Prediction of Life Satisfaction and Happiness

Final Report of STAT 841 (Statistical Learning: Classification)

Reza Valiollahi Mehrizi Bingfan Liu

Abstract

This report summarized the prediction results and machine learning algorithms used in STAT 841 Kaggle competition for predicting the life satisfaction among European people. In this competition, we employed, some well-known classification models in the literature such as K-Nearest Neighbors, Naive Bayes, Random Forest, Logistic Regression, Support Vector Machine, Gradient Boosting and Neural Networks. Finally, a subset of these models along with the important features chosen by Random Forest, Logistic Regression and Gradient Boosting are used in stacking and voting method to enhance the prediction accuracy.

1 Introduction

This report is a survey study about the life satisfaction level of the people living in the European countries. The goal is to predict whether the respondents are very satisfied with their overall lives based on their survey responses. The data collected in this Kaggle competition (https://www.kaggle.com/c/eurosat-w2020/data) contains 274 random variables including the response variable `satisfied`. There are 39,325 observations from which 30,080 of them are used as training data and the rest are used as the test data.

The summary of the report is as follows. Section 2 describes the pre-processing progress of the data. Exploratory analysis of the data is given in Section 3. Classification methods used in our analysis as well as their AUC scores are explained in Section 4. Finally, the report is concluded in Section 5.

2 Data Pre-Processing

As of the quote "Data scientists spend 80% of their time cleaning and manipulating data and only 20% of their time actually analyzing it", in the first step, we paid close attention to pre-process the data. Under many considerations, we manually removed some variables which seem apparently irrelevant to the response variable, `satisfied`. For instance, we omitted explanatory variables `v86-v97` and `v259-v270` from consideration. These are the variables of "the gender of the tenth person in the house" or "year of birth of ninth person in the house", etc. After performing this elimination, there are 125 variables out of 272 original ones remained for the further analysis.

Among the remaining explanatory variables, we split them into three categories: *Continuous Variables* (eg. `age`), *Ordinal Variables* (eg. `education`), and *Categorical Variables* (eg. `country`). This step is of importance for missing values imputation and one-hot encoding, which will be presented in the following of this section.

2.1 Imputation of Missing Values

The data contains a large number of missing values in the format of ".a", ".b", ".c", ".d" or "nan". This shows that caution must be taken during the imputation. We performed three types of imputations based on the three categories defined above. These three types are:

(i) Replace by a new value. Missing values for some categorical variables, such as `v70-employment`, `v160-marriage`, may have special reasons for why they are missed. For example, students will tend to answer ".a – not applicable" for the question `employment`. So we must regard this kind of answers as a new value of their categories.

(ii) Replace by an existing value. Missing values for some explanatory variables are suitable to be replaced by one of their existing values. For example, people who answered ".d – don't know" for the question `v177-interest in politics` can be replace by the answer "3", which is neutral. Therefore, we replaced the missing values with the existing levels.

(iii) Replace by mode or median. For the rest of the variables, we applied *mode imputation* for those that are categorical and ordinal. And we applied *median imputation* for those that are continuous.

Table 1 shows feature examples and the imputation methods applied to the data.

Feature	Method	Explanation
`v70`	(i)	Replace ".a" (not applicable) by a new category called "4 (other).
`v83`	(ii)	Replace ".b" (don't know) by "3" (Neither agree nor disagree).
`v84`	(ii)	Replace ".b" (don't know) by "3" (Neither agree nor disagree).
`v56`	(iii)	Replace ".a", ".b", ".c" by mode of the data
`v250`	(iii)	Replace missing values by median

Table 1: Various ways used for imputation of missing values

2.2 Feature Generation

Except for the methods mentioned above, we also decided to generate some new features using the existing variables. For this purpose, we introduced three new features `sad, happy` and `year_in_country`. This is because we suspected that the aggregated level of some features may be a stronger representative for respondents' total satisfaction level. These new features are defined as

$$\texttt{sad} = \frac{\texttt{v79}(feel\ depressed) + \texttt{v81}(feel\ lonely) + \texttt{v82}(feel\ sad) + \texttt{v222}(restless\ sleep)}{4},$$

$$\texttt{happy} = \frac{\texttt{v74}(enjoyed\ life) + \texttt{v80}(feel\ meaningful) + \texttt{v253}(feel\ happy)}{3},$$

$$\texttt{year-in-country} = \texttt{v153}(year\ came\ to\ country) - \texttt{v258}(year\ of\ birth).$$

2.3 One-Hot Encoding and Min-Max Scaling

In order to draw precise predictions using categorical or ordinal variables such as `country, employment relation`, we preformed one-hot encoding to make each different value of them an individual category. Moreover, some classification methods require the data to be scaled and standardized. To this end,

we applied min-max scaling, which is carried out via the equation $\frac{X-X_{min}}{X_{max}-X_{min}}$, on the features. Performing this process gave us 192 features.

3 Data Exploration

We began our analysis with some exploration of the data. To explore the distributions of the normalized features, we checked the histograms of a subset of features. Some examples of such histograms are displayed in Figure 1. In addition, correlation plots can give us a clear vision about the strength of linear correlation of each feature with the response `satisfied`. Bear in mind that the features may have non-liner relationships with the response variable, without loss of generality, we can start with suspecting the linear relationships between features first. Figure 2 displays the heating plot of correlations for a subset of features. We found that almost all of the variables had balanced skewness with some weak to moderate linear correlation with `satisfied`.

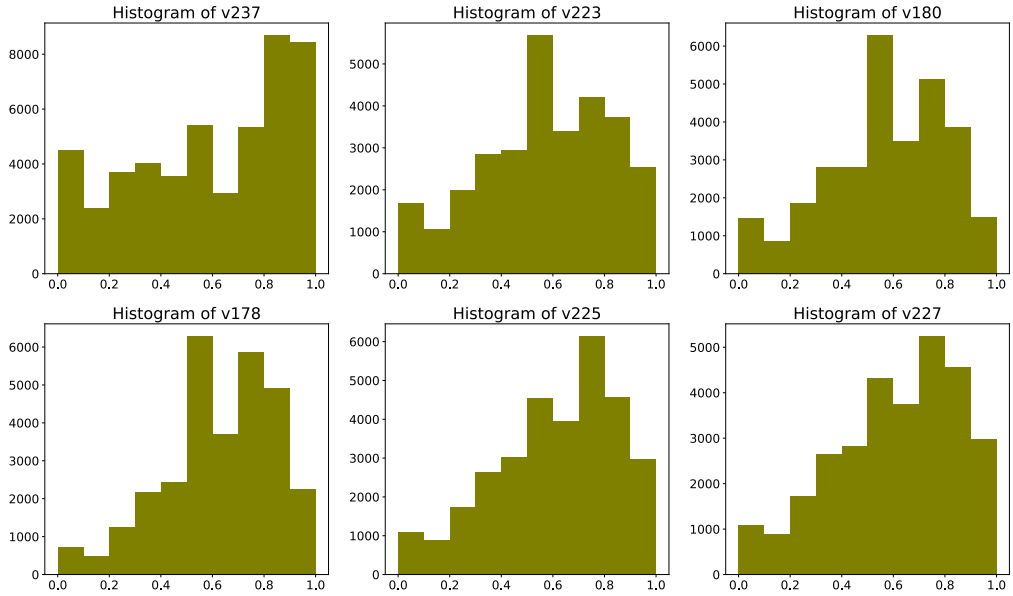

Figure 1: Histogram of some features in the data

4 Methodologies

In this section, we first fitted the models mentioned below on all of the 192 features after the data pre-processing. We also briefly explained these methods along with their advantages or disadvantages in this section. And then, we selected two sets of the important features using the models that performed the feature selection. Then, the stacking and voting models were applied for both of the two sets of the features.

The method used for parameter tuning was done by grid searches combined with 10-fold cross-validations. The best parameter values of the models as well as their corresponding AUC scores were presented in Table 2.

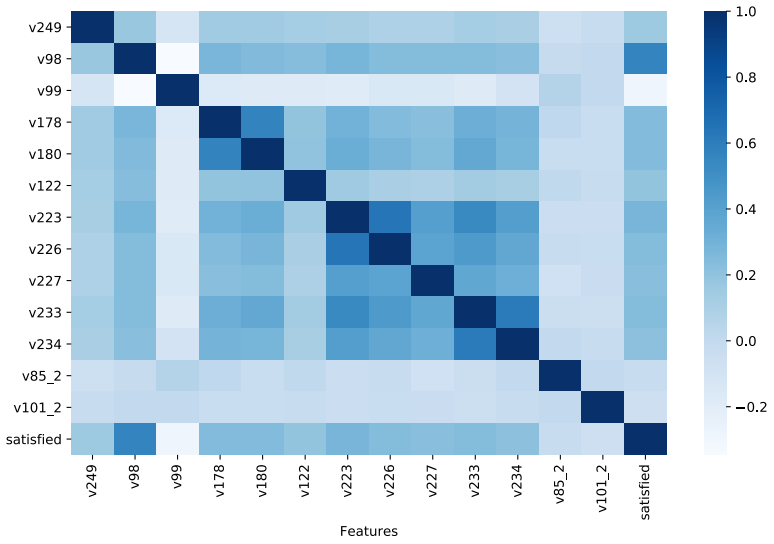

Figure 2: Heat map of correlation between some features and response variable

4.1 Basic Models

(i) **K-Nearest Neighbors (KNN):** KNN is a non-parametric method for classification. The model finds the closest K neighbors (the tuning parameter) for the target observation, then predicts the label of that observation based on the majority vote. The motivation behind KNN simply assumes that objects that are closed to each others are similar. This has made the technique very efficient regrading the computing speed.

(ii) **Naive Bayes (NB):** This is a simple technique for classification which naively assumes the value of a particular feature is independent of the value of any other feature, given the class label. This method works up to some extent, but usually the performance will not be the best compared to the other methods here.

(iii) **Logistic Regression (LR):** This method is highly interpretable and performs well in terms of prediction accuracy when the labels are linearly separable. Important features can also be selected using this approach by using the ℓ_1 penalization on it. We determined the important features by selecting the larger absolute values of the coefficients. For tuning the parameters, options for regularization (l1 and elastic net), alone with their regularization strength parameters are tuned.

(iv) **Random Forest (RF):** This model is a bootstrap aggregation model that trains a number of decision trees and take the average over them for the final prediction.This provides a relative reliable result by avoiding instability from fitting a single tree and usually generate a competitive prediction accuracy. However, the interpretations of the model become difficult, due to the nature of bootstrap sampling procedure on the features. By using its out-of-bag(OOB) samples, we fixed the number of the trees to use in the forest. Seen from the Figure 3, we decided to use 175 trees in the forest which we thought can provide a stable result and low error rates. The cost complexity parameter is used as the tuning parameter for restricting the tree size.

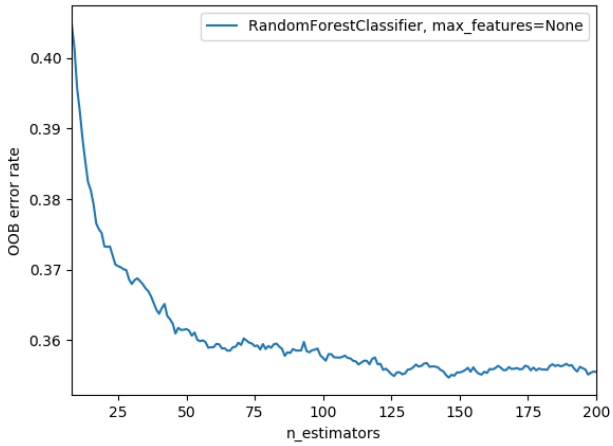

Figure 3: Selection of n-estimator

(v) **Support Vector Machine (SVM):** SVM draws hyper-plane and maximize the margin to the hyper-plane in order to perform classifications. It can handle the feature space where samples are either linearly separable or non-linearly separable. However, training a SVM model takes a long time which also making it hard for tuning the parameters. What's more, it is difficult to interpret the model. We tried both of the linear version and kernel version of the SVM, and performed parameter tuning using grid search and cross-validation.

(vi) **Gradient Boosting (GB):** This is an another tree-based algorithm that fits single decision tree at each iteration. Instead of averaging over all the trees, GB tries to find the best linear combination of fitted trees to explain the training data. As a result of this optimization, the GB model trains much slower but might generate competitive results. However, it is also known to possibly overfit the training data.

(vii) **Neural Network (NN):** Neural Network works as a universal approximator that sometimes predicts surprising results. But the method works like a black box and the structures of networks are usually determined by empirical expert experience. Based on some experiments, we decided to use a neural networks with 4 layers. And the activation functions along with their quantities used in layer 1 to 4 are: (relu, 64), (relu, 32), (relu, 16) and (sigmoid, 1).

Models	Best Parameters	AUC Scores
KNN	`n_neighbors= 49`	0.8514
Naive Bayes	`- - - -`	0.8506
Logistic Regression	`C= 0.0015, penalty= "l1"`	0.8650
Random Forest	`ccp_alpha=0.015 , criterion= "gini"` `max-features= "sqrt"`	0.8782
SVM	`C=0.1, kernel="linear"`	0.8645
Gradient Boosting	`ccp_alpha=0.01, learning_rate= 0.05`	0.8767
Neural Network	(relu, 64), (relu, 32), (relu, 16) and (sigmoid, 1)	0.8655

Table 2: Different approaches used for classification in this report.

4.2 Feature Selection

Feature selection plays a key role in data analysis, particularly in high dimensional setting. The current data has about 192 features after one-hot encoding. Our interest was to select those features which have the most influence on the response variable. Among the methods considered in our analysis, we decided to use l1-Logistic Regression, Random Forest, and Gradient boosting for feature selection.

After the best tuned models L1-logistic regression returned 34 important feature while Random Forest and Gradient Boosting gave 40 and 38 important features, respectively. We also visualized these selected features in Figure 4. Note that for sake of clarity, we plotted only the first 15 selected features by each method.

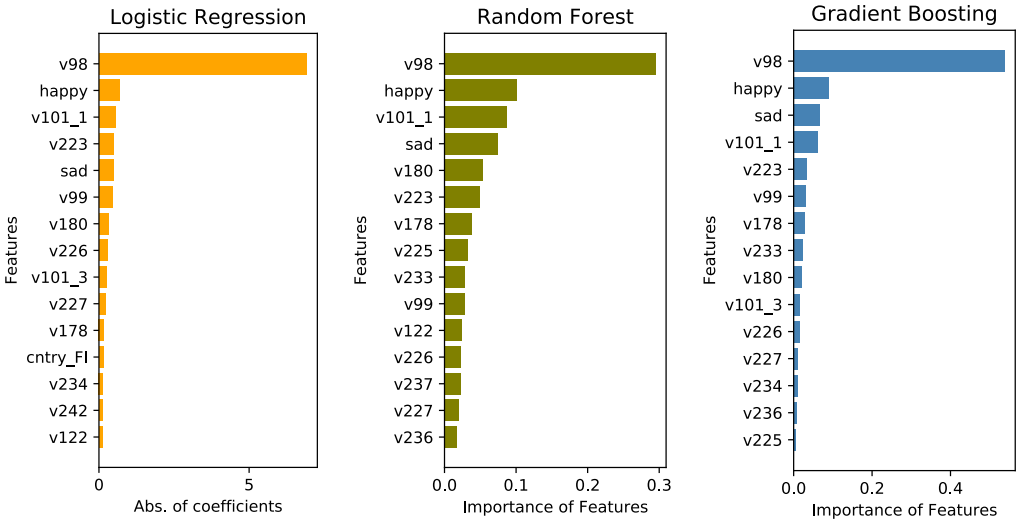

Figure 4: Important features selected by Logistic Regression, Random Forest and Gradient Boosting

We eventually decided two possible sets of features to use:

- **Set 1:** Using the joint of three sets of the features selected by models:l1-Logistic Regression, Random Forest and Gradient boosting.
- **Set 2:** Using the unique union of three sets of the features of the three models.

4.3 Stacking and Voting

Stacking and voting techniques combine the aforementioned methods to enhance the power of prediction. Stacking combines multiple classifiers by a logistic regression, i.e.

$$\text{logit}(\mathbf{p}) = \sum_{i=1}^{m} w_i \, \mathbf{p}_i,$$

where $\mathbf{p}_i$ is the predicted probabilities of different classifiers and w are weights to be determined by logistic regression. On the other hand, voting returns the class label associated with the largest

weighted sum of the predicted probabilities from each individual classifiers, that is

$$\mathbf{p} = \arg\max \sum_{i=1}^{m} w_i \, \mathbf{p}_i,$$

where w_i is the weight of the model i which is replaced by its AUC score.

After some experiments, we decided to use four classifiers in stacking and voting methods. These classifiers were: Logistic Regression, Random Forest, Support Vector Machine and Gradient Boosting. Table 3 summarized performances of all the models we tried for the two sets of features.

	Number of Features	Cross-Validated AUC Scores								
		stacking	Voting	KNN	NB	LR	RF	SVM	GB	NN
Set 1	21	0.8825	0.8812	0.8705	0.8675	0.8783	0.8811	0.8769	0.8804	0.8785
Set 2	55	0.8797	0.8785	0.8661	0.8639	0.8728	0.8774	0.8725	0.8765	0.8736

Table 3: AUC scores derived from various models including stacking and voting for the two sets

5 Conclusion

In this study, we used various of techniques in data cleaning, feature engineering and data exploration. We also performed predictions using seven well known different machine learning models and performed feature selections. Eventually, we combined four of the classifiers, Logistic Regression, Random Forest, Support Vector Machine and Gradient Boosting, using the stacking and voting methods with only 21 selected features.

Table 3 revealed that the stacking using only 21 features from the joint set of features produces the highest AUC score. What's more, using either the Set 1 or Set 2 can improve the AUC score of the all the seven basic models, use the joint set of the features (Set 1) can improve the AUC scores most.

What's more, some examples of the important features from set 1 are `happy`, `income`, `education`, `sad`, `general_health_state`, `being_trustworthy`, `democracy_in_country`, `feeling_safety`, `meeting_friends`, `daily_ work_hours` and etc.

Based on the feature importance and results from logistic regression, we can conclude that people with higher education, more income, and less daily work hours tend to be more satisfied with their life. On the other hand, some social factors, such as general health state, democracy state in the country and communication with friends, have important influence on the level of satisfaction. In other words, people who communicate less, or live in the country with untrustworthy legal system are more likely to be unsatisfied.