CSCI3320 Programming project
Prof. John C.S. Lui
Due date: May 2rd

Group Member:
Tu Shuo 1155092206
Li Zhirong 1155092195

2 Raw Data Preprocessing

2.2 Data Discretization and Analysis

For optional task, firstly drop useID since value in this attribute are all different and it has no relation with whether a person is happy or not. Then perform onehotencode on Income, HouseholdStatus, EducationLevel, Party, since these attributes contain several values and in real world, they have high chance to affect human's happiness.

2.3 Missing Data Filling

How to group the users:

1. drop all missing data and use rest data.
2. Then for every attribute that contains missing data. Use this attribute as target and the rest attributes as attributes to build a decision tree of height 2 by using DecisionTreeClassifier in sklearn.
3. Then find two or three attributes that are important in splitting the sample to predict the values of that target attribute by using function feature_importances_ in DecisionTreeClassifier.
4. Use those important attributes to group original sample(contain missing data).
5. In each group, replace missing data with the mean/median/most_frequency value of that group.
6. Since there are hundreds of attributes and nearly 100 attributes have missing data, we build a decision tree for all attributes that contains missing data inorder to infer the missing data. It costs some time actually and performing filling missing data by this method costs nearly 2 minutes.
7. There are cases that all values of a attribute in a group are NaN, so after this operation, only some of missing data is filled. Therefore, we try to perform this operation for several times. So to reach a higher accuracy of training data, we needs about 8 minutes to get the result.
8. Finally, there are still some missing data in the samples. So we will apply the basic strategy in the first task, which is replacing the missing data with mean/median/most frequency of that attributes.

3 Classification

3.1 Train Classifiers in Scikit-Learn

3.1.1 Logistic Regression

Refer to the Table1.

	accuracy	time consumed(second)
Scikit Learn	0.690	0.08
My Own Algorithm	0.688	20.18

Table 1: Logistic Regression

3.1.2 Nave Bayes

Refer to the Table2.

	accuracy	time consumed(second)
Scikit Learn	0.68	9.5e-07
My Own Algorithm	0.67	7.1e-06

Table 2: Naive Bayes

We choose GaussianNB, because it is super simple and it assumes all variable are gaussian distributed, which roughly match the fact.

3.1.3 SVM

We use linear kernel because other kernel cost lost of time while the result is not apparently better than linear model. Linear support vector machine could fight against overfitting.

3.2 Write A Report

(1)

Naive Bayes: super simple, a Naive Bayes classifier will converge faster than discriminative models like logistic regression. And even if the NB assumption doesnt hold, a NB classifier still often does a great job in practice.
Logistic Regression: Lots of ways to regularize your model, and you dont worry about your features being correlated.
Random Forest: Easy to interpret and explain. They easily handle feature interactions and theyre non-parametric, so you dont worry about outliers or whether the data is linearly separable.
SVM: overcome against overfitting, and with an appropriate kernel they can work well even if youre data isnt linearly separable.

(2)

1. Naive Bayes model is used when the training data is more likely to be independent since its algorithms based on applying Bayes theorem with the naive assumption of independence between every pair of features. A classical use case for Naive Bayes is document classification, which is to determine whether a given (text) document corresponds to one or more categories.

2. Logistic Regression model is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). In its implementation, Logistic Regression is used to ascertain the probability of an event. And this event is captured in binary format, i.e. 0 or 1. Here is an example: if we want to ascertain if a customer will buy my product or not. For this, I would run a Logistic Regression on the (relevant) data and my dependent variable would be a binary variable (1=Yes; 0=No).

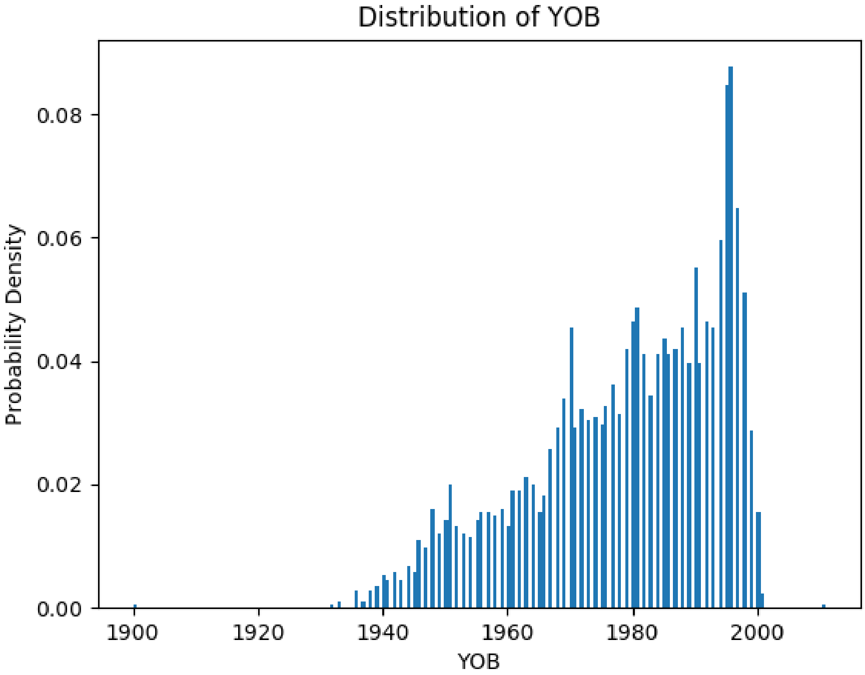
(3)

A model that would have a perfect score in training data but would fail to predict anything useful on yet-unseen data. This situation is called overfitting. So we separate the data set into training data and test data, and use model to fit the training then use test data to score. Seperating data is helpful to prevent overfitting since the validation is not training data itself but another data set.

4 Visualization

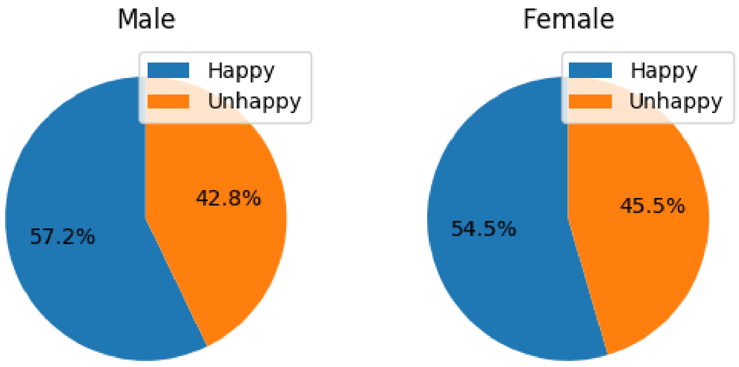
4.1 Basic Visualization Methods

4.1.1 Histogram of YOB



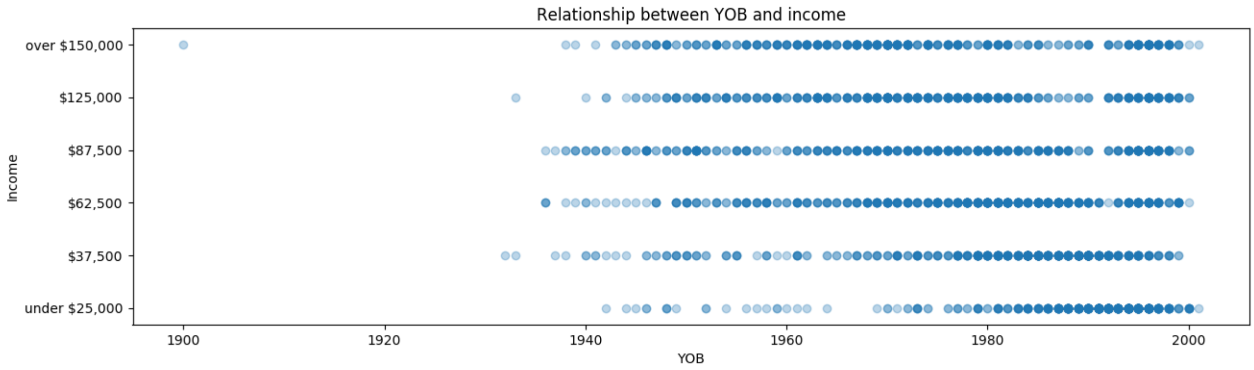
Description:
The sample inputs are most from people born between 1970 and 1990 and the number of people increases when the YOB getting larger.

4.1.2 Pie chart for the fraction of happy men/women



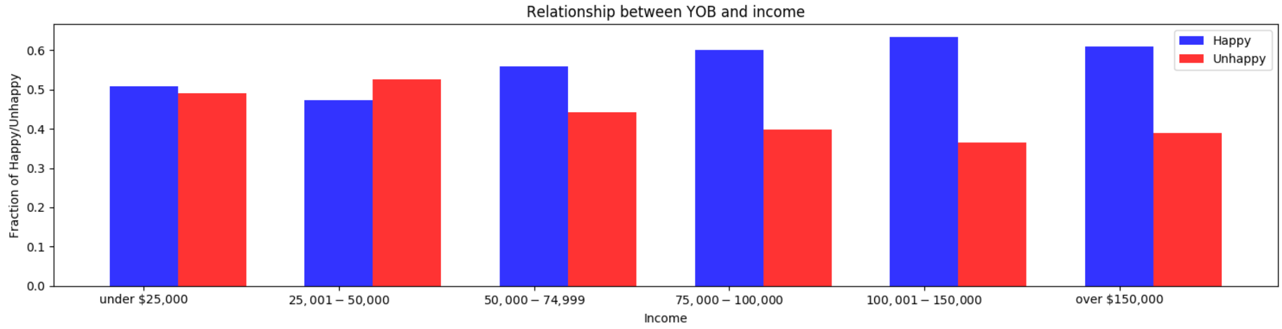
Description:
From the statics given, we can learn that number of happy people are more than unhappy people for both male and female and male are more likely to be happy than female.

4.1.3 Scatter plot of YOB and income



Description:
From the figure we can know that, people with higher income are mostly older but it is not absolute since some younger people also have high income.

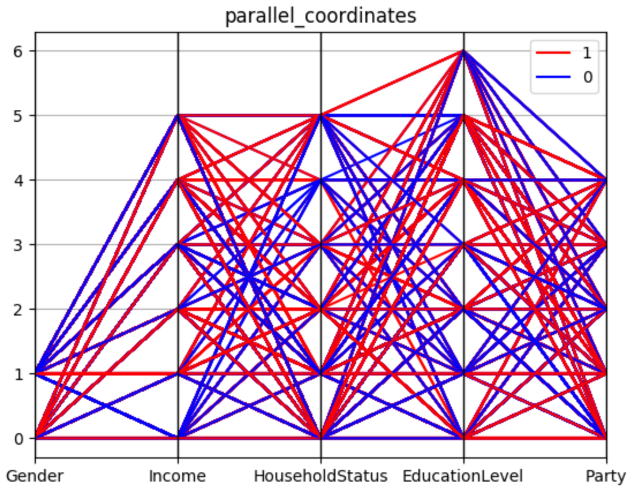
4.1.4 Bar chart of income and happiness



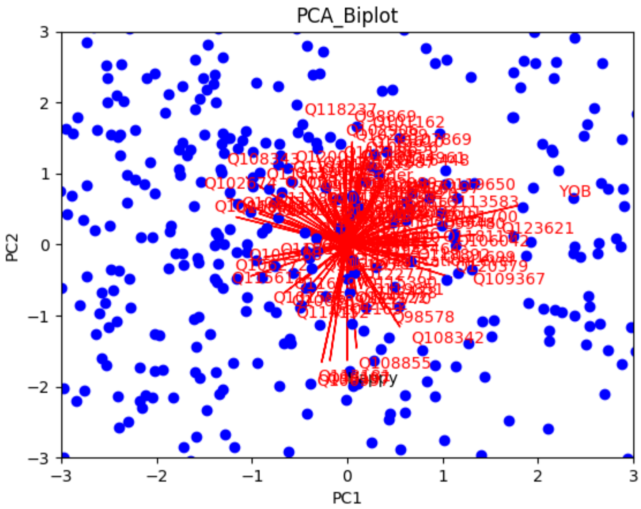
Description:
From the figure we get, people will be more happier with the increasing of their income in general.

4.2 Visualizing High-dimensional Data

4.2.1 Parallel Coordinates Plot



4.2.2 PCA and biplot

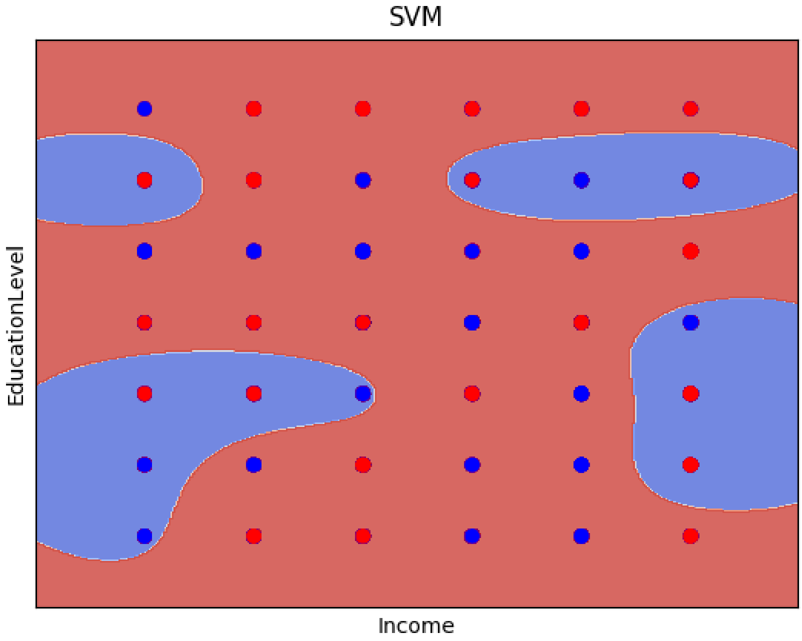


Q1:the vector means projection of the variables onto the two PCs.

Q2:the vector which is closed to happy is related to happiness. Because the angle of two vectors represents the correlation of the corresponded variables.

4.3 Visualizing Classification Result

4.3.1 Visualize SVM


SVM
EducationLevel
Income