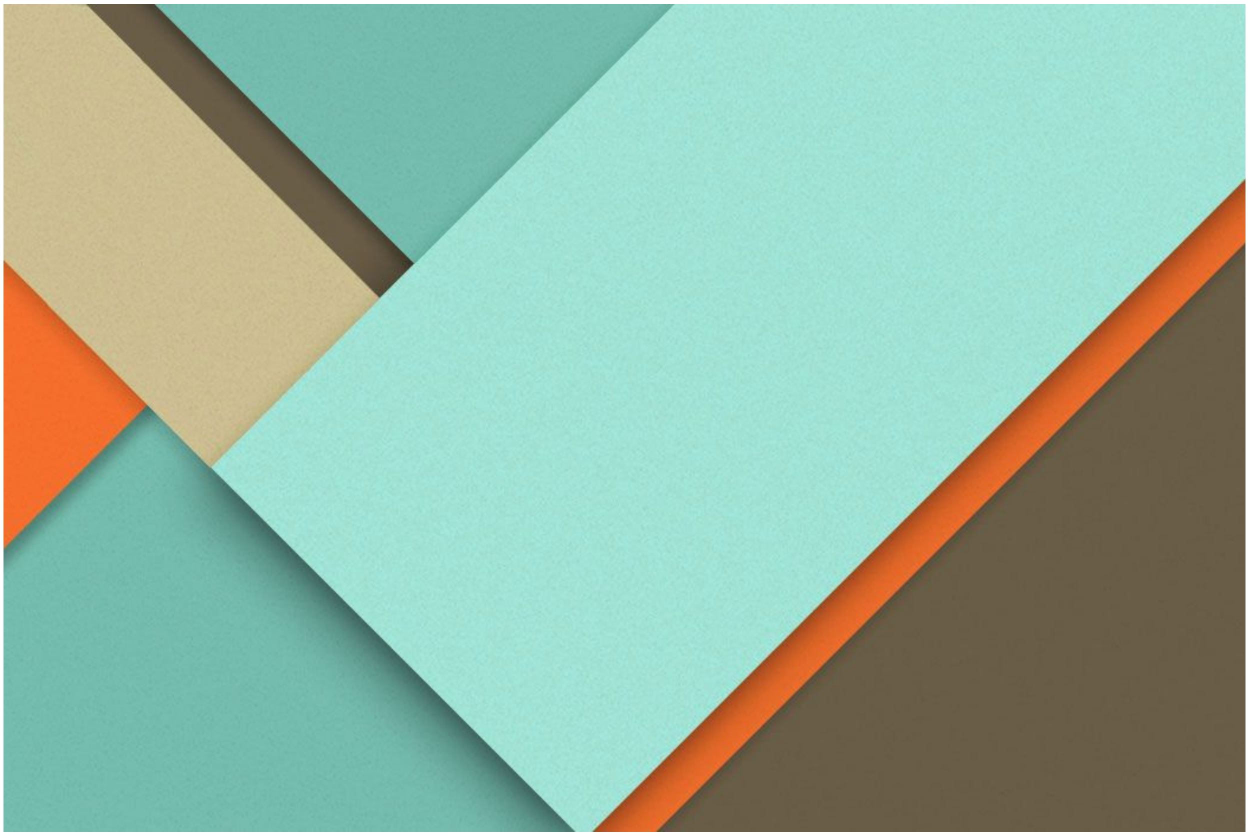


Search (ILZ-534) Final Project

09.12.2019

Pranav Gujarathi(pgujarat@iu.edu)

Yashvardhan Jain(yashjain@iu.edu)

Naga Anjaneyulu(nakopa@iu.edu)

Overview

We have built a recommendation system for the users, where we recommend businesses to a user and we have also built a service for businesses to help them identify the driving factors for their businesses.

Project Files

Task	File name	Contribution
Data Cleaning	cleaning_attributes_of_businesses.ipynb	Yashvardhan Jain
Data analysis	XGBoostModel_feature_importance.ipynb	Yashvardhan Jain
Task -1, Data Preprocessing	preprocessing_data.ipynb	Yashvardhan Jain, Pranav Gujarathi, Naga Anjaneyulu
Task -1 , Model 1	embedded_bias_model.py	Naga Anjaneyulu
Task-1 , Model 2.1,2.2	memory_embeddings.py, embedded_nn_model_1_results.py, embedded_nn_model_2.ipynb ,	Naga Anjaneyulu, Pranav Gujarathi
Task-2		Pranav Gujarathi

Exploring the Data

File Name	Number of records
User.json	1,637,138 unique users
Business.json	192,609 unique businesses
Tip.json	1,223,094 tips

Review.json	6,685,900 reviews
Check In.json	161,950 check-ins

Table 1: Yelp Dataset

The Yelp Dataset contains the above-mentioned data that we can use for our final project. We are also given a "photos.json" file that contains a large image dataset. But we discard that and only focus on datasets with textual features.

As we can observe, this given dataset is too large to work with. Hence, we decide to pick a well-defined subset of this dataset and use that subset for the project

Analyzing and Filtering Data

- Our first step is to subsample all the restaurants that are related to Pizza. We do this by checking the categories of all the given businesses for the tags "pizza" and "Italian".
- This gives us approximately 6000 businesses and corresponding to these businesses we get approximately 600,000 reviews.
- Furthermore, we only select the users that have given more than 5 reviews. This reduces the total number of reviews to approximately 160,000 reviews.
- Finally, if a user has given multiple reviews for the same business, we only consider the latest review.
- This sampling scheme finally gives us approximately 159,000 reviews to work with.

Data Preprocessing and Cleanup

- Once we have our smaller subset of the Yelp Dataset, we analyze the dataset to figure out the important features.
- The features that are related to the reviews, such as review text, review sentiment (cool, funny, useful), are not directly used to predict the star rating. Instead, we use the concept of legacy features.
- We define the legacy feature like this: For a given user, if all the reviews are arranged in chronological order, and we pick a review, then the aggregation of all these features before that review would become a legacy feature that can be used to predict the star rating of this review. We create such legacy features for all reviews.
- For example, if a user has written 5 reviews, then we arrange these reviews in chronological order. We say that for the chronologically 3rd review, the aggregate of the features of the first two reviews can be used to predict the star rating of the 3rd review.

- Finally, we have time-independent features such as the business attributes and the user attributes which we can use after appropriate data preprocessing on them. These features help us make predictions in case of a cold start.
- For the business attributes, we remove all the attributes that have over 80% values missing. We also remove attributes that have very skewed values and we remove non-variant features. Then, since all the attributes are categorical attributes, we one-hot encode them.
- This creates a proper clean dataset that we use for predictions.

Processing Text Data

- We use the review text submitted by the users as additional features for our models.
- We use the same concept of "legacy features" for the review text as well. We use the aggregate of the chronologically previous reviews to predict the star rating of the current review.
- We clean the review text using the common text preprocessing techniques such as removal of non-alphanumeric terms, removal of stopwords, and converting all terms to lowercase. Then we tokenize the terms of the review texts.
- We feed these tokenized terms to a TF-IDF vectorizer to get scores associated with each term, however, we only consider terms that have the document frequency above a certain threshold.
- Finally, we have terms and their TF-IDF scores as features that we can use for each star rating prediction.

Final Dataset View

Our final dataset has the following features:

Caters	lot	Restaurants Reservations	Dessert
divey	Noise Level	Restaurants Price Range	Lunch
trendy	Business Accepts Credit Cards	Good For Kids	Dinner
casual	Restaurants Delivery	Has TV	Bike Parking
garage	Alcohol	Outdoor Seating	Restaurants Table Service
street	WiFi		

Table 2: Business attributes used

business_legacy_stars_mean	business_legacy_funny	business_legacy_cool	business_legacy_useful
user_legacy_funny	user_legacy_cool	user_legacy_useful	user_legacy_stars_mean
legacy_review_text			

Table 3: Legacy features used

compliment_hot	compliment_more	compliment_profile	compliment_profile
compliment_cute	compliment_list	compliment_note	compliment_plain
compliment_cool	compliment_funny	compliment_writer	compliment_photos

Table 4: Other user related features used

business_id	user_id	review_id	yelping_since_td
stars_review			

Table 5: Other features

- The business attributes (as shown in Table 2) are categorical and hence are one-hot encoded.
- Table 3 shows legacy features. “Business_legacy” features are related to each review given for business. Whereas “user_legacy” features are related to the user.
- Table 4 shows other user-related features that we have used.
- “Yelping_since_td” feature in table 5 consists of all the “yelping since” dates for each user that we have converted to a time delta.
- “Legacy_review_text” in table 3 is the tokenized and TF-IDF vectorized review text for all the reviews.

Task 1 : Recommending businesses to users

Problem Statement

We would like to predict how a user would rate a business which he/she hasn’t visited before based on the reviews and user profile.Based on these star ratings, we would recommend restaurants to this user in decreasing order of the predicted star ratings.

Assumption: Star rating is the complete indicator of a user's preference for a business.

Approach

We have built two models for building this recommendation system

Model - 1: EmbeddedDot Model

This is a memory-based collaborative filtering model. We assume each business and user is associated with a dense latent vector, and *the dot product/cosine similarity of the latent vectors of the user and the business should be roughly equal to the metric we are trying to predict.* The vectors are then updated based on the backpropagation resulting from comparing the prediction and ground truth.

Predicted Rating = Dot product of embedded vectors + user bias + business bias

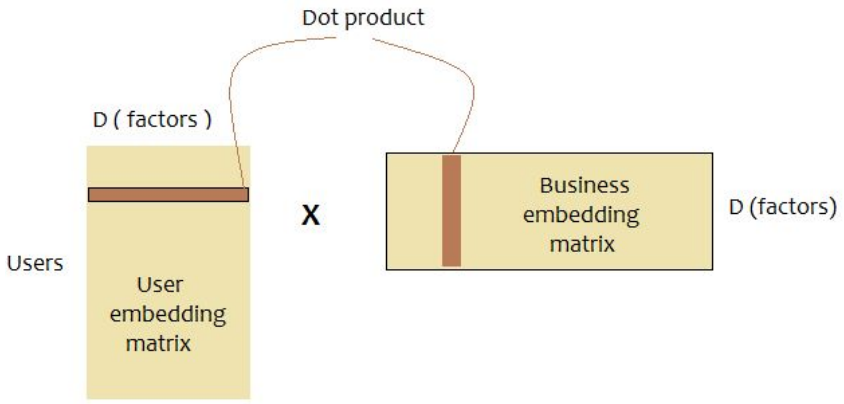


Figure 1 : Embedding matrices

We use the user bias and business bias in order to capture information about a specific user's or a business's nature which is independent of its interaction with other users/business .For example , some users generally would usually give only certain ratings , or some businesses would generally would be rated higher etc. Biases would capture such information.

Evaluation and Results :

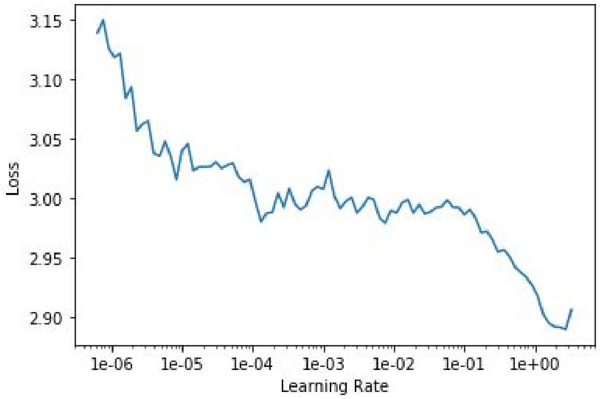
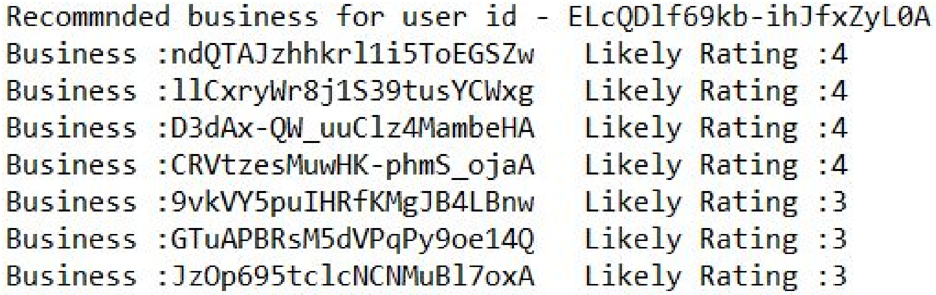


Figure 2 : Learning rate for the model

F1 score	Mean squared error
0.2809881	1.45093

Table 6 : Evaluation metrics



```
Recommnded business for user id - ELcQDlf69kb-ihJfxZyL0A
Business :ndQTAJzhhkrl1i5ToEGSZw   Likely Rating :4
Business :1lCxryWr8j1S39tusYCWxg   Likely Rating :4
Business :D3dAx-QW_uuClz4MambeHA   Likely Rating :4
Business :CRVtzesMuwHK-phmS_ojaA   Likely Rating :4
Business :9vkVY5puIHRfKMgJB4LBnw   Likely Rating :3
Business :GTuAPBRsM5dVPqPy9oe14Q   Likely Rating :3
Business :JzOp695tclcNCNMuBl7oxA   Likely Rating :3
```

Figure 3 : Recommended businesses for a user based on predicted rating

Model - 2.1: EmbeddedNN model

Instead of merely taking the dot product of the vectors as we have earlier done in our Model-1, *we concatenate the user and business embeddings and place additional Linear layers between it and the output, creating a Linear fully connected neural network. We use this neural network to predict the ratings.* Not only does this approach provide us with increased performance, but also we can now use multilabel cross-entropy loss for backpropagation, as opposed to being able to use only Mean Squared error loss in Model-1.

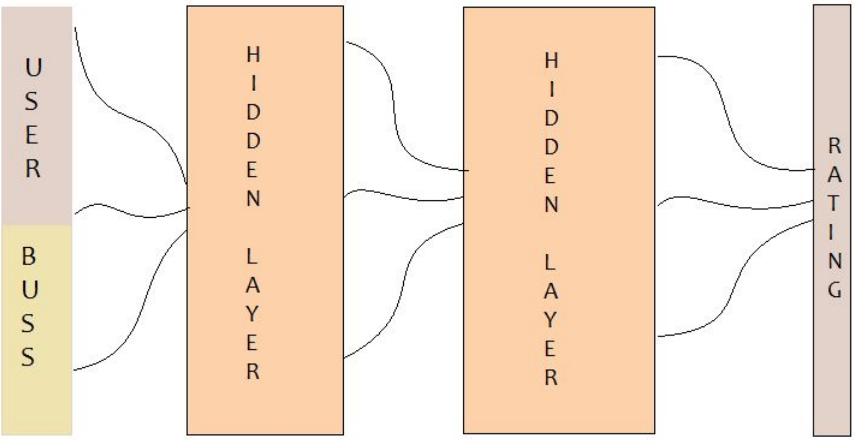


Figure 4: Network Architecture

Evaluation and Results :

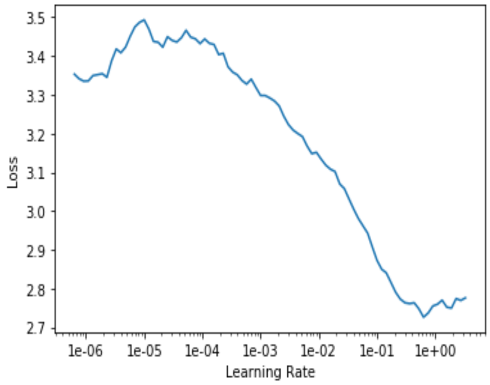
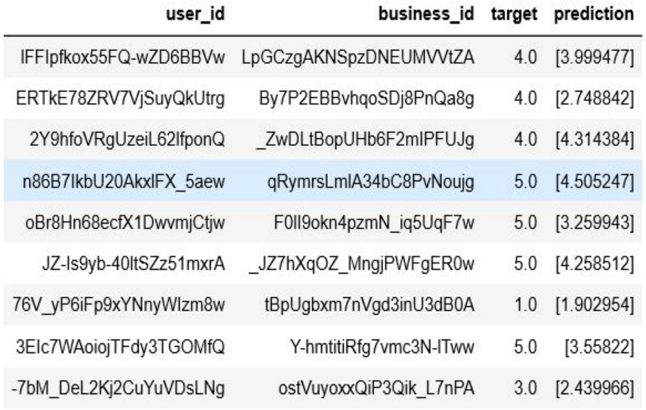
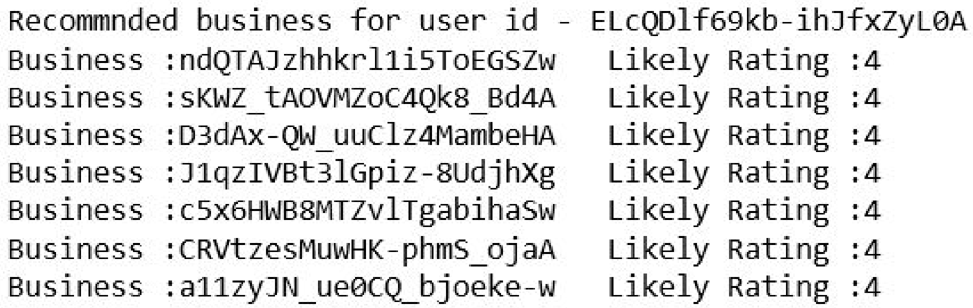


Figure 5 : Learning rate for the model



	user_id	business_id	target	prediction
	IFFIpfkox55FQ-wZD6BBVw	LpGCzgAKNSpzDNEUMVVtZA	4.0	[3.999477]
	ERTkE78ZRV7VjSuyQkUtrg	By7P2EBBvhqoSDj8PnQa8g	4.0	[2.748842]
	2Y9hfoVRgUzeiL62IfponQ	_ZwDLtBopUHb6F2mIPFUJg	4.0	[4.314384]
	n86B7IkbU20AkxlFX_5aew	qRymrsLmIA34bC8PvNoujg	5.0	[4.505247]
	oBr8Hn68ecfX1DwvmjCtjw	F0lI9okn4pzmN_iq5UqF7w	5.0	[3.259943]
	JZ-Is9yb-40ltSZz51mxrA	_JZ7hXqOZ_MngjPWFgER0w	5.0	[4.258512]
	76V_yP6iFp9xYNnyWlzm8w	tBpUgbxm7nVgd3inU3dB0A	1.0	[1.902954]
	3EIc7WAoiojTFdy3TGOMfQ	Y-hmtitiRfg7vmc3N-ITww	5.0	[3.55822]
	-7bM_DeL2Kj2CuYuVDsLNg	ostVuyoxxQiP3Qik_L7nPA	3.0	[2.439966]

Figure 6 : Predicted vs Actual Ratings



```
Recommnded business for user id - ELcQDlf69kb-ihJfxZyL0A
Business :ndQTAJzhhkrl1i5ToEGSZw   Likely Rating :4
Business :sKWZ_tAOVMZoC4Qk8_Bd4A   Likely Rating :4
Business :D3dAx-QW_uuClz4MambeHA   Likely Rating :4
Business :J1qzIVBt3lGpiz-8UdjhXg   Likely Rating :4
Business :c5x6HWB8MTZvlTgabihaSw   Likely Rating :4
Business :CRVtzesMuwHK-phmS_ojaA   Likely Rating :4
Business :a11zyJN_ue0CQ_bjoeke-w   Likely Rating :4
```

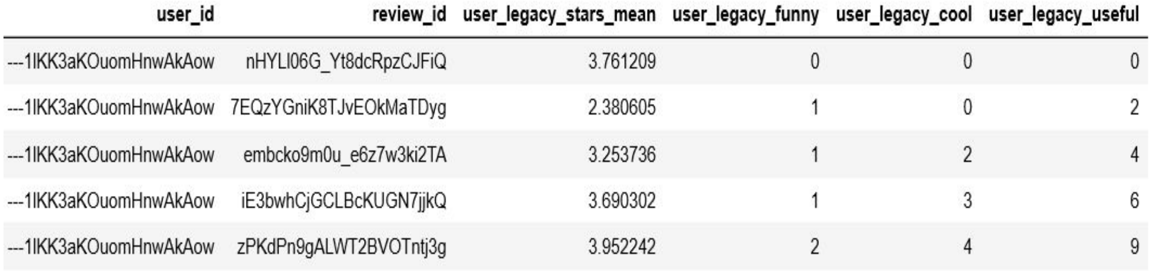
Figure 8 : Recommended businesses for a user based on predicted rating

F1 score	Mean squared error
0.409894	0.980186

Table 7 : Evaluation metrics

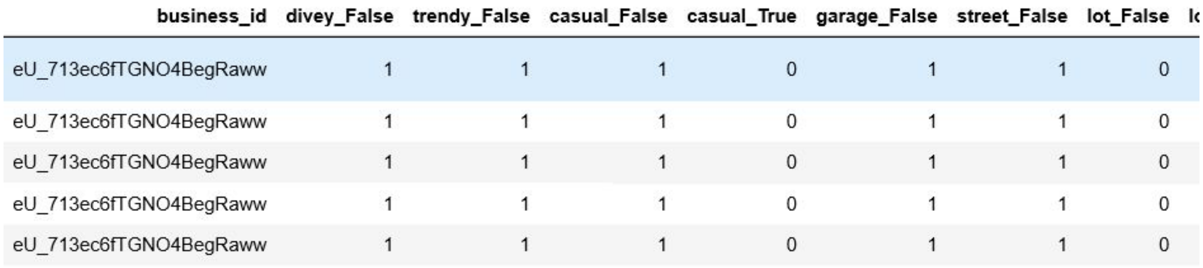
Model - 2.2 : Enhanced with additional features

One of the important advantages of using the framework in Model 2 is that, for a user and business combination, *we can additionally use all the different user and business specific features* we extracted earlier such as useful/cool/funny votes received by a user, ambiance, amenities provided by a business etc. *and concatenate it with the latent vectors to enhance our model's performance.*



user_id	review_id	user_legacy_stars_mean	user_legacy_funny	user_legacy_cool	user_legacy_useful
---1lKK3aKOuomHnwAkAow	nHYLI06G_Yt8dcRpzCJFiQ	3.761209	0	0	0
---1lKK3aKOuomHnwAkAow	7EQzYGniK8TJvEOkMaTDyg	2.380605	1	0	2
---1lKK3aKOuomHnwAkAow	embcko9m0u_e6z7w3ki2TA	3.253736	1	2	4
---1lKK3aKOuomHnwAkAow	iE3bwhCjGCLBcKUGN7jjkQ	3.690302	1	3	6
---1lKK3aKOuomHnwAkAow	zPKdPn9gALWT2BVOTntj3g	3.952242	2	4	9

Figure 7 : User specific legacy features



business_id	divey_False	trendy_False	casual_False	casual_True	garage_False	street_False	lot_False	lo
eU_713ec6fTGNO4BegRaww	1	1	1	0	1	1	0	
eU_713ec6fTGNO4BegRaww	1	1	1	0	1	1	0	
eU_713ec6fTGNO4BegRaww	1	1	1	0	1	1	0	
eU_713ec6fTGNO4BegRaww	1	1	1	0	1	1	0	
eU_713ec6fTGNO4BegRaww	1	1	1	0	1	1	0	

Figure 8 : Business specific legacy features

This enhanced model helps us solve the cold start problem. This model enhanced the performance and gave us the best output so far.

Evaluation and Results :

F1 score	Mean squared error
0.423	1.223

Table 8 : Evaluation Metrics

Task 2: Interpretation of Recommendation - how businesses can drive their strategy

Why do we need interpretation?

Drive business strategy and investment

Say a classification model predicts the star rating for the business of a yelp review(relevant/non-relevant, cool/uncool, funny/not funny). How does a user know using what previous ratings, business characteristics, or review text is affecting the recommendation? Our model has the capacity to predict star rating from any user/business combination. But a user will only most likely visit restaurants in his geographical proximity. Furthermore, how can we make predictions useful for a business owner who wants to improve his revenue? If we have, at a data point level, what affects the prediction positively and negatively, and by how much, a business owner can aggregate these impacts over only his potential customers and can guide his future investment and financial strategy based on that. This can be done not only because the impacts are available over a large dataset in a quantifiable fashion, but also because our model utilizes business attributes which simple memory based collaborative filtering techniques do not.

Avoiding Unconscious bias

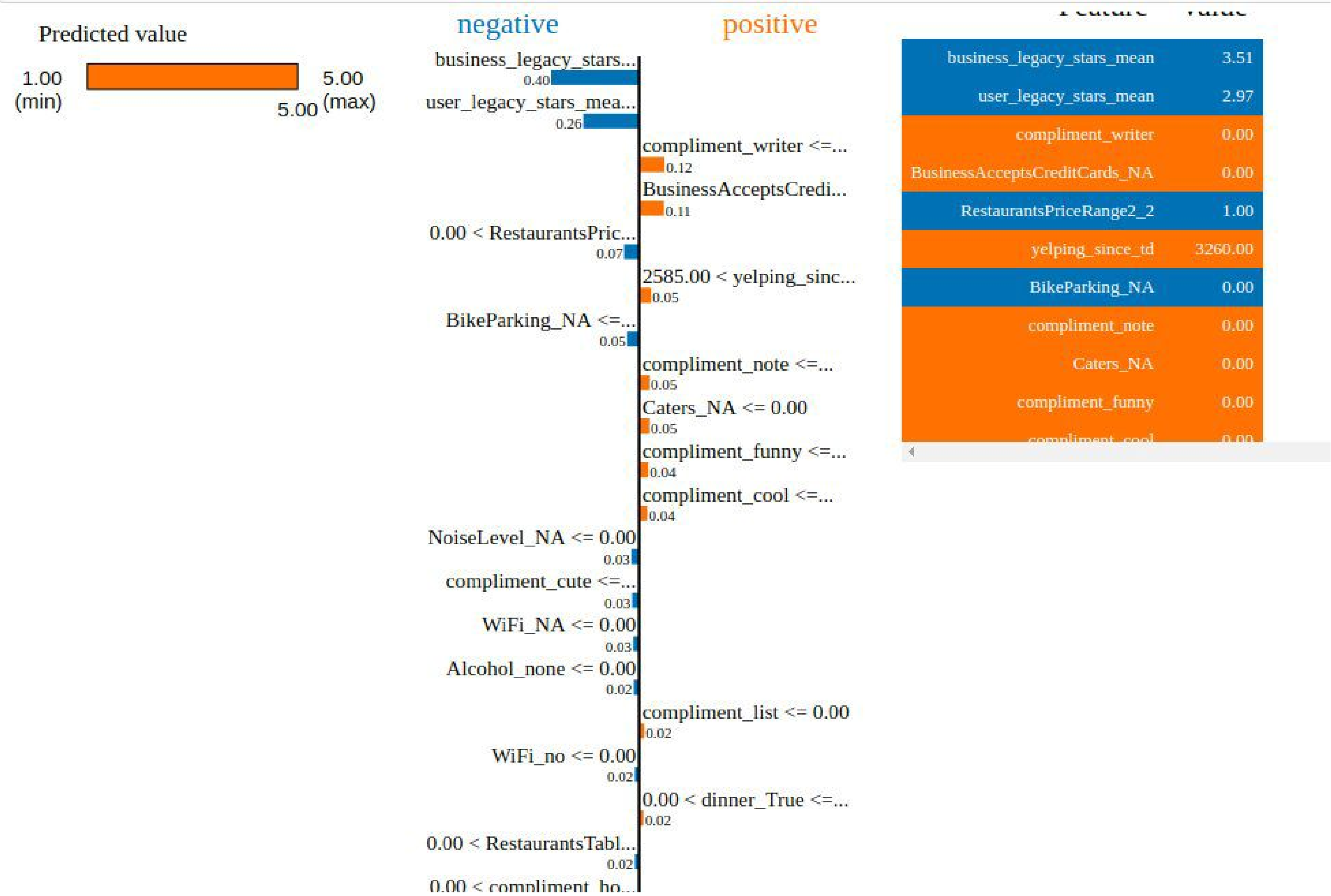
Unconscious biases are social stereotypes about certain groups of people that individuals form outside their own conscious awareness. Everyone holds unconscious beliefs about various social and identity groups, and these biases stem from one's tendency to organize social worlds by categorizing. In the case of our recommendation, the ML model by itself is a black box and may end up amplifying unconscious bias in its recommendations to improve performance. The only way of knowing if it is doing so is by analyzing the feature impacts on a data point level, and accordingly certain features can be masked from prediction.

LIME Analysis:

Local surrogate models are interpretable models that are used to explain individual predictions of black box machine learning models. Local interpretable model-agnostic explanations (LIME) is a paper in which the authors propose a concrete implementation of local surrogate models. Surrogate models are trained to approximate the predictions of the underlying black box model. Instead of training a global surrogate model, LIME focuses on training local surrogate models to explain individual predictions.

To explain in short, LIME works in the following way -

- Take a data point(of dimension say f, or we can say we have f features) and make some copies of this data point, say N.
- Introduce small random jitterations to s subset of features to each of the N duplicates so that they are now different from one another yet nearby in the f dimensional space.
- We now have (N*f) data points, we fit a Ridge regression/classification model on it to get coefficients for each of the f features. The data is trained against the output predicted by whatever model we used for prediction.
- We multiply each feature original value by its coefficient to get the impacts.



Feature	Value
business_legacy_stars_mean	3.51
user_legacy_stars_mean	2.97
compliment_writer	0.00
BusinessAcceptsCreditCards_NA	0.00
RestaurantsPriceRange2_2	1.00
yelping_since_td	3260.00
BikeParking_NA	0.00
compliment_note	0.00
Caters_NA	0.00
compliment_funny	0.00
compliment_cool	0.00

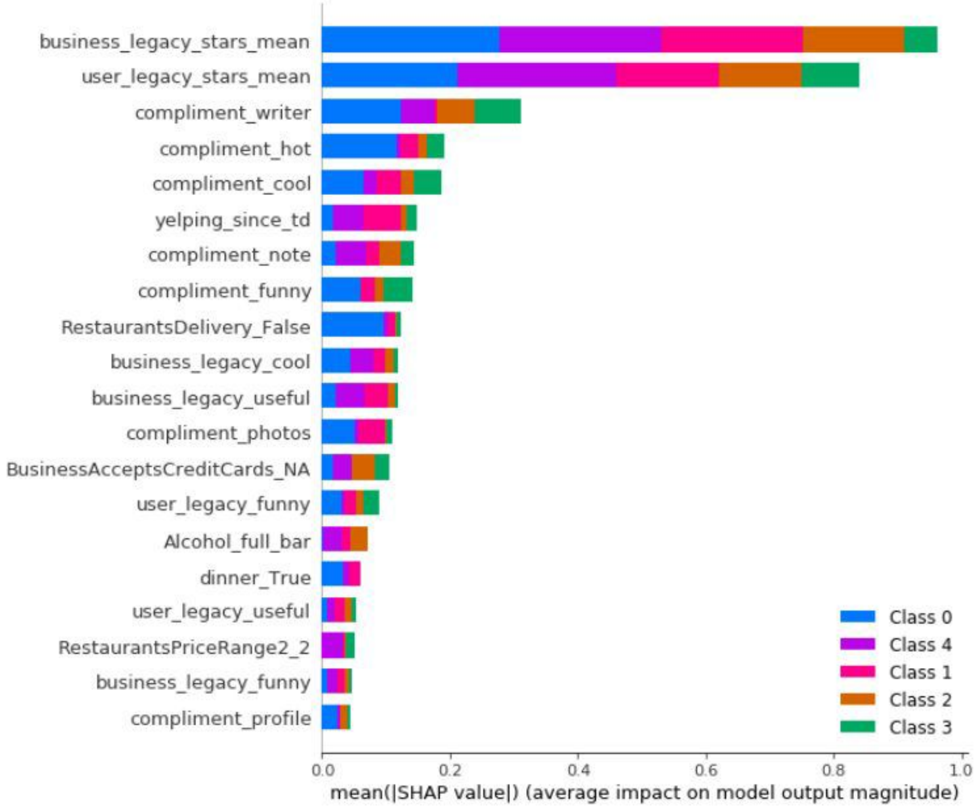
From the above diagram and impacts, it is a fair bet to say that had the business not accepted credit cards, it might not have got a 5 star rating. Hence he has to continue doing that. We can also say that price category of a restaurant is an important factor for this particular user's potential star rating.

SHAP analysis

SHAP (SHapley Additive exPlanations) by Lundberg and Lee (2016) is a method to explain individual predictions. SHAP is based on the game theoretically optimal Shapley Values.

The goal of SHAP is to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the "payout" (= the prediction) among the features. A player can be an individual feature value, e.g. for tabular data. A player can also be a group of feature values. For example to explain an image, pixels can be grouped to super pixels and the prediction distributed among them. One innovation that SHAP brings to the table is that the Shapley value explanation is represented as an additive feature attribution method, a linear model.

The following is an example of SHAP analysis made on the first 10 reviews-



We can see from the impact chart that, if a restaurant does not provide delivery, it is a high contributer to 0 star rating.

Conclusion

- Memory based CF is too simple and has poor performance, and is only driven by the users previous ratings and not business attributes. It is also prone to cold start problem.

- Where as our two Embedded NN models ,which improve upon the memory based CF, one by learning latent features using neural networks and the other while using the similar neural network architecture ,considers a whole lot of other features which are specific to user,business.

- However these models required high computation ,which might not worth the cost given the revenue addition due to increased performance.

- The NN models can also be improved upon by further hyper parameter tuning of the neural network.

- As an addition to these Embedded NN models, the text related data can be used more extensively using word embeddings and a sequential model.

References

[1] Collaborative filtering using collab | FastAI

[2] Various Implementations of Collaborative Filtering

[3]Collaborative filtering with fastai

[4] LIME and Shap