User Profiling through Multimodal Fusion

Akshay Singh Rana
Université de Montréal
akshay.singh.rana@umontreal.ca

Harmanpreet Singh
Université de Montréal
harmanpreet.singh@umontreal.ca

Himanshu Arora
Université de Montréal
himanshu.arora@umontreal.ca

Nitarshan Rajkumar
Université de Montréal
nitarshan.rajkumar@umontreal.ca

Sreya Francis
Université de Montréal
sreya.francis@umontreal.ca

ABSTRACT

In this work we investigate the task of social network user profiling. Specifically, the goal is to predict a user's age, gender, and psychological attributes, given a set of text, image, and social-graph features from that user. We apply a range of machine learning algorithms to this problem, including Linear Regression, Decision Trees, and methods such as Ensembles, Node Embeddings, and Chaining. We are ultimately able to beat a pre-specified baseline for each of these classification and regression tasks using a combination of various techniques.

1 INTRODUCTION

Most of today's users actively generate content in multiple online social media platforms like facebook, twitter and reddit. User profiling by inferring the age, gender and personality traits of these users play a major role in providing different services ranging from personalized systems to marketing and recommendation services. The personality traits obtained from the user generated content can help in developing the above stated services which can aid in improving user experience to a great extent.[2, 11]

User-generated content is rich in modality, varying from images and videos, textual public posts and (perhaps) more informal private messages, to sharing of posts, and liking of pages and posts. Machine learning algorithms which would be used to build user profiles given such information tend to be suited for single modalities, otherwise requiring extensive feature engineering to incorporate multi-modal data sources together. However, doing so can help to develop much better personalized system/service than when independently working with separate data modes.[3]

In our work, we analyze the characteristics of social user and techniques which model and update a tag-based profile. We analyze how to treat social annotations and the utility of modelling tag-based profiles for basic predictions of user characteristics such as age, gender and personality. The problem statement at hand is user modeling with multi-source user data such as text, images, and relations to arrive at accurate user profiles. The goals of this study is to build a system for automatic recognition of the age, gender, and personality of social media users which when given as input user generated content, should return as output the age, gender and personality trait scores specific to that user.

To be more specific, our system predicts the following information about the user as output:

- gender, as either "male" or "female"
- age, as either "xx-24", "25-34", "35-49", or "50-xx"
- personality, as a score between [1, 5] for each of the five traits of the Big Five personality model, namely Openness, Conscientiousness, Extroversion, Agreeableness, and Neuroticism

Predicting gender is a binary classification learning task, whereas predicting age is a multi-class classification task. Predicting the personality scores corresponds to solving five regression tasks. For the classification tasks on age and gender, we got the best results with Linear Support Vector Machines (SVM) and eXtreme Gradient Boosting (XGB). For the regression tasks on the traits Openness to experience, Conscientiousness, Extroversion and Agreeableness, we got the best results with Lasso Linear Regression whereas the best performance on the trait Emotional Stability (Neuroticism) was achieved with XGB.

Optimal feature selection and feature engineering helped us in further improving our results wherein early fusion of the features proved to be better than late fusion. In short, for most of the cases we got better results with simple linear algorithms than the non-linear ones. Our best results could be found in table 2 under the results section.

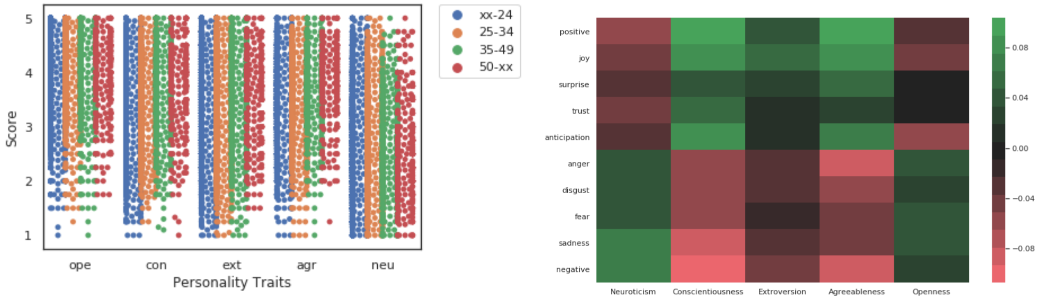
In this paper, we are mainly interested in multi source user data based user profile modelling. In Section 2, we present the various datasets used for the modelling as well as the evaluation metrics employed. In Section 3, we show the various techniques experimented on feature selection as well as extraction. In Section 4, we present the results obtained along with the details on all the approaches employed to attain the specified results. In Section 5, we present the conclusion as well as our perspective on future scope for our work.

2 DATASETS AND METRICS

The original datasource for user data is composed of Facebook status updates, profile photos and page likes. Unfortunately due to legal reasons we were unable to directly use the status updates and profile photos, and were instead provided three derived datasets: LIWC [8] and NRC [7] features (for status updates), and Oxford features (for profile photos). These datasets provide information on 9500 users.

2.1 Text Features

The Linguistic Inquiry and Word Count (LIWC) dataset uses the LIWC method to extract features from text. This is considered one of the standard methods for computerized text analysis. Most of the LIWC output variables represent percentages of total words within a text. For example the variable for Positive Emotions (posemo) represents the percent of words in a text that were positive emotion words. Some of the other LIWC features include anger, health, insight, motion, sadness, filler, Period, Comma, Colon, number, swear, social, family, friend, present, future, adverb etc. A few of the LIWC variables are calculated differently from just proportions. The first is word count (WC) – which is just the raw number of words within a file. The second is words per sentence (WPS) which is the mean number of words within each sentence within the file.



(a) Correlation between personalities traits and age

(b) Correlation between posts' sentiment and personality traits

Figure 1

The NRC dataset consists of emotion lexicons which capture nuances of emotion and can be used to identify personality. Lexical categories corresponding to fine-grained emotions such as excitement, guilt, yearning, and admiration are significant indicators of personality. Personality has a known association with emotion. Emotions are considered to be more transient phenomenon whereas personality is more constant. Earlier there were small lexical categories pertaining to a few basic emotions such as anger, joy, and sadness. However, personality detection can benefit from a much larger lexical database with information about many different fine-grained emotions which is what NRC is to be exact. Some of the main NRC Emotion Lexicon features include positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise and trust.

2.2 Image Features

The Oxford dataset consists of facial point features extracted from profile photos. Some of the face-related attributes included in the dataset are: face width, eye positions, lip positions, facial hair presence and shape, and other numeric attributes of the position, alignment, and description of the face.

In the provided dataset, 2326 users had missing Oxford features and there were 741 duplicate user entries.

2.3 Relational Features

Graph sources include User-Page-likes which is based on user page profiling. The relation profile includes list of user ids along with their corresponding like ids. There are a total of 9500 users with the total number of features being 65 for Oxford, 1 for Relationships, 81 for LIWC and 10 for NRC.

2.4 Metrics

The predictive capabilities of our system was tested on a "hidden" test dataset of n = 1334 users. This set contains the 334 users from the public test dataset, as well as 1000 new users who are not in training dataset nor in the public test dataset.

The metrics employed for evaluation of age and gender classification is the classification accuracy which is the ratio of number of correct predictions to the total number of predictions.

$$Accuracy = \frac{\text{number of correctly classified instances}}{\text{total number n of instances}}$$

The prediction of five personality traits namely Openness, Conscientiousness, Extroversion, Agreeableness and Neuroticism were evaluated based on the root mean square error(RMSE) which is the standard deviation of prediction errors/residuals.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

with y_i the actual value and $\hat{y}_i$ the predicted value.

3 METHODOLOGY

3.1 Exploratory Data Analysis

As a first step, we perform exploratory data analysis on all of the given datasets. This is an important step, as often, optimal feature pre-processing and engineering is the dominant factor in getting high performance with a machine learning model. We find many useful correlations between the variables as part of this analysis. For instance, we find that the age of the users is positively correlated with positive personality. In other words, as the users age, there is less chance of a low score on OPE, CON, EXT, AGR, and a high score on NEU. Also, the more negative the users' posts were, the more open to experience they were and the more pages they had liked.

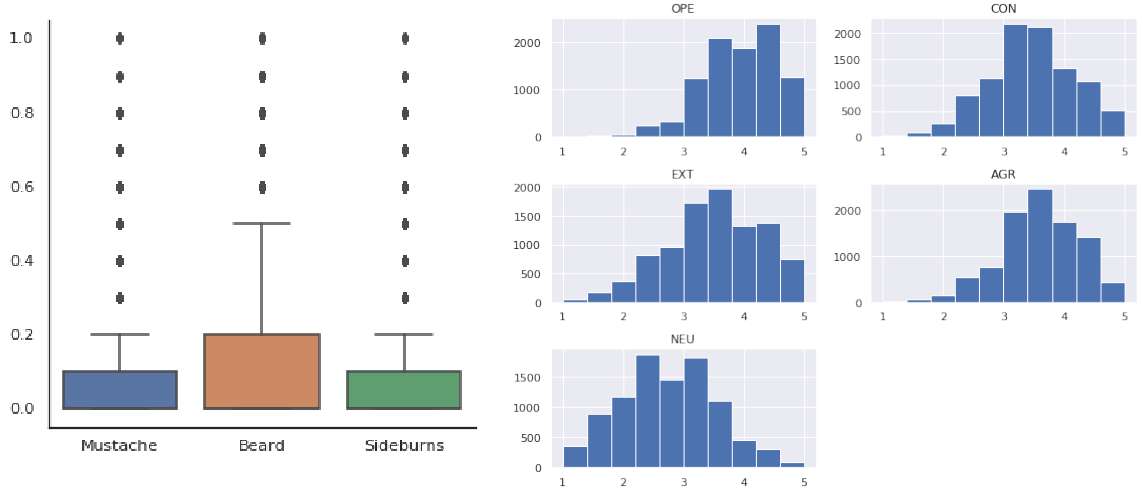
3.2 Feature Preprocessing

3.2.1 *Outlier Removal.*

Feature preprocessing can drastically affect the performance of a machine learning model. Although, some algorithms are robust to outliers, it is still recommended to remove them. We detect and remove the outliers from all the features using the interquartile range (IQR) method. Specifically, we find the IQR of the feature in consideration by subtracting the value at the 75^{th} percentile with the value at the 25^{th} percentile. Then, using a threshold value of 1.5, we clip all the values below $-1.5 \times IQR$ and above $1.5 \times IQR$ to the respective values.

3.2.2 *Feature Normalization.*

Most of the machine learning algorithms need feature normalization to work properly. Especially the ones that need to calculate distance between training samples or use a gradient-based optimization technique. Min-Max normalization and standard or



(a) Outlier Detection

(b) Distributions of personality trait targets

Figure 2: Feature Pre-processing (a) and (b)

z-score normalization are the two most commonly used feature normalization methods. Min-Max normalization scales all the original values to values between 0 and 1 whereas standard normalization makes the feature values zero-centered with a standard deviation of one. We performed min-max normalization on all the continuous-valued features in the datasets.

3.2.3 ***Target Normalization***. Similarly to input features, the distributions of the targets could also vary, requiring some form of normalization. For example, we can see in Figure 2b that the distributions of the personality traits are all different, and skewed in their own ways. We can make use of Quantile Standardization to correct this, by converting each value into the quantile at which that value is present for the feature. For example, a value of 3.5 for *EXT* would convert to roughly 0.5, and 5 would convert to 1. This allows for the regression target to be uniform over (0, 1).

3.2.4 ***Missing Values***. The Oxford features are not present for all users (all other data sources are complete in this regard), which poses a problem for incorporating this information with the other data sources. One strategy could be to use algorithms such as XGBoost which have support for branching on missing values. Another simpler choice would be to fill in missing values with averages for the column, and to also create a new feature which indicates whether the Oxford data for a user was originally missing or not.

3.3 Feature Selection

The three most common techniques to identify important features for a specific task are filter, wrapper, and embedded methods. The filter method uses correlation as a metric to compare the importance of features and removes a subset of redundant highly correlated features. Wrapper methods start with a subset of features and iteratively add or remove features depending on the performance of the model on the given set of features. Embedded methods extract implicit feature importance from algorithms such as lasso and random forests. We tried all of these and report the most frequently observed important features for all the tasks in this table.

Table 1: Important features using feature selection methods

Oxford	LIWC	NRC
facialHair_mustache	ipron	negative
facialHair_beard	swear	anger
facialHair_sideburns	disgust	Oxford
faceRectangle_width	negemo	fear
faceRectangle_height	feel	joy

3.4 Feature Extraction

In this section, we discuss how we represent each data source for the task of user modelling in social media. We define three data source embeddings: a data source embedding from the textual content, a data source embedding from the visual content, and a data source embedding from the relational content.

3.4.1 **Textual Feature Representation.** Textual data source embeddings for each user were represented with 81 Linguistic Inquiry and Word Count (LIWC) and 10 NRC Emotion Lexicon features. We concatenate these NRC and LIWC features horizontally to obtain data source representation from text.

3.4.2 **Facial Feature Representation.** For each user, visual feature representation is acquired with 65 facial features fetched using the Oxford API. The extracted facial features are face rectangle features to capture the location of the face in the image, face landmark features which include 27-point face landmarks pointing to the important positions of face components, face characteristics including age, gender, facial hair, smile, head position and glasses type.

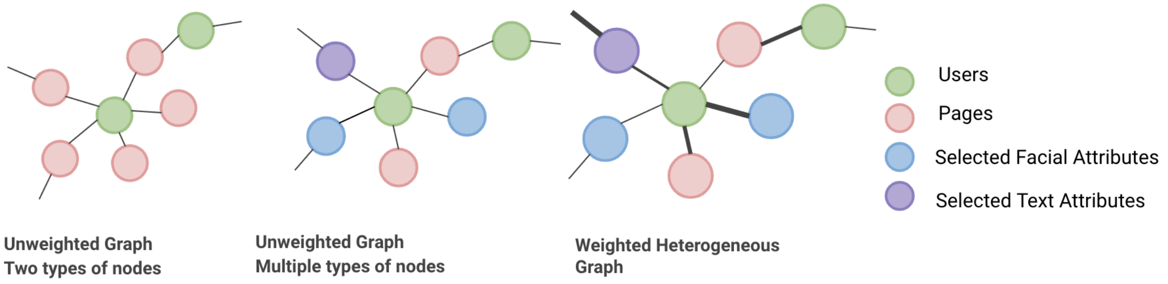


Figure 3: Node2vec Heterogeneous Graph Formulation

3.4.3 **Relation Feature Representation.** We investigate multiple feature representations approaches for relational data as discussed in the subsequent sections.

Multi-Hot Vectorizer: Simplest approach to extract relational data features is to use multi-hot user-like adjacency matrix. This approach is inspired from the Bag of Word approach used heavily in text analysis. As most users typically have a very small subset of the liked pages, the resulting matrix have many feature values that are zeros introducing sparsity.

Tf-Idf Multi-Hot Vectorizer: In a huge user-like network, some pages are generally liked by lots of users, hence carrying very little meaningful information to encode the actual user. In downstream tasks, these frequent pages would shadow the frequencies of rarer yet more interesting pages. The way to combat this problem is to use Tf-Idf. It re-weights the count features into floating point values by normalizing them to provide better representation. Although this method is simple and efficient to encode relations, it produces a very large dimensional discrete vector making it harder to generalize in statistical learning. Dimensionality reduction techniques such as PCA can be deployed to learn dense representation, but they suffer from both computational and statistical performance drawbacks. Moreover, these methods optimize for objectives that are not robust to the diverse patterns observed in networks.

Node2vec Embeddings: Prediction tasks over nodes and edges in networks require careful effort in engineering features used by learning algorithms. In networks, the conventional paradigm for generating features for nodes is based on feature extraction techniques which typically involve some seed hand-crafted features based on network properties. Therefore, this motivated to automate this process of learning a mapping of nodes to a low-dimensional continuous space of features that maximizes the likelihood of preserving network neighborhoods of nodes. Node2vec [4] and DeepWalk [9] frameworks efficiently learn task-independent social representations of graph's vertices, by using sampling strategy to model a stream of short random walks. These methods extend the Skip-gram [5, 6] architecture to graph networks to learn high-quality distributed vector representation. DeepWalk is based on unbiased random walk sampling strategy and can be seen as a special case of Node2vec.

In this work, we perform various experiments by formulating different kinds of heterogeneous graph setups as shown in Figure [3], based on the user attributes to learn Node2vec embeddings. Graph formulation vary from considering just the user and page like relations to extending it to user, page likes, selected facial and text attributes [12]. Feature selection of facial and textual attributes was done by combining filter and embedded based filter selection approaches. Furthermore, this approach of modelling complex heterogeneous graph was extended by incorporating weights to vertices by normalizing them using tf-idf transformation. This helps us jointly consider weighted heterogeneous attributes or contents (e.g., text or image) associated with each node to provide us a enriched representation. The weights for users' facial and text edges were fetched from the attribute values in data-set. We represent users with pages that they like as well as their attributes, and find similar users by a flexible biased random walk procedure to explore neighborhoods in both Breadth-First Sampling (BFS) as well as Depth-First Sampling (DFS) fashion. BFS investigate nodes which are immediate neighbors of the source, while DFS explore nodes at increasing distances from the source node. We iteratively execute random walks on the graph to sample neighbors for each node based on Node2vec based walk strategy, and then train a skip-gram architecture to learn representation for each node.

We find the best hyperparameters to learn good relational data representation by performing multiple Node2vec experiments. We evaluate the quality of embeddings by assessing the performance of downstream tasks. With the limited computational power available to us, our results were produced with the embedding dimension size of 128, number of random walks of 20 and walk length of 80. We chose return hyperparamter (p) value as 0.8 and inout hyperparamter (q) as 0.9 to prioritize BFS over DFS. Skip-gram architecture model was trained with context window size of 10 for 10 epochs.

3.5 Algorithms

3.5.1 ***Lasso Regression.*** Lasso Regression is an extension of Linear Regression, which also adds an L1 regularization term on the weights of the model. This serves the purpose of pushing weight values to zero, acting as a form of feature selection. The similar ridge regression uses L2 regularization, which does not as effectively push weights to 0 and thus cannot effectively perform feature selection as well.

3.5.2 ***Support Vector Machines.*** A Support Vector Machines (SVMs) is another method which seeks to find a linear decision boundary between data points, but makes use of a unique Hinge loss function that guarantees that if a linear decision boundary exists, the algorithm will find the maximum-margin boundary (i.e. the boundary which is of greatest distance from all points). Though there are kernel methods which allow SVMs to find non-linear decision boundaries, we restrict our use to only the linear formulation for computational reasons.

3.5.3 ***Random Forests.*** It is the most commonly used bagging method in machine learning. Random forests use a modified decision tree learning algorithm that selects, at each candidate split in the learning process, a random subset of features in the data. It is a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model.

3.5.4 ***Gradient Boosted Decision Trees.*** Gradient Boosted Decision Trees (GBDTs) extend on forests. It is slightly different from the adaptive boosting in the way the weak learners are trained. Rather than changing the sampling distribution based on the performance of a weak learner, it trains the subsequent weak learners on the errors of the prior learners. A more common form of gradient boosting XGBoost, also known as eXtreme Gradient Boosting, essentially works on the same principle but it is more regulaized which reduces overfitting and increases the generalization performance. The XGBoost library also supports missing values in data, which allows us to avoid preprocessing on missing face data for users.

3.6 Ensembling Methods

3.6.1 ***Chaining.*** Another way we can leverage multiple models is via chaining together regressors or classifiers. This entails training a model on each target feature, but with successive models taking as input the output predictions of earlier models. For example, we could train a model for the age and gender for a user, and then another model for the personality traits, which takes the age and gender predictions as additional input features. We can see in Figure 4 that there exist some clear correlations between these personality traits, which a model might be able to leverage if regressing on them sequentially vs independently.

3.6.2 ***Stacking (Late Fusion).*** We can also leverage multiple models via stacking, in which models are independently trained on the problem, with the addition of a new meta-model which takes as input the predictions of the component models, before outputting the final predictions for the ensemble as a whole.

3.6.3 ***Feature Concatenation (Early Fusion).*** In addition to leveraging multiple classifiers, we can also attempt to leverage multiple data sources together. In the given problem, we are provided text, image and relation datasets, and could naively learn models using these datasources independently, or can instead concatenate the data together (after appropriate per-source pre-processing), and train models that take as input all the combined features at once.

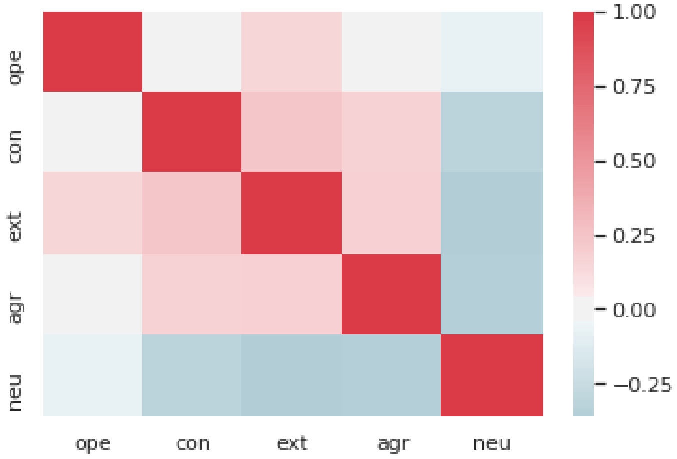


Figure 4: Correlation between each personality trait

4 MODEL ARCHITECTURE

Integrating two or more sources of data to form a unified picture or make a better decision are the main goals of data integration frameworks [1–3, 10, 11] . We take motivation from these papers to perform all sorts of experiments to reach a successful result. Feature selection on LIWC, NRC and Oxford features was done using wrapper and filter methods. Later, these filtered features were transformed into an adjacency matrix utilized to learn Node2vec representation. This matrix consists of users as rows, and selected attributes along with relations as columns. This graph is later trained to get embeddings of 128 length vector for both users, pages as well as other attribute nodes. The user embeddings if used directly gives good results, but since we may encounter new users while testing as the graph is incomplete, we cannot use user embeddings directly. As we might have to retrain the graph at inference when adding more users to get their embeddings, it won't be feasible. Therefore, we averaged the page embeddings for the pages liked by users in the train dataset. We maintain a page vocabulary for pages with more than 5 users liking it.

We have 70 text features, 65 oxford image features and 128 length embeddings from the graph. From Table 2, in gender classification, we concatenated these three features and used eXtreme Gradient Boosting to predict gender for best results. For age prediction, we used the predicted value of gender joined along with other features using Linear SVM. The age values are bucketed into four buckets i.e. "xx-24", "25-34", "35-49", or "50-xx".

For personality trait prediction, the outputs were in the range of 1-5 and we used regression chaining where the prediction for one personality trait is again fed back and used for predicting the another personality trait using XGB Regressor or Lasso Regression. The predictions from gender and age were added with the concatenated input sources to predict the first personality trait. The predicted value was added back to the input source to predict next personality task. The order for chaining personality traits selected after multiple experiments was Openness, Agreeableness, Conscientiousness, Neuroticism and Extroversion. Table 2 summarizes

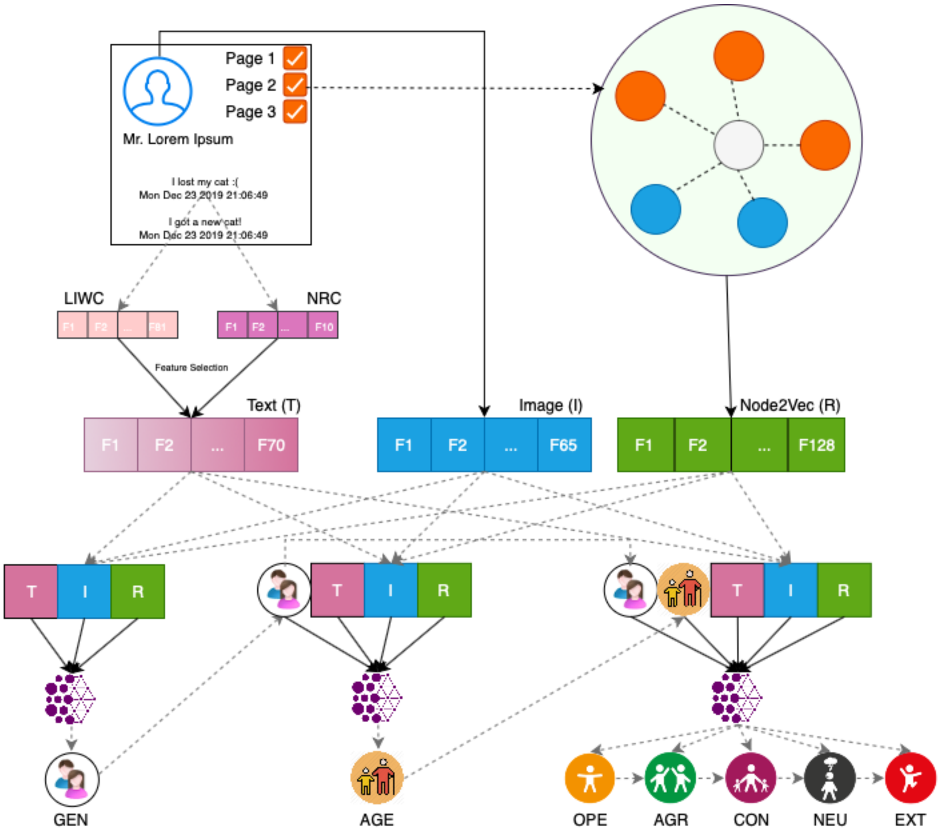


Figure 5: Multi modal fusion architecture

our best scores using these methods.

We tried to follow the above approach repeatedly where the predictions at time t_1 would collectively be fed back again to the system to generate other set of predictions at time t_2. We observed that this doesn't really improve our scores and after many epochs, this could result in poorer scores. However, we believe that this approach can be helpful if used with deep neural networks. We explored different machine learning algorithms at each stage of prediction and found that Linear SVM, Lasso Regression and eXtreme Gradient Boosting to outperform all the other methods. Deep Neural networks with hyperparameter tuning can improve the results further.

5 RESULTS

In this section, we present the approaches that performed the best among the several methodologies that we tried. In Table 2, we report the performance of the methods that we applied on the validation set of our dataset using a training-validation set split of 80-20. We find that for most of the tasks, simple linear algorithms such as linear regression and linear SVMs performed better than non-linear algorithms such as decision trees and deep neural networks. This also means that the performance achieved is largely dependent on optimal feature selection and engineering. Except for age classification, all the tasks were learned better when features from all the three datasets were used. Age classification was learned better without using the text features. For all of the tasks, early fusion of the selected features always gave better results than late fusion. To get these results, we tried both single-modality models and multi-modal models with early fusion of the modalities.

To begin with, we trained each of the gradient boosting trees, linear regression, and linear support vector machine algorithms using a single data source for all the prediction tasks. We found that image features work well for predicting gender of the users but not the age and the personality. For gender, this could be because of the facial hair features as identified as important features for this task using embedded feature selection methods. A picture of a user might be able to predict the momentary mood of the user, but might not be useful to predict the user's enduring personality. For age, we might need to have a better representation of the image which clearly reflects the age difference e.g. using convolutional neural networks on raw images, the filters could learn these differences more accurately. Text features turned out to be useful for personality prediction but not for age and gender. This seems intuitive as certain words strongly suggest a particular personality trait but there are no gender or age specific words. Using page-like information turned out to be the most useful single modality for all prediction tasks. This could be because page-likes implicitly creates clusters of similar users which might encode a lot of useful information to predict information about them. We tried to improve this accuracy further by using features extracted by Node2Vec, but that didn't work well when used as a single source.

Table 2: Accuracy/RMSE scores using early fusion. XGB used Gradient Boosted Decision Trees from the XGBoost library, LR uses Lasso Regression from the sklearn library, and SVM uses Linear SVM Classifiers from the sklearn library

Data	Model	Gender (Acc)	Age (Acc)	Opn (RMSE)	Neu (RMSE)	Ext (RMSE)	Agr (RMSE)	Con (RMSE)
Baseline (Test)	—	0.551	0.626	0.623	0.778	0.778	0.648	0.701
ONE SOURCE								
Text (T)	XGB	0.551	0.626	0.623	0.778	0.778	0.648	0.701
	LR	-	-	0.625	0.779	0.778	0.648	0.705
	SVM	0.551	0.614	-	-	-	-	-
Images (I)	XGB	0.810	0.615	0.631	0.778	0.823	0.665	0.728
	LR	-	-	0.634	0.779	0.819	0.669	0.726
	SVM	0.681	0.614	-	-	-	-	-
Sparse Relations (R)	XGB	0.773	0.634	0.621	0.773	0.791	0.650	0.712
	LR	-	-	0.623	0.776	0.805	0.657	0.710
	SVM	0.785	0.675	-	-	-	-	-
Node2Vec (N2V)	XGB	0.558	0.613	0.609	0.781	0.787	0.652	0.708
	LR	-	-	0.602	0.778	0.770	0.647	0.702
	SVM	0.560	0.617	-	-	-	-	-
TWO SOURCES								
T + I	XGB	0.837	0.633	0.618	0.785	0.792	0.650	0.701
	LR	-	-	0.631	0.791	0.804	0.656	0.716
	SVM	0.797	0.615	-	-	-	-	-
T + N2V	XGB	0.831	0.619	0.611	0.785	0.783	0.649	0.710
	LR	-	-	0.615	0.780	0.785	0.644	0.709
	SVM	0.855	0.713	-	-	-	-	-
I + N2V	XGB	0.849	0.674	0.611	0.786	0.779	0.646	0.707
	LR	-	-	0.600	0.780	0.769	0.640	0.700
	SVM	0.855	**0.713**	-	-	-	-	-
THREE SOURCES								
I + T + N2V	XGB	**0.863**	0.686	0.609	**0.777**	0.787	0.641	0.702
	LR	-	-	**0.599**	0.779	**0.764**	**0.638**	**0.697**
	SVM	0.856	0.708	-	-	-	-	-

We selected certain combinations of two and three modalities to train multi-modal models. Interestingly, while Node2Vec features didn't perform as good as other data sources when used alone, they significantly improved the accuracy of the models when combined with text or image features. In other words, Node2Vec features provided the complementary set of information for either image and text features required to accurately predict the target variables. However, combining all three of them didn't seem to improve the accuracy much.

6 CONCLUSION AND FUTURE WORK

We explored and leveraged many feature-preprocessing techniques, machine learning models, and ensembling and data fusion techniques in order to beat the provided baselines for each of the user profile classification and regression tasks provided. The key takeaway from this has been that early fusion of disparate data sources can allow a model to learn better representations than from the independent datasets. We found this to provide better model performance than was possible merely from using higher complexity models and ensembles.

Due to the largely preprocessed nature of the text and image features, we were able to leverage simple models such as Linear Regression and Linear SVMs to great effect, largely beating out the use of the tree-based methods we explored as well. We did not find time to deeply investigate the use of neural network approaches, though initial results were not promising, and would have required extensive hyperparameter tuning (already an issue for tree-based methods) to explore further.

This work was hampered in a key way as we never had access to the underlying text and image data, and only had derived features of it. This prevented us from exploring very promising avenues such as building text embeddings, sophisticated models such as BERT, as well as face detection and analysis using Convolutional Neural Networks. These methods would have both reduced the need for extensive feature-engineering as the NRC, LIWC and Oxford datasets entail, and would have been able to learn more complex features in the data. Additionally, there is much recent work on combining multi-modal data and using deep learning models to successfully learn from the joint representation of the data.

With respect to the user-page relationship graph, we merely used the simple Node2Vec technique, which is problematic in that we were only able to generate embeddings for nodes which we had seen, and could not generate new embeddings for unseen test data. Given more time, we would have liked to explore newer techniques such as Graph Convolutional Neural Networks, that are able to generalize to unseen nodes.

ACKNOWLEDGMENTS

Akshay contributed to training models on single and multiple data sources using different machine learning algorithms. Himanshu contributed by preprocessing the data and helped in visualization. Harmanpreet contributed to Node2Vec. Nitarshan contributed to exploring chaining and stacking of machine learning models, and performance of these models on combined data sources. Sreya contributed to performance analysis of basic classification/regression algorithms.

This paper was jointly written and edited by all the authors.

REFERENCES

[1] Grigory Antipov, Sid-Ahmed Berrani, and Jean-Luc Dugelay. 2016. Minimalistic CNN-based ensemble model for gender prediction from face images. *Pattern recognition letters* 70 (2016), 59–65.

[2] Golnoosh Farnadi, Geetha Sitaraman, Shanu Sushmita, Fabio Celli, Michal Kosinski, David Stillwell, Sergio Davalos, Marie-Francine Moens, and Martine De Cock. 2016. Computational personality recognition in social media. *User modeling and user-adapted interaction* 26, 2-3 (2016), 109–142.

[3] Golnoosh Farnadi, Jie Tang, Martine De Cock, and Marie-Francine Moens. 2018. User profiling through deep multimodal fusion. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 171–179.

[4] Aditya Grover and Jure Leskovec. 2016. node2vec. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16* (2016). https://doi.org/10.1145/2939672.2939754

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. arXiv:cs.CL/1301.3781

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[7] Saif Mohammad and Svetlana Kiritchenko. 2013. Using nuances of emotion to identify personality. In *Seventh International AAAI Conference on Weblogs and Social Media*.

[8] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

[9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14* (2014). https://doi.org/10.1145/2623330.2623732

[10] H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one* 8, 9 (2013), e73791.

[11] Wu Youyou, Michal Kosinski, and David Stillwell. 2015. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences* 112, 4 (2015), 1036–1040.

[12] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. ACM, New York, NY, USA, 793–803. https://doi.org/10.1145/3292500.3330961

User Modelling

We tried multiple approaches and the final model is in the model_final.py where we have combined all three different modalities.

All the below features were normalized using Min Max Scaling. We also used SelectKBest methods to select the top features for face and text data.

1. Face Data (Imputed mean values for users with no face data and used the first face when a user had multiple face data)
2. Text Data (Concatenated the liwc and nrc data together after selecting the best features among them)
3. Relation Data (got node2vec embeddings for pages with more than 5 users, and averaged the page embeddings for each user), we could also use user embeddings but then we would have to retrain the graph while testing which wasnt feasible.

```
X = pd.concat[face, text, relation_n2v]
```

Gender Classification:
All the above three data sources were concatenated and trained using XGBoost to predict gender.

Age Classification:
All the above three data sources were concatenated alongwith the prediction for gender and then trained using XGBoost to predict age.

Personality Prediction:
All the above three data sources were concatenated alongwith the prediction for gender and age.
We used regression chaining where the prediction for one personality trait is again fed back and used for predicting the another personality trait using XGBRegressor.
The below order for chaining was selected after multiple experiments
[openness,]

```
> python model_final.py
```