SJSU SAN JOSÉ STATE UNIVERSITY

Project Report
CMPE 256 Large-Scale Analytics

HappyDB: Work-Life-Balance during COVID19

Dated: 04/29/2020

Submitted to Prof. Magdalini Eirinaki

Submitted by
Ausaf Ahmed | 013744315
Gayathri Ganesh | 013804674
Osama Soliman | 012659985

Table of Contents

Chapter 1 - Introduction

Motivation

The science of happiness is an area of positive psychology that studies the factors that sustain people's happiness over time (Seligman, 2011; Fredrickson, 2009; Lyubomirsky,2008). One of the interesting findings of the field (Dieneret al., 1999) is that while 50% of our happiness is genetically determined, and only 10% of it is determined by our life circumstances (e.g., finances, job, material belongings),40% of our happiness is determined by behaviors that are under our control. Examples of such behaviors include investing in long-term personal relationships, bonding with loved ones, doing meaningful work, and caring for one's body and mind. Consequently, positive psychologists have focused on devising methods to steer people towards those behaviors. Fostering happiness has also received attention at the national policy level – in a recent interview (Murthy, 2016) the U.S. Surgeon General claimed that fostering happiness is an important priority as one of the main ways to prevent disease and live a longer, healthier life[1]. This paper describes HappyDB and its properties and outlines several important NLP problems that can be studied with the help of the corpus. Also, this paper tries to incorporate the dataset as a reference for Work-Life-Balance in the time of a pandemic as COVID19. Our results demonstrate the need for deeper NLP techniques to be developed which makes HappyDB an exciting resource for follow-on research.

Objective

With the outbreak of the COVID 2019 pandemic, many of us are advised or forced to work from home full-time. These circumstances are stressful, and a good work-life balance becomes even more difficult to maintain. Our project aims to study the factors contributing to the happiness quotient, and the underlying relationship of demographics with different categories of happiness. Backed by Machine learning, with a focus on Natural Language Processing technique, the approach can eventually identify the stress alleviation elements and subsequently rebalance our lives.

Chapter 2 – System Design & Implementation Details

Our project dataset considers into two subsets primarily, text and demographic dataset. One of the most challenging and riveting aspects of the project was to identify how to use the two subsets spread across two different domains of datatypes. Either we could club all the features together, or we could perform meta-modeling and fuse the independent results to compute the *final* result. For the scope of this project, we proceeded with the former approach, necessarily, converting the text + demographics features into a numerical format and combine them to form a *super* vector.

Algorithms Considered

The scope of the project covers the following major algorithms:

- **TF-IDF(Term Frequency Inverse Document Frequency):** The fundamental step in text processing is transforming raw text to numbers, otherwise also known as text vectorization that Machine Learning algorithms can understand. The idea sets the premise of feature engineering in Natural Language Processing applications. A bag-of-words model, or BoW, is a way of extracting features from the text that relies on the vocabulary of known words and considering each word count as a feature. Although BoW provides a similarity comparison, it fails to account for the relevancy of a word in a corpus. Our project uses TF-IDF (term frequency-inverse document

frequency) for feature extraction to address the relevancy constraint. TF-IDF (term frequency-inverse document frequency) is a statistical measure that assesses how important a word is to a document in a document collection. It is achieved by combining two metrics: how many times a word appears in a document, and the word's inverse frequency of a document across the collection. It has a wide range of applications in Information Retrieval and plays a crucial role in scoring words across Natural Language Processing tasks.

Mathematically, TF-IDF for a word t in document d from the document collection D can be expressed as follows:

$$tfidf(t, d, D) = tf(t, d) \,.\, idf(t, D)$$

where:

$$tf(t, d) = log(1 + freq(t, d))$$

$$idf(t, D) = log(\frac{N}{count(d \in D : t \in d)})$$

- **Support Vector Machines:** SVM is a popular supervised machine learning algorithm that distinctly classifies the data points by finding an optimal hyperplane in N-dimensional space, where N = number of features. We wanted to try SVM due to its effectiveness in high dimensionality and compute-intensive footprint. However, it underperformed for our use case as SVM is not suitable for large, sparse datasets that accounts for lack of a clear margin of separation between classes.

- **Decision Trees:** A decision tree is a map of the possible outcomes of a series of related choices. The goal is to construct a model that predicts the value of a target variable by learning basic rules of decision inferred from the data characteristics. A major benefit of a decision tree is that it allows all possible consequences of a decision to be considered and maps each direction to a conclusion.

- **Random Forests:** Random forests is an ensemble method combines multiple independent decision trees resulting in a stronger classifier. Our engineered feature vector primarily composes features across text and demographics, which resulted in many *decisions*. Hence, random forest algorithm improvises on bagging by decorrelating these weak decision trees on a random subset of features. Hence, not only random forests handled the collinearity in features implicitly, but also dealt with the relative importance of the features, especially the highly correlated ones.

Technologies and Tools

- Google Colaboratory to leverage the power of Google Cloud. Using Colab notebooks, we could realize the value of peer programming without heavily relying on versioning and dependency overheads.
- Scikit-learn, an arsenal library providing wrappers to feature engineer, implement algorithms, and, evaluate and optimize the model.
- NLTK, natural language toolkit to use its interfaces for language modeling tasks.
- Matplotlib, Seaborn for visual analysis and plotting.

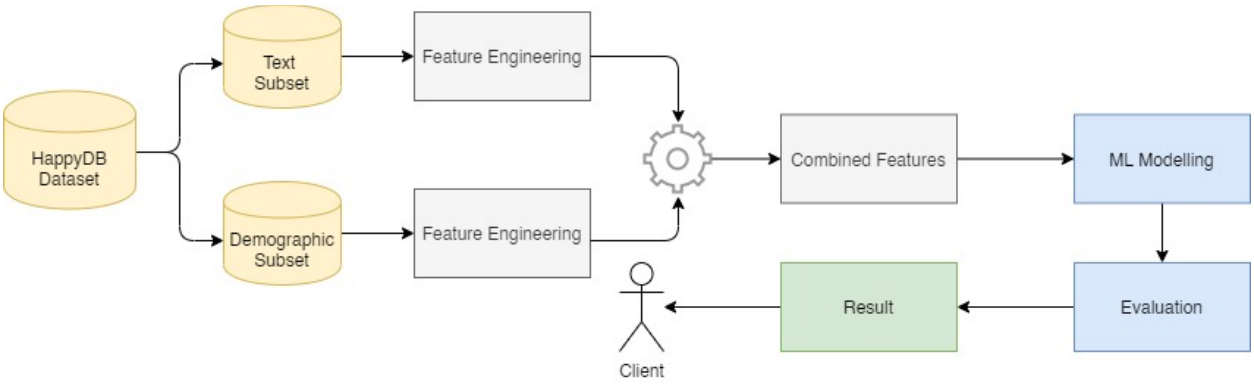


Figure 1. High-Level Architecture diagram for modeling work-life-balance during COVID19

Chapter 3 - Experiments / Proof of Concepts and Evaluations

Dataset: For our project, we have used the HappyDB Dataset. HappyDB is a corpus of 100,000 crowd-sourced happy moments. The goal of the corpus is to advance the state-of-the-art of understanding the causes of happiness that can be gleaned from text.

Dataset source: Dataset is referenced from the research paper published in Cornell University - '*HappyDB: A Corpus of 100,000 Crowdsourced Happy Moments'*, submitted on 25 Jan 2018, authored by Akari Asai, Sara Evensen, Behzad Golshan, Alon Halevy, Vivian Li, Andrei Lopatenko, Daniela Stepanov, Yoshihiko Suhara, Wang-Chiew Tan, Yinzhan Xu.

Dataset Description: Simply stated, HappyDB is a collection of happy moments described by individuals experiencing those moments. The following are some examples:

1. *I went on a successful date with someone I felt sympathy and connection with.*
2. *I was happy when my son got 90% marks in his examination*
3. *I went to the gym this morning and did yoga.*
4. *I achieved my daily income goal.*

Collecting happy moments: The happy moments are crowd-sourced via Amazon's Mechanical Turk. The paper presented each worker with the following task:

What made you happy today? Reflect on the past 24 hours and recall three actual events that happened to you that made you happy. Write down your happy moment in a complete sentence.

In this task, the "past 24 hours" is what the paper calls it the *reflection period*. HappyDB also contains happy moments with reflection periods "past week" and "past month".

A look at the dataset - `cleaned_hm.csv` containing the reflection period.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100535 entries, 0 to 100534
Data columns (total 9 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   hmid                  100535 non-null  int64
 1   wid                   100535 non-null  int64
 2   reflection_period     100535 non-null  object
 3   original_hm           100535 non-null  object
 4   cleaned_hm            100535 non-null  object
 5   modified              100535 non-null  bool
 6   num_sentence          100535 non-null  int64
 7   ground_truth_category  14125 non-null  object
 8   predicted_category    100535 non-null  object
dtypes: bool(1), int64(3), object(5)
memory usage: 6.2+ MB
```

Along with each happy moment, the dataset also consists of the demographic information of the worker who provided the moment.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10844 entries, 0 to 10843
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   wid         10844 non-null  int64
 1   age         10809 non-null  object
 2   country     10771 non-null  object
 3   gender      10812 non-null  object
 4   marital     10787 non-null  object
 5   parenthood  10813 non-null  object
dtypes: int64(1), object(5)
memory usage: 508.4+ KB
```

Data Preprocessing steps taken:

As our datasets contained both numerical and categorical features, we applied different data preprocessing techniques.

1. **Text data**

   For our text data, we performed the following data preprocessing steps:

   - Removal of Noise (like punctuations, period etc.) and less meaningful words to get a better analysis
   - Removal of regex, stop words and converting all the text into lower case
   - Dropping the instances with null values
   - Used Countvectorizer() to convert into a sparse matrix
   - Using TF IDF() to convert the input into CSR Matrix
   - Stemming and Lemmatization

- Tokenization of sentences using Spacy

2. **Demographic data**

We converted categorical data to numerical values for 6 features namely :- reflection_period, marital_status, gender, predicted_category, parenthood and country as follows:

- **Reflection_period**: We replaced the categories of reflection_period which were '3 months' and '24h' to 1 and 0 respectively .
- **Marital**: We replaced marital category values like married and unmarried ones with binary values of 1 and 0 respectively. We dropped rows with missing marital values too.
- **Gender**: We replaced gender attribute values of 'f','m','o' to numerical values with 'f' as 1,'m' as 0,'o'as 0.
- **Predicted_category**: 'For this feature, we converted the categories of 'Exercise', 'Enjoy_the_moment' etc. to the values Exercise' - 1,'Enjoy_the_moment' -2,'Achievement'-3,'Nature'-4,'Bonding'-5, 'Affection'-6,'leisure'-7.
- **Parenthood**: For this attribute, we converted the values 'y' and 'n' to binary values of 1 and 0, respectively.
- **Country**: For this project, we confined ourselves to the USA. Therefore, the value 'USA' was converted to 1 and all other countries like IND,VEN,CAN,GBR, were converted to 0.

Evaluation methodology followed

For our project, we used 3 folds cross validation.

Algorithm Parameter Grid:

- Random Forest:

```
forest = RandomForestClassifier(bootstrap=False, ccp_alpha=0.0,
class_weight=None,
criterion='gini', max_depth=None, max_features='auto',
max_leaf_nodes=None, max_samples=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, n_estimators=600,
n_jobs=None, oob_score=False, random_state=None,
verbose=0, warm_start=False)
```

- Decision Trees:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None,
criterion='gini',
max_depth=None, max_features=None, max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
min_samples_leaf=1, min_samples_split=2,
min_weight_fraction_leaf=0.0, presort='deprecated',
random_state=None, splitter='best')
```

Test size and train size

Test size - 20% and Train size - 80%

Confusion Matrix

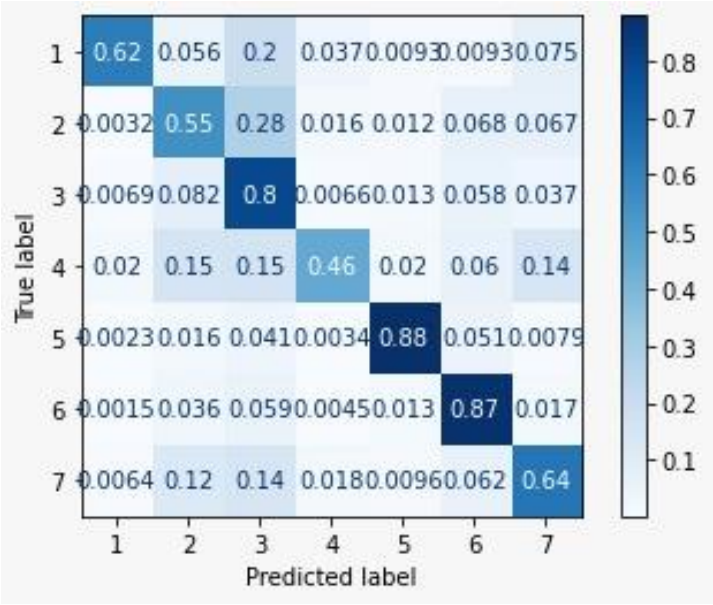


Figure 2. Confusion Matrix for Decision Trees

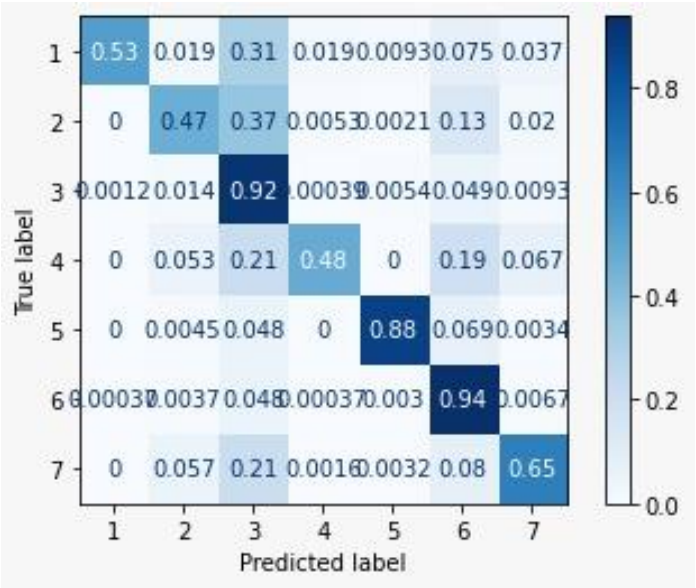


Figure 3. Confusion Matrix for Random Forests

Data Analysis

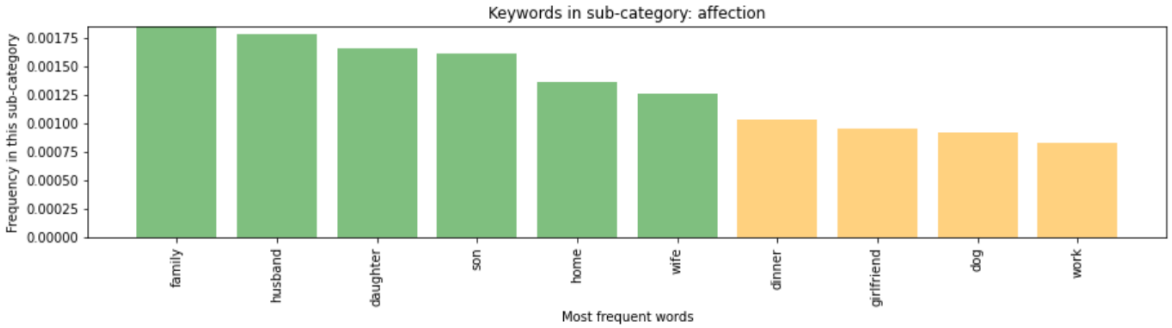


Figure 4. Plot showing the frequency of the most used words in the sub-category affection

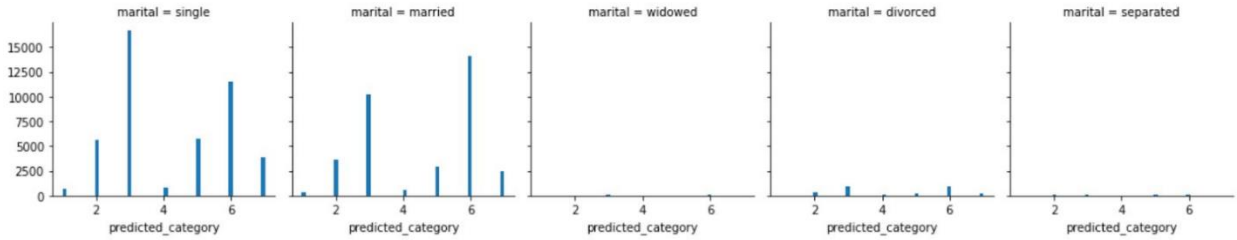


Figure 5. Bivariate analysis of age-ranges and predicted_category

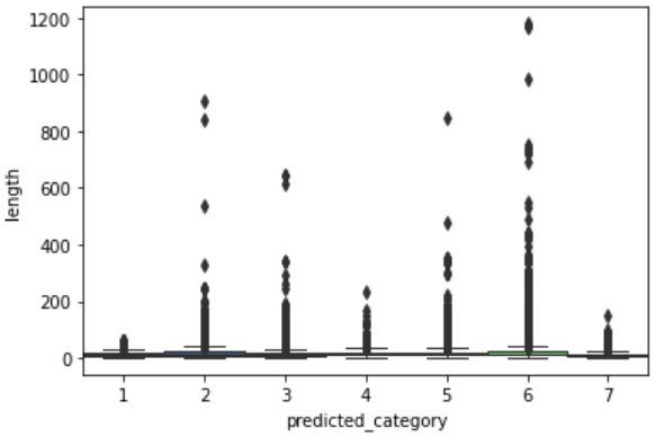


Figure 6. Boxplot for the length of a sentence with predicted_category

Metadata Analysis

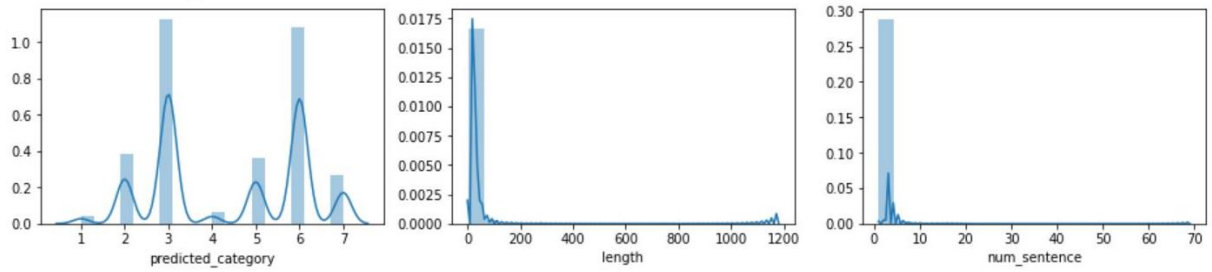


Figure 7. Distribution of predicted_categories, length and num_sentence

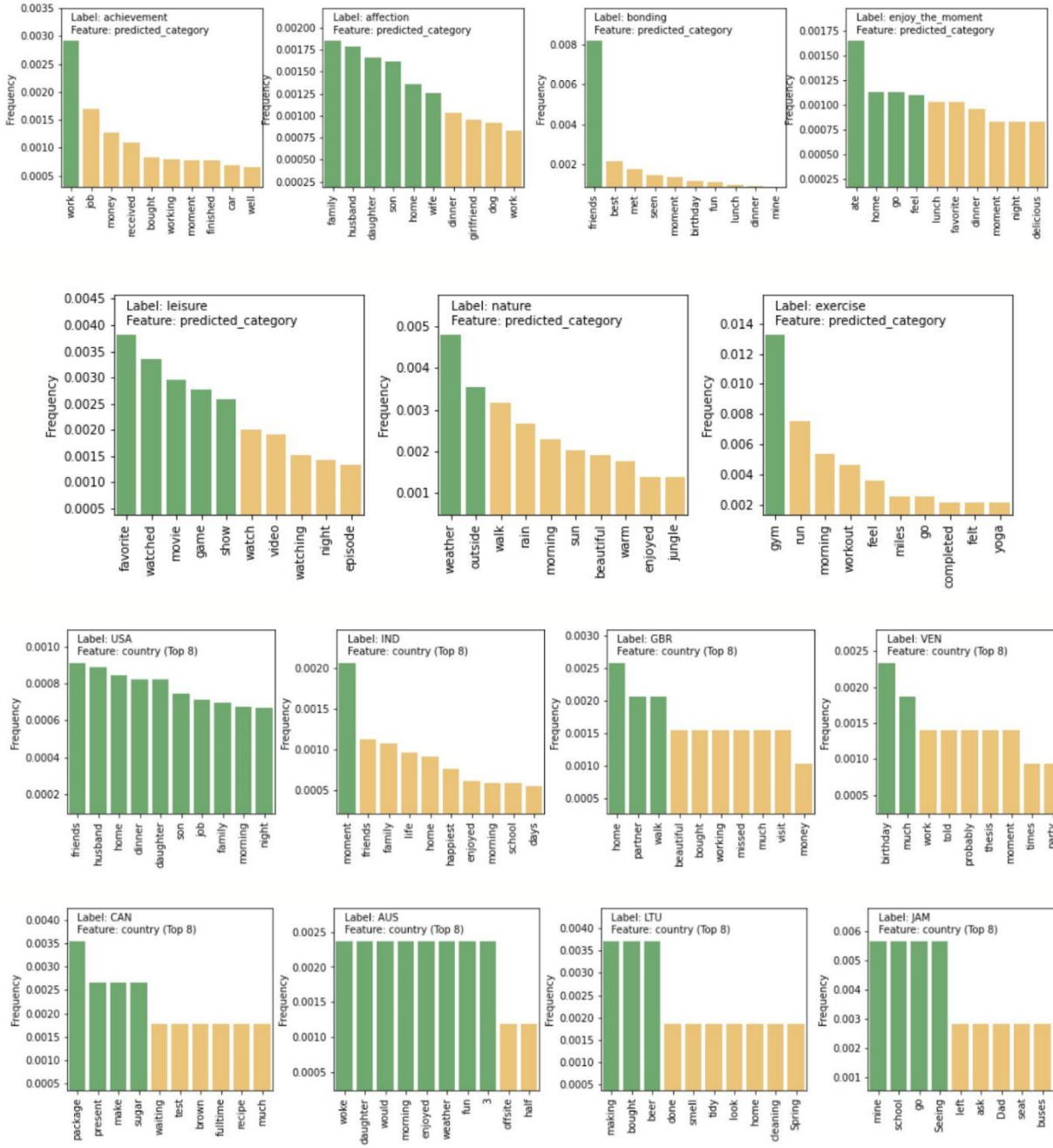
Figure 8. Word cloud before Text preprocessing

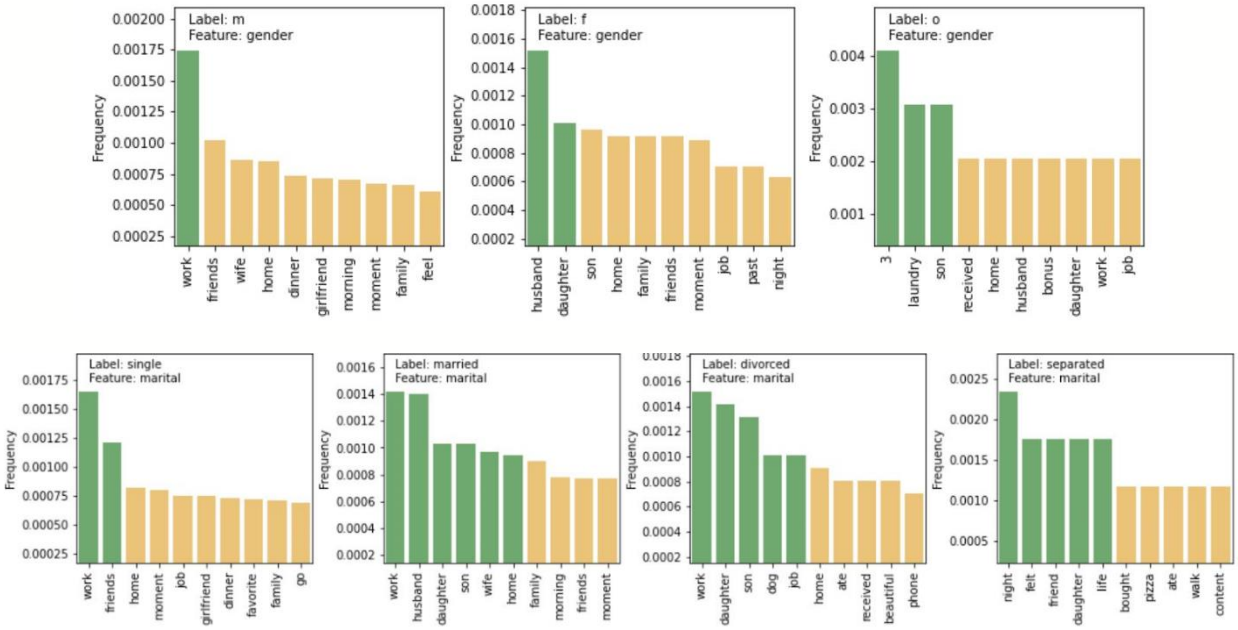
Figure 9. Word cloud after Text preprocessing

Keywords analysis for all labels of multiple features

The plots of this section denote the most meaningful words with the highest frequency.

The features selected are predicted_category, country, gender, marital, parenthood





Analysis of Results

We could successfully implement the model combining the features across Text and Demographics. The combined features were fed to three machine learning algorithms, namely Support Vector Machine, Decision Trees and Random Forests. Random Forests performed to be the best by improvising on bagging, decorrelating the weak decision trees on a random subset of features and handled the collinearity in features implicitly. The metrics across precision, recall, F1-score and accuracy for RF model are as follows:

	precision	**recall**	**f1-score**	**support**
exercise	0.93	0.53	0.68	107
enjoy_the_moment	0.55	0.55	0.55	943
achievement	0.78	0.80	0.79	2594
nature	0.53	0.46	0.49	150
bonding	0.90	0.88	0.89	881
affection	0.88	0.87	0.88	2696
leisure	0.63	0.64	0.64	628
accuracy		0.83	7999	
macro avg	0.87	0.70	0.76	7999
weighted avg	0.84	0.83	0.82	7999

Chapter 4 – Discussion & Conclusion

Things that worked

- We were able to successfully convert the categorical data into numerical one in our demographic dataset. Doing so, we were able to train our algorithms well.
- Random forest algorithm turned out to be accurate.
- Using a grid search estimator was a good decision as we were able to get the best fit.
- Using TFIDF was a good decision to analyze the text.

Difficulties faced/Things that did not work well

- The Support Vector Machines algorithm took time to execute and train.
- Our Source dataset was a big one - we faced system crashes multiple times as we tried to work with a merged dataset.
- Presence of noise in our text data - punctuations, period and irrelevant data.
- Deciding feature importance, hyperparameter tuning for SVM.

Conclusions

- We observed that people associated affection with family the most and with work the least.
- Associated work with achievement the most.
- Associated family with affection the most and work the least
- Associated friends with bonding the most
- They enjoyed the moment - when they ate
- Associated leisure by doing their favorite activity
- Associated nature with weather the most.
- People associated the gym with exercise and the least with yoga .
- People in the following demographic location:-
- In US, people are associated with friends
- In IND, people are happy with being in the moment
- In GBR, people are happy being at home
- In Venezuela people are happy on their birthdays.
- In Canada, people are happy with package/presents.
- In Australia, people are happy when they are awake.
- In LTU, people are happy when they are making something.
- In Jamaica, people are happy, when they associate something with themselves.
- Gender wise, males are happy at work.
- Single and married folks are happy at work. Widowed people are happy with their boyfriends.

Chapter 5 – Project Plan / Task Distribution

Task	Contribution
Data Preprocessing	Gayathri
Data Visualization	Gayathri, Osama
Feature Engineering	Ausaf
Random Forest	Osama
Decision Trees	Ausaf
Results and Conclusion	Osama, Ausaf, Gayathri
Presentation	Osama, Gayathri
Report	Osama, Ausaf, Gayathri

We ensured to collaborate effectively as a cohesive team, dividing project into subtasks with uniform contribution and timely deliverables.

References:

[1] *HappyDB: A Corpus of 100,000 Crowdsourced Happy Moments*, Akari Asai∗, Sara Evenseny, Behzad Golshanz, Alon Halevyz, Vivian Liz, Andrei Lopatenkoz, Daniela Stepanovz, Yoshihiko Suharaz, Wang-Chiew Tanz, Yinzhan Xuy, arXiv:1801.07746v2 [cs.CL] 25 Jan 2018
[2] https://megagon.ai/projects/happydb-a-happiness-database-of-100000-happy-moments/
[3] https://www.kaggle.com/ydalat/happydb-what-100-000-happy-moments-are-telling-us
[4] https://www.nature.com/articles/d41586-020-01059-4
[5] https://machinelearningmastery.com/clean-text-machine-learning-python/