STAT479 Project Report: Predict flight delays

Jingshan Huang
jhuang456@wisc.edu

Yuan Cao
cao234@wisc.edu

Runze You
ryou3@wisc.edu

Abstract

We use machine learning algorithms to make a decision whether or not a flight will be delayed. Not only international students, people who often travel by plane know that whether or not a flight is delayed is always uncertain. Hence we obtain the dataset from Kaggle, which has some features about delayed or not delayed flights, to help us establish models for predicting.

The supervised models we use on the training dataset including k-Nearest Neighbors algorithm, decision tree and random forest, combined with k-fold Cross Validation (k-Fold CV) to evaluate the predictive model and change the hyperparameters. Besides, in order to get a balanced train set, we apply resampling methods such as SMOTE or SMOTETomek and compare them.

In kNN, we use Euclidean Distance with $k = 10$, rectangular weighting method and 2-fold cross validation considered large enough dataset, then get the model accuracy which is 56.26% and the recall which is 0.69. Also, we try to split the train set in four parts to make it balanced and use four decision trees to do a "majority voting". With different hyperparameter in different trees, we get 0.695 recall and 62.45% accuracy. Last, by using random forest and resampling methods, we obtain 0.59 recall and 68.23% accuracy when just repeating the minority class and 0.28 recall and 77.04% accuracy when applying SMOTE.

1. Introduction

Nowadays, aircraft is one of the most important and popular methods of transportation in many countries. Its high speed makes it convenient for business and private travel, especially in long distance.

However, it is still not a perfect way. Although rarely hindered, it also has a common disadvantage of public transportation, delay. Not only for us international students, people who often travels by plane know that flight delays are always uncertain, but usually affects people's entire journey greatly.

According to the report of FLIGHTSTATS[1], from September 20th to October 20th, there are 171,582 delayed ights and 11,355 canceled ights in the United States (Figure: 1), 501,499 delayed ights and 33,704 canceled ights in the world. Statistic data from 46 major international airports around the world in September shows that the ight delay rate at London Gatwick Airport and Airport Schiphol is even more than 40 percent (Figure: 2). In addition, there are eighteen, i.e., 40 percent of international airports have an average delay time of more than 60 minutes. Particularly, Chengdu Shuangliu International Airport, San Francisco International Airport and George Bush Intercontinental Airport have an average delay of more than 70 minutes (Figure: 3).

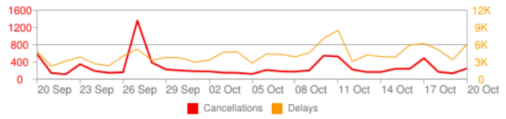

Figure 1. Last 30 days flight delays and cancellations of the United States

Airport Code	Airport City	On-time Ranking	Flights	Tracked	Comp. Factor	On-time (D14)	Delayed (15..	Avg. Delay
LGW	London, EN, GB	46	13,006	99.38%	96.63%	57.43%	42.57%	46.2
AMS	Amsterdam, NL	45	21,023	91.18%	98.96%	57.95%	42.05%	41.9

Figure 2. Flight delay rate

Airport Code	Airport City	On-time Ranking	Flights	Tracked	Comp. Factor	On-time (D14)	Delayed (15+ min.)	Avg. Delay
CTU	Chengdu, CN	14	14,866	99.72%	99.04%	86.35%	13.65%	76.5
SFO	San Francisco, CA, US	33	16,906	99.16%	95.65%	78.83%	21.17%	74.2
IAH	Houston, TX, US	25	18,241	98.85%	94.95%	82.83%	17.17%	72.8

Figure 3. Flight average delay time

As a result, we can see that the delay situation in international airport is still very severe nowadays, which may disturb people a lot, especially for passengers on long distance or international travel. In fact, as international students, we also experienced ight delay when we came to the United States. So we want to build a model through the real world datasets for the next unknown flight forecasts whether it is delayed or not as a result which can provide a reference for people's travel planning.

We obtain the real world dataset from Kaggle, there are many features that is very common seen in the ticket or

[1] https://www.flightstats.com/v2/

found out through website and can affect the ight in the real life, for example, month, day, departure time, ight company career, ight origin or destination, the distance and so on. So in our project, after data preprocessing, we want to establish a supervised model on the training set with machine learning algorithms, such as k-Nearest Neighbors, decision tree and ensemble methods. Use the validation set and to evaluate the predictive model and change hyperparameters. At last we use the independent test set to calculate the accuracy and get final performance estimate.

2. Related Work

Establishing a model to predict flight delays in order to plan travels better is a work that has lasted for years. Sternberg, Soares, Carvalho, and Ogasawara (2017) presents a literature review of approaches in flight delay prediction. At first, statistical analysis and probabilistic models, such as regression, statistical inference, econometric models and density functions, are used for prediction. In recent years, as machine learning has become an important and powerful method of classification and regression, the researches on flight delay prediction are increasing the usage of machine learning methods. Commonly used algorithms are k-Nearest Neighbors, neural networks, SVM, and random forests [12].

There are several related researches about flight delay prediction. A large scale alarm of flight delays was from Lu, Wang and Zheng (2008). They combined unsupervised learning and supervised learning. After using clustering to get a standard of different "levels" for each delay class, they did supervised learning such as C4.5 decision tree to get prediction of new "levels" [14]. Choi, Kim, Briceno and Mavris (2016) focused on delays caused by inclement weather and compared prediction accuracy and receiver operating characteristic (ROC) curves of several common algorithms such as decision tree, random forest, adaptive boosting and k-Nearest Neighbors. Also, sampling techniques, SMOTE and random under-sampling, were applied in order to overcome the disadvantages of imbalanced data. They found random forest performs better than other algorithms [5]. Rebollo and Balakrishnan (2014) presented a new class of models for predicting delays, incorporated both temporal and spatial (airline network) delay states as explanatory variables, and applied Random Forest to predict the departure delays 2-24 hours in the future [11].

Compared with the work of Choi et al., our project also perform and compare different classifiers and need to deal with imbalanced data to get a more robust result. However, we use the flight information such as departure or arriving time and origin or destination as features instead of weather. In feature part, the study of Rebollo and Balakrishnan is more similar because we both focus on the information of the flight itself.

3. Proposed Method

As our project wants to solve a classification problem, we fit several classifiers including k-Nearest Neighbors, decision trees and random forest. In addition, we use some resampling methods to balance our train dataset such as SMOTE because there is a big difference in the number of labels. Last, recall and accuracy are considered as two evaluation metrics in our project. Detailed reasons and methods are in the following parts.

3.1. Evaluation metric

Let A represent event "flight is delayed" and B represent event "flight is predicted to be delayed". Then, the Bayes formula is

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

where $P(B|A)$ means the probability of a flight is predicted to be delayed when it is actually delayed. Then,

$$P(B|A) = \frac{TP}{TP + FN} = Recall$$

Therefore, in order to improve the probability that the flight is truly delayed after we predict it will be late, with the condition that $P(A)$ mainly depends on the new data, which is truly random, $P(B)$ depends on different models we choose, recall should be the common model evaluation judgment for different model. In other words, we want to train our model to get higher recall with higher accuracy.

A clearer example to illustrate is that we always want to bring our umbrella when it is rainy outside, so we do not mind taking our umbrella on some sunny days.

3.2. SMOTE

SMOTE, synthetic minority over-sampling technique, is an easy comprehension over-sampling method. Through SMOTE, the minority class is over-sampled by creating "synthetic" examples [4]. For every minor sample x, select its k-nearest neighbors among the minority class, determine the amount of SMOTE $N\%$ according to the proportion of minority class, then synthesize new samples with:

$$x_{new} = x + randn(0, 1) * |x - x_{neighbors}|$$

Figure 4 show this process visually[2].

3.3. SMOTE+Tomek links

Although SMOTE avoid overfitting and spread the decision boundaries for the minority class to the majority class space to balance class distributions, the majority class might

[2] https://imbalanced-learn.readthedocs.io/en/stable/auto_examples/index.html

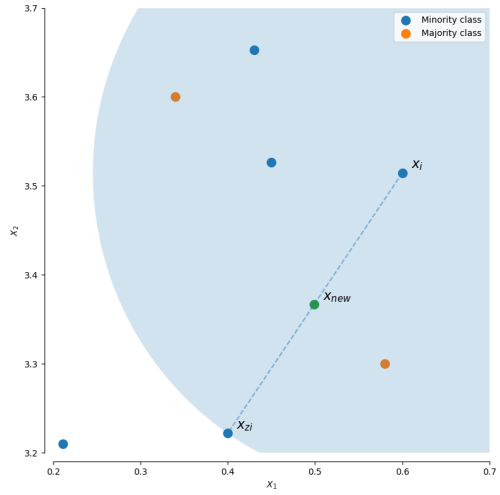

Figure 4. The illustration of SMOTE

be invaded deeply by synthetic minor samples, which leads to class clusters not well defined and an overfitting classifier. Thus, in order to solve this problem, Tomek links can be used to clean the over-sampled train set [2].

Two examples E_i, E_j from two different classes form a Tomek link if there is not an example E_l s.t. $d(E_i, E_l) < d(E_i, E_j)$ or $d(E_j, E_l) < d(E_i, E_j)$, where $d(\cdot)$ represents distance. If E_i and E_j form a Tomek link, then one of them is noise or both are on borderline [1].

3.4. k-Nearest Neighbors

We use kNN algorithm as a beginning point. The kNN algorithm categorizes a data point depending on a plurality voting of its k neighbors. The distance function is showing below. We use Euclidean distance (L^2), and treat dep_time and distance as continuous features in the whole data.

$$d(x^{[a]}, x^{[b]}) = \sqrt{\sum_{j=1}^{2}(x_j^{[a]} - x_j^{[b]})}$$

where $x^{[a]}$ and $x^{[b]}$ are two data points in the 2-dimensional space. And each x_j represents different feature. The algorithm select the first k neighbors of each point in test dataset and it determines the predicted label of the test point by performing a plurality voting among k points.

Besides, there are many different weighting methods, such as rectangular, triangular, Epanecknikov, Gaussian, rank and optimal.

$$h(x^{[t]}) = \frac{\sum_{i=1}^{k} w^{[i]} f(x^{[i]})}{\sum_{i=1}^{k} w^{[i]}}$$

We perform 2-fold cross validation to determine the best value of k and the best weighting method in the kNN algorithm, so as to avoid overfitting. The training dataset is split into 2 sub-partitions. As a result, the model is trained using one of the sub-partitions and validating using the remaining part of the data.

Before having any procedure on the data, we calculate the first and the second principle components by Principle Component Analysis (PCA) to cope with this problem. A possible way is to use SVD method to get the best differences in the two dimensions.

Furthermore, with too many labels in the whole dataset, we want to do some dimension reduction for the label feature to choose the best descriptions. The best description means for data having these labels, it is more clearly to determine whether or not a flight will delay through the data having same labels in the training set.

A more general question here is to find whether we can make the label having more value sense, which we can not deal well with. After checking some references, surely, it can give these labels some value sense by choosing some distributions for them. However, it is not as useful as we think. The feature selection part mainly uses codes in the GitHub we discuss during the class[3]. Then, with these label features having more accuracy, we divide the whole kNN to several smaller kNN parts to have better prediction.

A package named "kknn" in R[4] gives a more visualized way to help us understand parameter k and different weighting methods.

Besides, Richard A.Johnson (2002) talks more about principle component analysis and the question we mention above. He mentioned that giving some distribution to the label is not much useful [6]. Furthermore, Muhammad Ejazuddin (2014) mentioned more details about different weighting method and essential procedures [13].

3.5. Decision Tree

Decision tree that is a classical method used in classification problems, especially when dealing with this problem whose variables are categorical variables. And based on the characteristics of this algorithm, it requires less data cleaning compared to some other modeling techniques.

In our project, we implement decision tree in order for this binary tasks. From the point of view of data structure, decision tree, obviously is a tree structure (could be a binary tree or non-binary tree). Each non-leaf node represents a test on a feature, each branch represents the output of the feature in a range, and each leaf node stores a category.

The process of using the decision tree to make a decision is to start from the root node, test the corresponding feature in the sample to be classified, and select the output branches according to their values until they reach the leaf

[3] https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/13_feat-sele/code/13_feat-sele__code.ipynb

[4] https://rdrr.io/cran/kknn/man/kknn.html

node, and take the category stored in the leaf node as the decision result. According to this principle, we can easily and intuitively get the category judgment of an example. as long as we know the specific values of each feature, the decision tree decision process is equivalent to the traversal of the tree from the root node to a certain leaf node. How to traverse each step is determined by the specific feature value or attributes of each feature of the data.

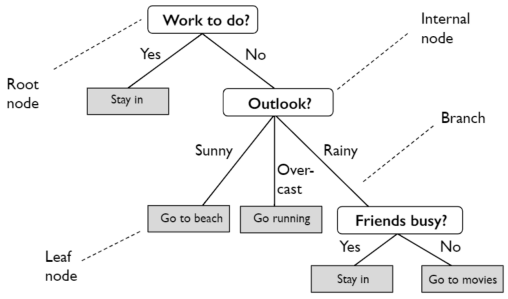

Figure 5. Example illustrating how decision tree works.

The core process in this algorithm is how to construct the tree. In fact, we usually think of the construction of decision trees as an application of recursive algorithm. Starting from the root node, we select a feature to get the maximum information gain when the parent node is splited. And then do the same thing to their child nodes.

Below is a more formal expression of the algorithm outlined above [9]:

GenerateTree($\mathcal{D}$):

- if $y = 1 \; \forall \; \langle x, y \rangle \in \mathcal{D}$ or $y = 0 \; \forall \; \langle x, y \rangle \in \mathcal{D}$:

  - return Tree

- else:

  - Pick best feature x_j:

    * $\mathcal{D}_0$ at $Child_0 : x = 0 \; \forall \; \langle x, y \rangle \in \mathcal{D}$:
    * $\mathcal{D}_1$ at $Child_1 : x = 1 \; \forall \; \langle x, y \rangle \in \mathcal{D}$:

  return Node(x_j,GenerateTree($\mathcal{D}_0$),GenerateTree($\mathcal{D}_1$))

The general way to compute the information gain are based on the formula below:

$$\mathcal{GAIN}(\mathcal{D}, x_j) = H(\mathcal{D}) - \sum_{v \in Values(x_j)} \frac{|\mathcal{D}_v|}{|\mathcal{D}|} H(\mathcal{D}_v)$$

Where $\mathcal{D}$ is the training set at the parent node, and $\mathcal{D}_v$ is a dataset at a child node upon splitting.

In fact, different function $H()$ in computing the information gain could result in different trees.Three are two popular way to define information gain in a mathematical way.They are:

- Gini
- Entropy

In experiences, we would use method of best hyperparameter search to find which one is more suitable for our model.

In spite of so many advantages of this algorithm, however [8], decision tree is easy to be overfitting, some proper constrains on model parameters and pruning needed to be processed in our next step of experiments.

3.6. Random Forest

Apart from k-nearest neighbors and decision tree, we also consider random forest, an ensemble method. It uses a random selection of features to split each node and because of that, random forest decreases overfitting and gets more robust result with respect to noise. The generalization error of random forest is:

$$PE \leq \frac{\bar{\rho}(1 - s^2)}{s^2}$$

where $\bar{\rho}$ is mean value of correlation between trees and s is the strength of trees [3]. From the formula of generalization error, decreasing correlation ρ among trees and increasing strength s of trees can get smaller error. with usage of bagging and randomization, random forest has better performance[5].

4. Experiments

In this section, we describe the dataset that we use to analyze flight delay, and specific methods we used for three different method.

4.1. Dataset

We obtain our dataset from Kaggle's mlcourse.ai: Flight delays.There are 100,000 items and 8 features all without missing data. The features includes like distance for each journey, where it began and where it landed, and in what airways.

The huge data set brings inconvenience in feature processing. On the one hand, we want to get more information from the limited 8 features to fit our data, on the other hand, because we use python and sklearn as algorithm tools, we have to convert categorical-type data into dummy variables, which brings the problem of high feature dimension. Because our dataset is reality-based, before any feature selection algorithms being implemented, we use the data table Airport.csv downloaded from the our Airports [7] website to extract information from the departure and origin features

[5] https://github.com/rasbt/stat479-machine-learning-fs19/blob/master/07_ensembles/07-ensembles__notes.pdf

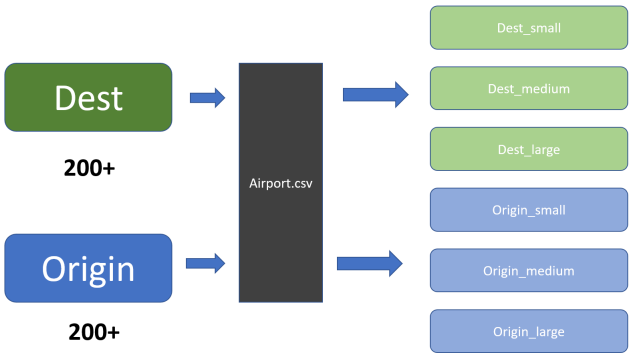

Figure 6. Transformation of feature

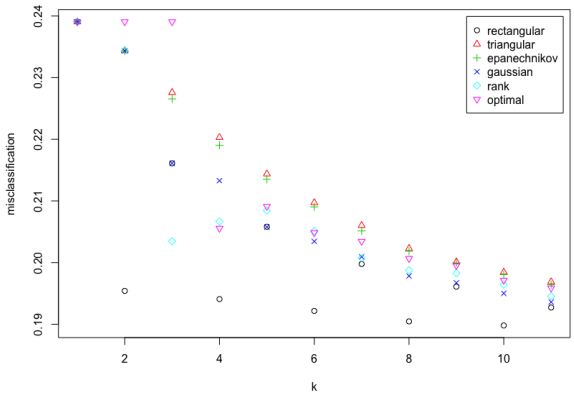

Figure 7. Misclassification with different k and different weighting method

of the dataset, which each contain more than 200 different airports, including the size of each airport, and so on. Show in Figure 6 We hope to use such kind of method to achieve a function similar to PCA, that is, the dimension reduction of categorical data.

Finally,since the target is just binary label whether is 'N' (no delay) or 'Y' (delay). So we can use these 8 features to determine whether a flight would delay or not. But another important thing that needs to be noted is that our data set is imbalanced, that is to say, only about 20% of items is labeled with delay. So the in the real work process, we will choose different methods for each three algorithms to deal with this problems.

4.2. Software

- Python 3.7
- Jupyter notebook
- R studio
- Excel
- Sklearn 0.21.3
- Mlxtend

4.3. k-Nearest Neighbors

First, with 2-fold cross validation, the results for different k and different weighting method is showing in the Figure 7.

As shown in the Figure 7, it is easy to find that with k=10 and use rectangular weighting method, which means each points given same voting power for the final decision, the precision goes higher.

Then, about feature selection for label feature, with same divided method mentioned in the part of Dataset and the code mentioned given in the GitHub, the selected features are 'DayofMonth', 'DayofWeek', 'Origin_AMA', 'Origin_JFK', 'Origin_OGG', 'Origin_STL', 'Dest_ALT' and 'Dest_EWR'. The reason we choose these features is that there accuracy is higher than the proportion of not delayed flight in the train data.

However, we do not want to use these selected features to separate our data to several parts. First, the procedure to get this features is too much costly. We run the code nearly 36 hours and we only get these features which are not all the features satisfied the condition. Besides, the accuracy is from 81% to 80.5%, which changes too little compared to the huge dataset. Therefore, we do not use the results of feature selection.

Last but not least, considered our principle in the model evaluation, in order to get a higher recall, we want to give each point with delayed label a higher 'voice'. For example, in k nearest neighbors of each test points, every delayed point should have more voice to determine whether or not the test point is delayed. This is different from weighting method, which deals with Euclidean distance. Then, calculate how much 'voice' each training point should have in order to have relatively high recall and accuracy (Here we want to make accuracy*recall as large as enough). The result is showing in Figure 8.

Therefore, we give voice of each delayed point four times larger than the not delayed points.

4.4. Decision Tree

Our dataset has 100,000 items, and after we encode the features into dummy variables, the number of features reaches more than 600. Given the fact that our dataset is imbalanced, we use a special way in addressing this problem.Figure 9 shows part of this process.Inspired by bagging algorithm, first, we implement holdout-method to split the dataset into train-set and test-set. And then, we split the data in training set with label "N" into 4 groups, combined each of them with the data with label "Y".All of this process are

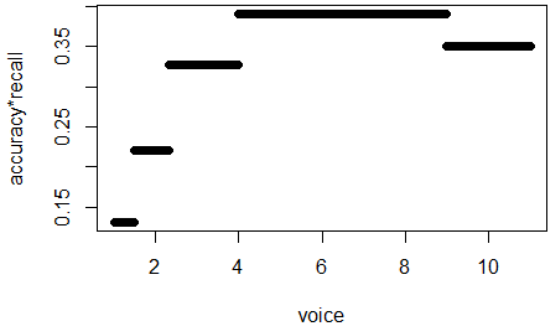

Figure 8. Give delayed training points different voice

implement with under the constrain that keeping the ratio of two label unchanged. So we will have 4 different, balanced, and smaller dataset.

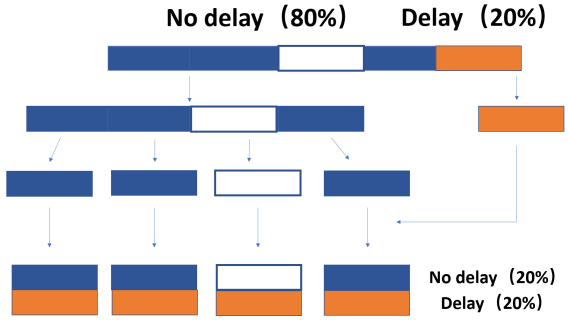

Figure 9. Get balanced data

Our decision tree algorithm is based on sklearn 0.21.3 and is applied to each small data set. And considering the problem of overfitting of the decision tree, we use the sequential feature selection method [10] for feature selection and the grid search method for parameter selection in the fitting process of each model, which provide the best features, the most suitable depth of the tree, the largest number of leaf nodes and the most suitable calculation method of information gain for each small decision tree.Figure 10 shows this process.

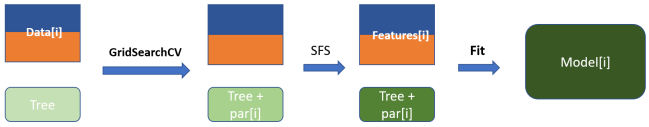

Figure 10. Feature selection of decision tree

And finally, we use the model.fit.prob () function to predict each model, combine four different results with the mean method, and draw the final conclusion with 0.5 as the threshold.

4.5. Ramdom Forest

Repeat minority class First consider simply repeat the minority class samples, which means setting class weight to "balanced". In hyperparameter tuning, we use grid search cross validation on train set, and found the depth of trees will has a great influence on the recall and accuracy. As Figure 11 shows, we set the max depth of trees to 15 based on the results of train and validation set.

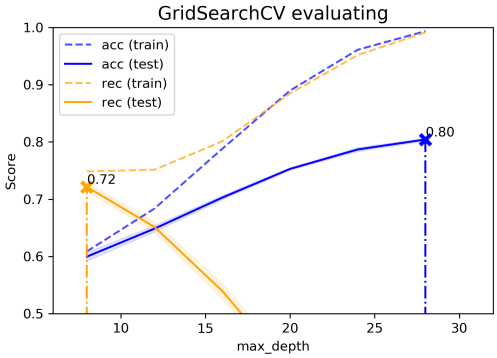

Figure 11. Grid search cross validation without resampling

SMOTE Second, we use SMOTE to balance the train set. Then set the max depth to 28 according to the grid search cross validation shows in Figure 12. Also, we can see both recall and accuracy of the validation set have big variance, in other words, they are sensitive to the selection of validation set. More details will be covered in the following result part.

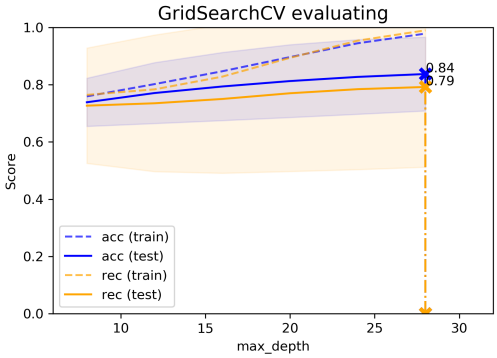

Figure 12. Grid search cross validation with SMOTE

SMOTE+Tomek links Last, we use SMOTE to oversample the train set, then delete the Tomek links in the resampled train set. Again, as the grid search shows in Figure 13, we set the max depth to 28. In addition, after deleting Tomek links, the grid search result seems not change, which

may suggest that using SMOTE and SMOTETomek will not make a difference in the result.

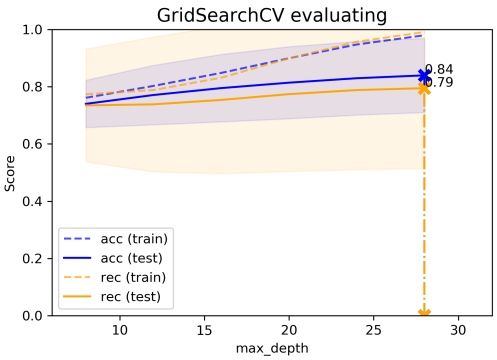

Figure 13. Grid search cross validation with SMOTETomek

5. Results and Discussion

5.1. k-Nearest Neighbors

After procedure mentioned in the previous experiment part, the accuracy is 56.27% and the recall is 0.69.

5.2. Decision Tree

By using the method of decision tree, we finally get a recall, of 0.695 and ensure that the accuracy is about 62.45%. What's more,the AUC of the ROC curve (Figure 14) can reach 0.71, but there is still a lot of ways for improvement. Given such a large and unbalanced data set, it is gratifying to use only a single decision tree to achieve this effect. But at the same time, through the evaluation of each small model by methods such as learning curve (Figure 15), we can see that there are still signs of overfitting in the model.

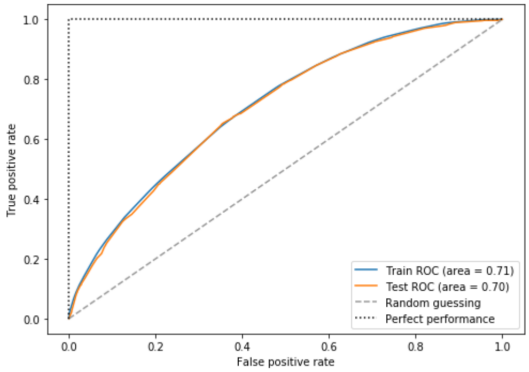

Figure 14. ROC curve of one specific model.

5.3. Random Forest

Repeat minority class When we do not use resampling method and just repeat the minority class, the random forest

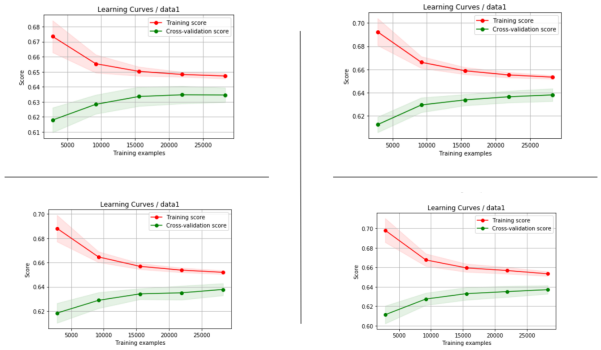

Figure 15. Learning curves for different 4 models.

can achieve a test accuracy of 68.23% and a recall of 0.59, the confusion matrix is shown as Figure 16.

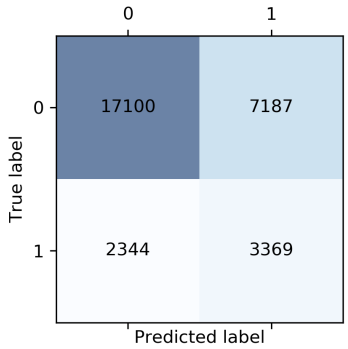

Figure 16. Confusion matrix without resampling

SMOTE When we use SMOTE to resample the train set, the random forest can achieve a test accuracy of 77.04% and a recall of 0.28, the confusion matrix is shown as Figure 17. From the result, we can find that the accuracy is higher but recall decreases a lot. In the previous experiment part of random forest, Figure 12 shows the accuracy and recall of validation set have high variance, now the recall of test set is even lower than $\mu_{recall} - \sigma_{recall}$, which means using SMOTE may cause some problems in the generalization performance of random forest. In other words, there is some kind of "overfitting" problem after performing SMOTE to the train set.

SMOTE+Tomek links Table 1 shows difference of different resampling methods. As we suggest before, Figure 12 and 13 are same indicates deleting the Tomek links does not make a difference to the result. In fact, the results of SMOTE and SMOTETomek are same. The reason might be that random forest itself can deal with Tomek links as it is an ensemble method and will not be influenced by some problematic samples.

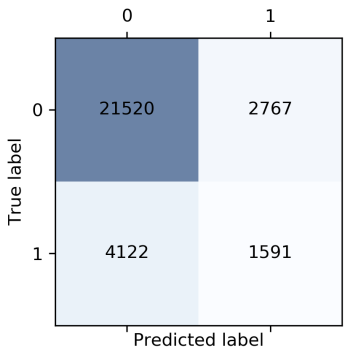

Figure 17. Confusion matrix with SMOTE

Method	label 0	label 1
Original Train	56669	13331
SMOTE	56669	56669
SMOTETomek	54560	54560

Table 1. The number of labels after resampling methods

6. Conclusion

In our project, we demonstrate a variety of methods to predict the problem of flight delays. We use three machine learning algorithm to solve it including KNN, decision tree and Randomforest. Furthermore, other machine learning techniques such as PCA, bagging are applied for a better result. We also use both recall and accuracy as criterion to evaluate our model in order to achieve our goal.

Based on the result, we conclude that Randomforest and decision tree are more suitable for our project. Randomforest keeps a high accuracy with low variance and the decision tree maintains a high recall with relativity fair accuracy.

In the future, we hope more researchers could provided more detail about one flight journey. For example, as we know, weather is a very important factor affecting flight delays, so we hope weather forecasts can make more accurate forecasts, so as to provide more and more accurate information for researchers and better prepare passengers for their long-term travel plans.

And finally, we hope that our model can be used in the mobile phone APP to connect with the airline database, so that passengers can easily know the on-time probability of the plane they are expected to take by simply entering their flight number in the APP.

7. Contributions

Jingshan Huang found the original dataset in Kaggle that contains 100,000 items and completed the preliminary data cleaning work. He also code the decision tree model in a deep way. Besides, he found the important dataset from web that help us extract more information from the dataset and created a special way in dealing with imbalance data.

Yuan Cao uses the random forest algorithm to further improve the results of the decision tree and optimize the overfitting problem. In dealing with the problem of unbalanced data sets, he uses a variety of advanced methods and their deformations (such as SMOTE method), fits a variety of models and compares them in different ways.

Runze You consulted the relevant literature and provided strong theoretical support for this project. Through python and R, he also used the k-nearest neighbors algorithm in a strong way. All of them worked together and finished the project report.

References

[1] Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(11):769–772, Nov 1976.

[2] G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1):20–29, 2004.

[3] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[5] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris. Prediction of weather-induced airline delays based on machine learning algorithms. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pages 1–6. IEEE, 2016.

[6] R. A. Johnson, D. W. Wichern, et al. *Applied multivariate statistical analysis*, volume 5. Prentice hall Upper Saddle River, NJ, 2002.

[7] T. I. Murphy. Line spacing in latex documents. `https://ourairports.com/data/airports.csv`. Accessed April 4, 2010.

[8] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description lenght principle. *Information and computation*, 80(3):227–248, 1989.

[9] S. Raschka. *Python machine learning*. Packt Publishing Ltd, 2015.

[10] S. Raschka. Mlxtend: Providing machine learning and data science utilities and extensions to python's scientific computing stack. *The Journal of Open Source Software*, 3(24), Apr. 2018.

[11] J. J. Rebollo and H. Balakrishnan. Characterization and prediction of air traffic delays. *Transportation research part C: Emerging technologies*, 44:231–241, 2014.

[12] A. Sternberg, J. Soares, D. Carvalho, and E. Ogasawara. A review on flight delay prediction. *arXiv preprint arXiv:1703.06118*, 2017.

[13] M. E. Syed. Attribute weighting in k-nearest neighbor classification. Master's thesis, 2014.

[14] L. Zonglei, W. Jiandong, and Z. Guansheng. A new method to alarm large scale of flights delay based on machine learn-

ing. In *2008 International Symposium on Knowledge Acquisition and Modeling*, pages 589–592. IEEE, 2008.