In-class Kaggle Competition: Boy or Girl

CHUN-HUA, SHIH
112423004
a29424315@gmail.com

CHIA-HENG, KUO
112423066
Henryedu0904@gmail.com

TING-LIN, KUO
112423061
ks05196618@gmail.com

Introduction

The following describes our experimental design process. In this experiment, we will examine in detail how we solve three major problems in data processing, namely: missing values, imbalanced datasets, character selection, and extraction. We will focus on preprocessing techniques, model training techniques, and text-processing techniques to focus on the right data. In this report, we will not only show the methods we have chosen and the reasons behind them, but also explore the improvements we have implemented and how these improvements help us to predict gender more accurately.

Keywords

Random Forest; SMOTE; TF-IDF; Decision Tree; SVM

Methodology and Selection

1. Data preprocessing
   This dataset is used to classify individuals into "male" and "female" categories. This dataset contains nine distinct characteristics, which are as follows: 1. The star sign; 2. The mobile OS 3. stature, 4. mass, 5. Size of sleep; 6. IQ; 7. Facebook friend count; 8. YouTube viewing duration; and 9. Self-introduction. These nine traits will be used to determine their gender.

   In the data pre-processing part, we first deleted one feature: star_sign, because based on the critical research there is no obvious relationship between star sign and gender. Next, the following will show four important steps that we have performed pre-processing for the dataset.

   a. Missing value imputation
      When dealing with missing values, we consider the type of data, including discrete data and continuous data.

      For discrete data, such as phone_os, we use the mode to fill missing values. There are only 3 operating systems in this data set. Although it can be converted into numerical values using encoding, its nature still cannot be sorted so it’s the reason why we use mode.

      For continuous data, such as height, weight, IQ, etc., we considered two candidates: median or mean. After browsing the dataset, we finally chose to fill in the median to avoid too many extreme outliers affecting the entire data.

b. Detecting and Treating Outliers

To address extreme outliers and ensure model stability, we experimented with various outlier removal methods due to the dataset's excessive length. After careful consideration, we opted for manually setting upper and lower bounds. Any record outside this range is assigned a value of 0 in the code. Below are our methods we have tried in process.

- MinMax
  Initially, we explored the application of MinMax scaling as a means to normalize the data within the range of [0, 1]. This method was chosen with the intention of facilitating smoother operation of the prediction model. However, despite the scaling efforts, certain data points continued to exhibit values exceeding the constraints of the float32 datatype. Consequently, due to these persistent challenges, the MinMax scaling approach was deemed impractical and subsequently abandoned.

- Find the maximum value for each feature
  Another approach involved eliminating outliers by determining the maximum values for each feature. This targeted removal was conducted utilizing the 'drop' method within the Pandas module. Given that outliers often skew the distribution and adversely affect model performance, this method proved effective in enhancing the robustness of our dataset.

- QuantileTransformer()
  This method alludes to feature scaling or normalization techniques used in data preprocessing. Its primary aim is to reshape features so that they approximate a uniform or normal distribution. By doing so, it encourages the dispersion of the most common feature values, promoting a more even distribution. Furthermore, this transformation aids in mitigating the influence of outliers, particularly those that are marginal or extreme. Through spreading out the feature values, the effect of outliers is diminished, contributing to a more resilient preprocessing approach.

- Setting the upper and lower bound manually
  We adopted a manual approach to setting upper and lower bounds. Within this framework, we defined a range from 0 to 10000, beyond which any values were set to 0. By imposing explicit bounds, we ensure that the random forest (RF) model can smoothly analyze the dataset.

In conclusion, after experimenting with various outlier removal methods, manually setting upper and lower bounds proved to be the most effective approach. Despite initial attempts with MinMax scaling, which faced challenges with data exceeding datatype constraints, and other methods like identifying maximum values and using QuantileTransformer(), manually setting bounds from 0 to 10000 ensured straightforward outlier removal. This approach emerged as the preferred choice for outlier treatment, offering practicality and reliability in dealing with outlier challenges in our dataset.

c. NLP and Feature selection

This dataset contains textual information, this field may interfere with the training of the model and prevent the model from functioning properly, so we need to convert the text into vectors for model training. Since "self_intro" represents the personal introductions of each data entry, it focuses on the importance of words in the text and key information that can influence gender classification, rather than the contextual structure of sentences. Therefore, we choose TF-IDF for vector transformation.

First, we preprocess the text by converting it to lowercase, handling contractions, removing punctuation, tokenizing words, eliminating stopwords, and lemmatizing words. After preprocessing, we attempt to perform TF-IDF transformation, which extracts nearly 300 vocabulary. However, training on all these terms might lead to overfitting issues due to the curse of dimensionality.

To address this, we first count the occurrences of each word in the training set and only select words that appear at least three times for TF-IDF transformation. Then, we apply the chi-square test to the selected words, conducting supervised training with the target prediction category—gender field, and ultimately extract the top 20 features as the final input.

The following are the test results using only the original training set, divided into 80% training data and 20% validation data, after preprocessing with handling missing values, outliers, etc. Different numbers of word features were selected, and random forest was used as the training method to observe predictions made using three different random splits of the original training set. Accuracy was calculated as the basis for selecting the number of features. From the table below, it can be seen that the accuracy is highest when selecting 20 word features. Therefore, 20 features were chosen as the final dimensions for the training set’s word features.

Table 1 The accuracy of selecting different numbers of text features

Random_state	14	28	42
46 features (occurrences > 3)	0.9176	0.9059	0.9176
30 features (chi-square test)	0.8941	0.9176	0.9059
20 features (chi-square test)	0.9176	0.9411	0.9176
10 features (chi-square test)	0.8941	0.9294	0.9059

d. Dealing with Imbalanced data: SMOTE

Class imbalance can lead to biased models that perform poorly in predicting the minority class, as the model tends to be biased towards the majority class due to its higher prevalence in the dataset.

The number of men and women in the original data is 316:107, and the ratio is close to 3:1. In order to avoid data imbalance that prevents the model from learning data with few categories well, we use smote to deal with this problem.

The SMOTE is to mitigate the effects of class imbalance by oversampling the minority class. SMOTE works by generating synthetic samples in the feature space, interpolating between existing minority class samples. This helps in balancing the class distribution and provides the model with more representative samples of the minority class.

2. Training model
    a. Support vector machines (SVMs)

       Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

       SVC, NuSVC and LinearSVC are classes capable of performing binary and multi-class classification on a dataset. As other classifiers, SVC, NuSVC and LinearSVC take as input two arrays: an array of shape holding the training samples, and an array of class labels.

    b. Random Forest (RF)

       Support Vector Machines (SVMs) are versatile supervised learning methods used for classification, regression, and outlier detection. Among SVM implementations like SVC and Linear SVC, these classes handle binary and multi-class classification tasks. They operate by taking two arrays: one for training samples' features and another for class labels.

    c. KNN

       KNN, short for k-Nearest Neighbors, is a supervised learning algorithm based on instances and is commonly used for classification and regression problems. The basic idea of this algorithm is to find the k nearest neighbors (data points) to a new data point by comparing its distance to all data points in the training set, and then predict the label of the new data point based on the labels of these neighbors.

       To test the predictive ability of KNN, we divided the original training set into 80% training data and 20% validation data. After preprocessing, we used a total of 28 dimensions as input for KNN training. However, the accuracy of predictions was only 0.76. Furthermore, when we tried training on the entire dataset and testing on Kaggle, the accuracy was still only 0.65. This indicates that KNN is not a suitable model for this dataset.

    d. Decision Trees

       Decision Trees (DTs) are a versatile supervised learning technique used for regression and classification tasks. They work by learning simple decision rules from the data features to predict the value of a target variable.

       In our approach, we rely on the Scikit-Learn tree module and follow the methodology outlined in the official documentation. Firstly, we split the original male and female training sets into training and test sets using the train_test_split method from the sklearn_selection module. Then, we utilize methods from Scikit-Learn to build and train our decision tree model.

All these models are not only interpretable but also provide valuable insights into the decision-making process. We select Random Forest as our final method for several reasons below:

1) Ensemble Method: Random Forest is an ensemble method that combines multiple decision trees to make predictions. By aggregating the predictions of multiple trees,

a random forest tends to have higher accuracy and better generalization than a single decision tree.

2) Reduced Overfitting: A Decision Trees are prone to overfitting, especially when they are deep. Random Forest mitigates this issue by averaging the predictions of multiple trees, which helps reduce overfitting and improves the model's ability to generalize to new data.

Procedure

To predict the gender of a given respondent for an online questionnaire.
The training processes as following,

1. Data Collection: We use “Boy or Girl 2024 new “Dataset in Kaggle which consist of items from the questionnaire at the beginning of the course.
2. Data Preprocessing: Preprocess the data by handling missing values and outliers, encoding categorical variables, and scaling numerical features if necessary. For missing value, we choose mode to fill discrete data, and use median to fill continuous data. To handle outliers, we manually setting upper and lower bounds. Finally, we convert the text into vectors by TF-IDF.
3. Feature Selection: Select the features that are most relevant for predicting the gender of the respondent. This can be done using techniques like correlation analysis or feature importance. We first deleted the ‘star_sign’ field, and use chi-square to select the important terms in TF-IDF. Finally, we choose 28 features as inputs to training model.
4. Class imbalance: The SMOTE is to mitigate the effects of class imbalance by oversampling the minority class. SMOTE works by generating synthetic samples in the feature space, interpolating between existing minority class samples.
5. Split the dataset into training and validation set: The training set will be used to train the model, while the validation set will be used to evaluate its performance.
6. Model Selection and Training: Choose a machine learning model that is suitable for the task of gender prediction, we use k-Nearest Neighbors, support vector machines, random forest, and decision trees. Then train the selected model.
7. Model Evaluation: Evaluate the trained model on the validation data to assess its performance. Common evaluation metrics for classification tasks include accuracy, precision, recall, and F1 score. Finally, we choose random forest as our final method.

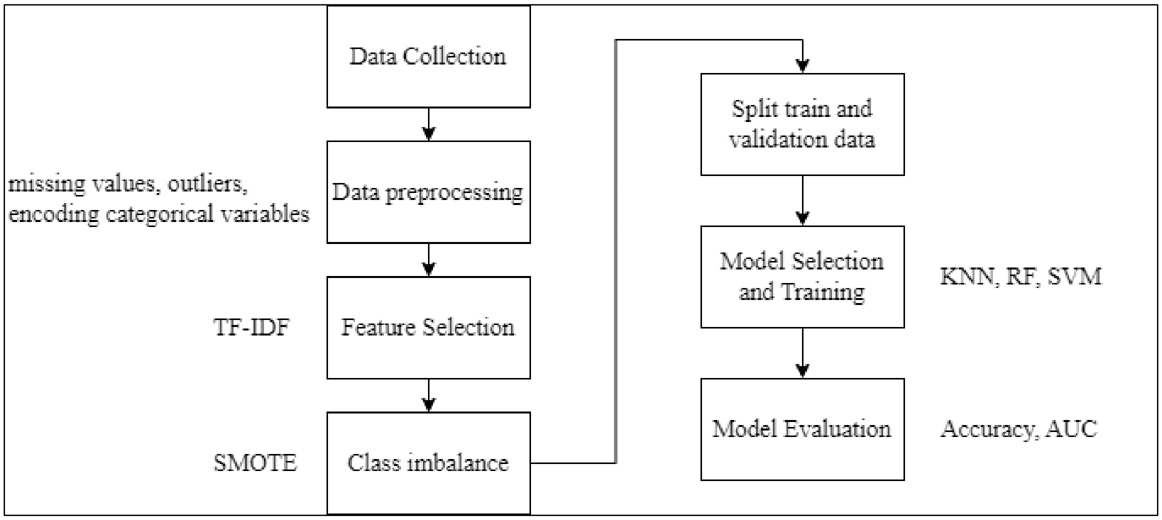

Figure 1 The flowchart of the training process

Discussion

Table 2 The result of different preprocessing and training methods in Kaggle competition

	Data Preprocessing			Model Prediction	Performance	
No.	Sampling Method	Semantic Analysis?	Outlier Method	Classifier	Private	Public
1	X	X	X	X	0.50505	0.50761
2	SMOTE	X	Maximum of 5 rows	SVM(RBF)	0.80808	0.82741
3	SMOTE	X	Median ± 1.5*IQR	SVM(RBF)	0.81313	0.87309
4	SMOTE	X	Median	SVM(RBF)	0.39393	0.34517
5	X	V	Median	KNN	0.59595	0.65482
6	SMOTE	X	Median+setting up/low bound	Random Forest	0.84343	0.89847
7	SMOTE	V	Median+setting up/low bound	Random Forest	0.83838	0.90355
8	SMOTE	V	Median+setting up/low bound	SVM(RBF)	0.55555	0.51776
9	X	V (Total 28 features)	Median+setting up/low bound	Random Forest	0.84343	0.87817
10	X	V (Total 54 features)	Median+setting up/low bound	Random Forest	0.83333	0.86802
11	SMOTE	V	Median+setting up/low bound	Random Forest	0.87373	0.91878

In our study, we tackled various data challenges such as imbalance, missing values, and outliers. Median imputation effectively handled missing values, while manual outlier handling ensured data consistency.

In text mining, TF-IDF and semantic analysis improved gender inference from self-introduction texts.

During model evaluation, the random forest model with SMOTE emerged as the most effective, demonstrating high accuracy and generalization. Conversely, linear SVM struggled with gender distinction, leading us to exclude it.

Overall, we address data challenges and leveraging techniques like SMOTE and semantic analysis enhanced predictive accuracy and informed our selection of the most suitable classification approach.

Group Division of Work

Name	Work description	Contribution
CHUN-HUA, SHIH 112423004	Building model (SMOTE, Random Forest) Document writing Devising empirical experiment	34%
CHIA-HENG, KUO 112423066	Building model (SVM, Decision Tree) Document writing Devising empirical experiment	33%
TING-LIN, KUO 112423061	Building model (TF-IDF, KNN) Document writing Devising empirical experiment	33%