San Francisco Crime Classification

Ruixue Zhang 20619404 r267zhan@uwaterloo.ca
Yican Shi 20618295 y233shi@uwaterloo.ca
Mengzhen Gao 20610458 m28gao@uwaterloo.ca

ECE 657A
Electrical and Computer Engineering
University of Waterloo

Abstract. In this paper, we do data analysis on incidents from 2003 to 2015 in derived from SFPD Crime Incident Reporting system[1] in San Francisco. The primary goal of crime data analysis is to find the hidden pattern of crime information and then make meaningful summary in a practical way. The project is classification. We use supervised classification algorithms like Naive Bayes, k-Nearest Neighbors and Logistic Regression. We compare the Multi-class logarithmic loss[2] and accuracy and then conclude the pros and cons of the algorithms.

Keywords: Data Analysis, Crime, Classification, Naive Bayes, k-Nearest Neighbors, Logistic Regression

1 Introduction

From 1934 to 1963, San Francisco was infamous for housing some of the world's most notorious criminals. However, rising wealth inequality, housing shortages increase the number of incidence. For the safety of San Francisco, we can do crime analysis to prevent the potential crime from happening. From Sunset to SOMA, and Marina to Excelsior[3], the crime dataset provides 12 years' crime reports from across all of San Francisco's neighborhoods from 2003 to 2015.

Our task is to predict the category of crimes that occurred in the city by the bay. The result of the project can tell police that which kinds of incidence often happen and distribution of top crime categories on San Francisco map. This can improve police efficiency and save their time. Given time and location, the goal of the project is that we predict the category of crime that occurred. We use several supervised classification algorithms such as Naive Bayes, k-Nearest Neighbors and Logistic Regression. We implement them on San Francisco Crime dataset. We compare the efficiency and accuracy and then conclude the pros and cons of the algorithms.

In Section 2 we describe the crime dataset of San Francisco and the result of data visualisation. Section 3 is brief review of literature on the selected methods and their application to similar problems. We discuss the method selected with details on the options and parameters in section 4. Section 5 is implementation such as Software used, data structures, program structures, data representation

and any special set up needed. We talk about test cases on the selected datasets and evaluation of the performance in comparison with base line methods. Section 6 is discussion of results and conclusions.

2 Datasets and Data Visualisation

2.1 Datasets

We do data analysis on incidents from 2003 to 2015 in derived from SFPD Crime Incident Reporting system[1]. Training Data has 434480 records of crimes. And there are 884261 rows in testing dataset. There are 9 features in training data and 7 features in testing data:
Dates: detailed time of the crime incident
Category: category of the crime (only in training data)
Descript: description of the incident (only in training data)
DayOfWeek: the day of the week
PdDistrict: which Police Department resolved the incident
Resolution:: how the crime incident was resolved
Address: :detailed address of the incident
X:Longitude
Y:Latitude

2.2 Data Visualisation

Further exploration for the dataset, we found some other information from more than 878,000 records of different crimes. This data set covers a wide variety of crime, from Figure 1, we see that there are 39 categories in total. Among these categories, larceny/theft, other offenses, non-criminal, assault, and drug/narcotic rank top five in frequency. Statistically, top five crime categories make up about 66 of all recode, which means that several mostly occurred crimes make up the majority. We conclude that it is necessary for the police to put more force on dealing with major top five crimes.

In addition, we plot to analyze number of total crimes occurred in each of ten police department districts. We can see, from Figure 2, that Police department of Southern district deals with most amount of crime. Follow with Mission district, Northern district, Bayview district and other six districts. Richmond district has the least number of crimes, which is around one forth of Southern district.

Apart from above exploratory analysis, we also attempt to have a geometrical view of the dataset so we plot scatter of crimes as well as density line on San Francisco map. From Figure 3, it could be inference that crimes occurred mostly in northern part of the city. After some online searches, we found that the northern part is the downtown of San Francisco. It is consistent with what we expected from the former two plots.

Moreover, we are interested in the amount of total crimes occurred hours of day. As shown in Figure 4, crime rate is relatively low between 3 am and 6 am, and reach to their peak at 12 pm, and second peak around 5 pm to 6 pm.

Then, we want to know whether there is correlation between number of crimes of each of ten police department districts and the hour of the day. As indicated in Figure 5, there is a valley through wave of all ten districts at around 5 am and most of crimes break out in afternoon. At about 12 pm there is a peak in crime number, which may be caused by chaos of afternoon break. We notice that all ten districts have the same pattern, which is consistent with Figure 4. Therefore, we suggest that police can concentrate their force in some crime peak times.

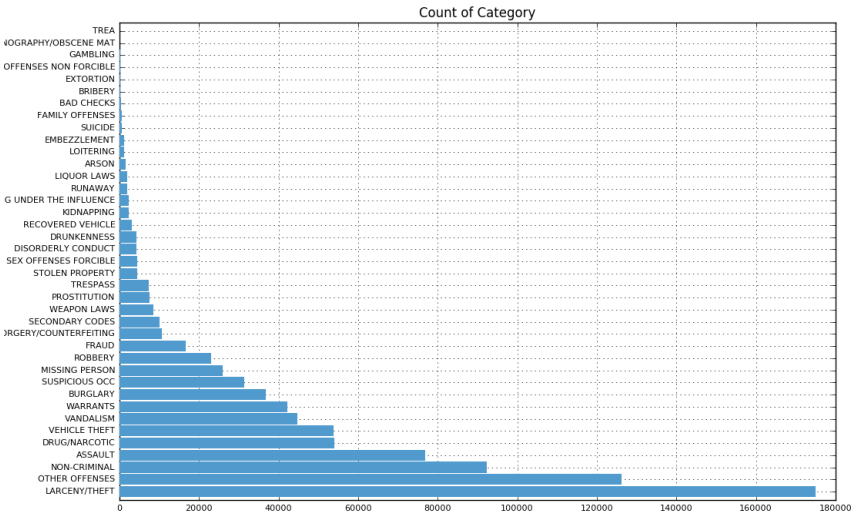

Fig. 1. Crime Category Distribution

3 Brief Review of Literature

Data mining is a powerful tool that enables criminal investigators who may lack extensive training as data analysts to explore large databases quickly and efficiently[4]. The primary goal of crime data analysis is to find the hidden pattern of crime information and then make meaningful summary in a somehow practical way. Some of the most widely used techniques are classification, clustering and so on.

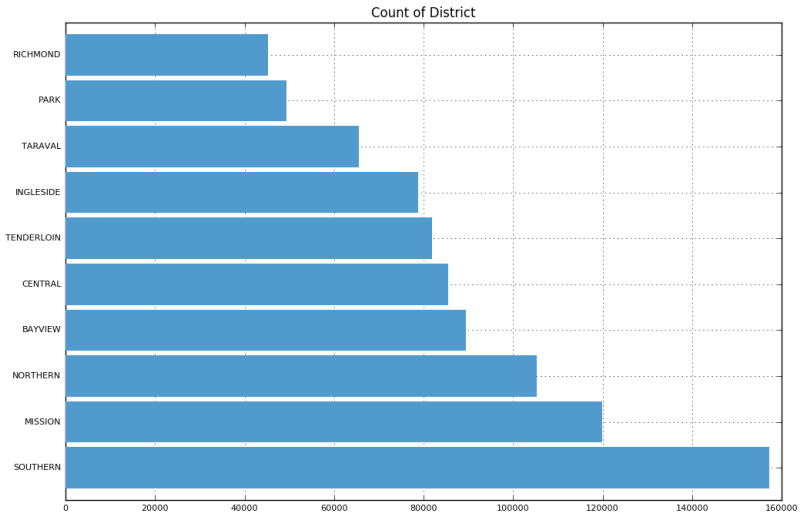

Fig. 2. Crime Category by District

There are some related researches on crime data mining, but not that much. Chen Hsinchun from University of Arizona[5], studies several different data mining techniques including entity-extraction techniques to analyze the behavioral patterns, association, clustering, neutral networks and pattern visualization, then matches them with most suitable crime types for analysis, such as traffic violations, sex crime, theft, fraud, arson, gang/drug offense, violent crime, and cybercrime which can facilitate police work and enable investigators to save their time to other valuable tasks.There is some related research on crime data mining, but not that much. Iqbal Rizwan from Universiti Putra Malaysia[6], covers investigation of data pre-precessing and classification methods on crime dataset acquired from UCI machine learning communities and discusses the results of the classification algorithms for predicting the Crime Category attribute. Basically, this paper applies two classification algorithms Naive Bayes and Decision Trees, to put U.S states into different categories such as low, medium and high crime rates, mainly based on median household income, population density, unemployment rate and population that is under poverty threshold. The results are evaluated by accuracy (correctly classified instances), precision, recall values and F-measure for both of the algorithms. For decision tree, the accuracy and precision are 83.9519 and 83.5, which performs better than Naive Bayes with accuracy and precision of 70.8124 and 66.4 respectively. Shyam Varan Nath from Florida Atlantic University[7], uses clustering algorithm to detect crime patterns. Along aside clustering method, the research also uses k- means clustering

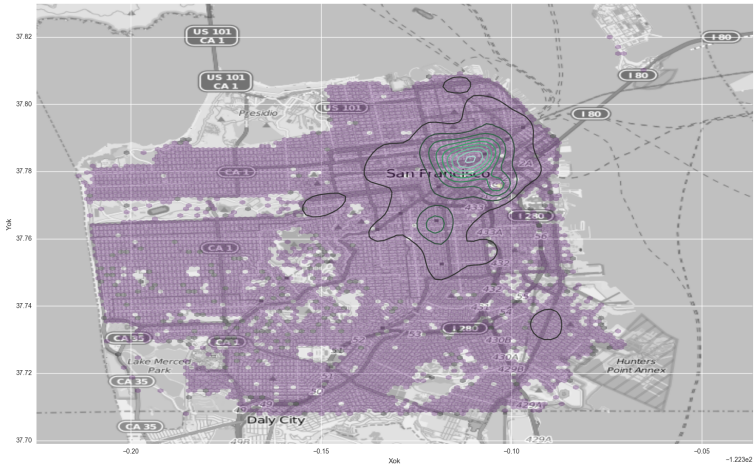

Fig. 3. Crime Density Map

with some enhancements to assist the process of pattern identification, semi-supervised learning technique to help increase predictive accuracy and weighting scheme for attributes to deal with limitations of clustering tools and techniques. The proposed combined method mentioned above is applied with geo-spatial plot to display the result graphically. As a result, significant attributes have been identified and crime patterns have been formulated.

4 Description of Classification Methods

The main approach of this project consists of three parts, preprocessing the dataset, building classifiers through applying three algorithms to training set, doing evaluation through test set and comparing the performance of three algorithms.

When preprocessing the dataset, we need to select features from the given dataset at first. As to the features with categorical data or word-based description, we need to translate these data to numerical representation. For some

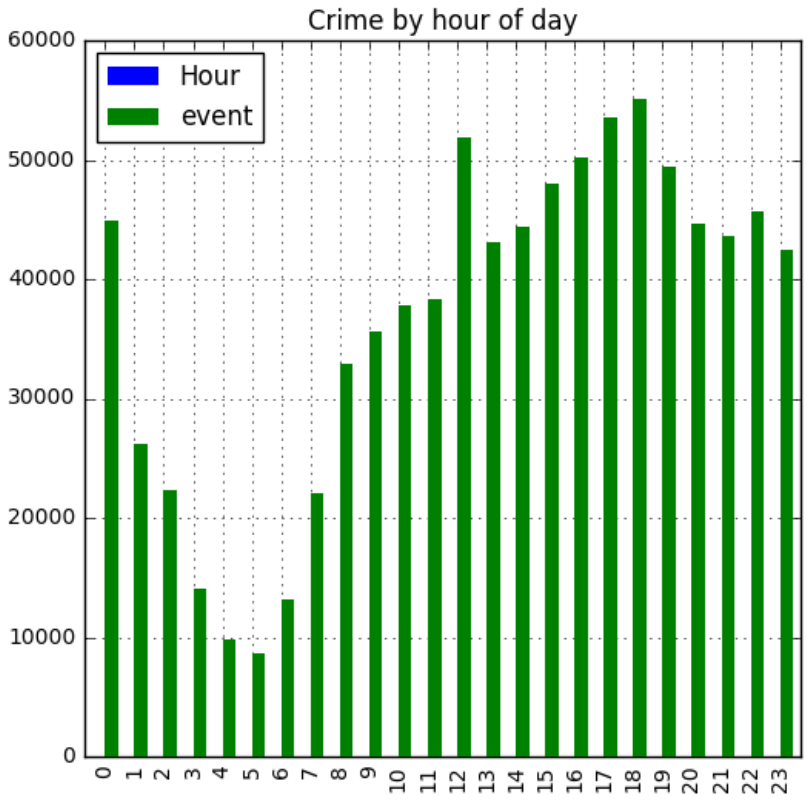

Fig. 4. Crime by Hour of Day

algorithms, we will do further vectorization of the data to fit the requirement of the specific algorithm. No missing values are detected in dataset. Whether preprocessing outliers or noise and whether doing feature reduction depend on the previous step.

The next part is to build classifiers. Three algorithms will be implemented, Naive Bayes, k-Nearest Neighbors (kNN) and Logistic Regression.

4.1 Naive Bayes

Nave Bayes classifier is a simple probabilistic classifier based on Bayes theorem with strong independence assumptions between the features[8]. If have training data, we can learn the prior and conditional from the training data (freq. prob.) we assume distributions and learn parameters using MLE or other methods. In

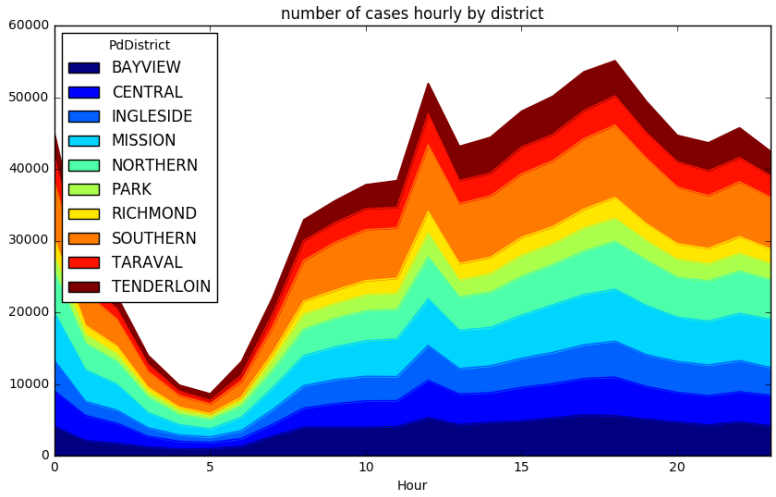

Fig. 5. Crime Hourly by District

Naive Bayes classifier, optimum in the sense of probability of error that for given prior probabilities, loss function and class-conditional densities, no other decision rule will have a lower risk (expected value of the loss function, for example, probability of error). In practice the class-conditional densities are estimated using parametric and non-parametric methods.

The features in the San Francisco crime datasets are time and different geographical information, which obviously are independent.

4.2 k-Nearest Neighbors

K-Nearest Neighbors classifier is a simple example of a supervised, non-parametric, classification model. In k-NN classification, the basic concept is that an object is assigned to the class most common among its k nearest neighbors[9]. For some test datapoint x define a set of K points in training set nearest to x. We count how many members of each class are in the nearest neighbors set and return empirical fraction for each class as a probability Optionally take highest probability as class label for x. k-Nearest Neighbors Classifier does not need details of model construction need to be considered, and the only adjustable parameter in the model is k, the number of nearest neighbors to include in the estimate of class membership.

K-Nearest Neighbors classifier is non-parametric because the number of neighboring points used depends on the data. The mothod is biased towards classes with more samples. And it is sensitive to noise in the data (eg. random neighbors) and computationally expensive in large dimensions or large K. We could compute distance using only some dimensions and remove redundant points from

training set (eg. ones surrounded by same class). For low dimensions could use search trees to organize training points into independent subsets.[10]

4.3 Logistic Regression

Logistic Regression can be used as a simple linear classifier in the same way we did with Naive Bayes. Logistic regression predicts the probability of an outcome, the appropriate class for an input vector or the odds of one outcome being more likely than another. Logistic Regression classification compare probabilities of each class and treat the halfway point on the sigmoid as the decision boundary.

Logistic Regression classification is relatively easy to fit to data using gradient descent, many methods for doing this. Learned parameters can be interpreted as computing the log odds. It is known as a parametric method. This distinction is important because the contribution of parameters in logistic regression (coefficients and intercept) can be interpreted.[10]

4.4 Evaluation Formula

In the last part, we evaluate the performance of our three classifiers through the method provided by Kaagle, using the multi-class logarithmic loss[2]. Each incident in test set has been labeled with one true class. We will also provide the evaluation model with a set of predicted probabilities (one for every class). The *formula* is then,

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \tag{1}$$

where N is the number of cases in the test set, M is the number of class labels, log is the natural logarithm, y_{ij} is 1 if observation i is in class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j. At same time, when testing with different methods, we also divide the training set into two parts, training and validation with proportion of 7/3 and compute the accuracy to provide some evaluation. The last step is to comparing the performance of the three models and choose the best one.

5 Implementation of Algorithms

We use programming language of python to implement three classification algorithms under library scikit-learn, imbalanced-learn. The flow of program is divided into four parts, reading csv file, dealing with original data (prepossessing, selecting features, dimensionality reduction and balancing data), applying algorithms and exporting results into csv file.

5.1 Data Preprocessing

In data preprocessing part, we firstly check missing values with nothing found. The next step is to manipulate with outliers. Some data points are identified with latitude as 90 degree, which is impossible in terms of location in San Francisco. These data points are naturally deleted. Some data are detected with weired address representation, such as different order of street name or different address with same latitude and longitude. These are left unsolved temporarily because of small amount and little effect on feature selection.

5.2 Feature Extraction and Selection

As seen in datasets section, we have nine features to do further extraction and selection. Two kinds of feature sets are eventually tested. First feature set is very elaborate, consisting of 'DayofWeek', 'PdDistrict', 'Year', 'Month', 'Day', 'Hour', 'Minute', 'StrNo', 'Inter', 'Address', 'X', 'Y'. Second feature set is trying to reduce the scatter of data to avoid overfitting, with features 'DayofWeek', 'PdDistrict', 'Year', 'Season', 'Time', 'StrNo', 'Inter', 'Address', 'X', 'Y'. In both sets, 'Resolution' and 'Descript' are deleted with no appearance in test sets. 'Category' is obviously used to label different crime classification.

In feature set 1, 'Year', 'Month', 'Day' and 'Hour' are extracted from 'Dates', 'StrNo', 'Inter', 'Address' from original 'Address'. 'StrNo' represents the number in address. If there is no number in address, 0 will be used to replace. 'Inter' is used to differentiate whether the location is at the intersection of two streets. New 'Address' is the remaining part in original 'Address'. 'DayofWeek' and PdDistrict' keep meaning unchanged but being transformed into numerical values as scalar. 'X', 'Y' are still representing longitude and latitude with nothing fixed.

? ? As to feature set 2, 'Season' is converted from 'Month' in the first feature set with 'Month' being classified into four parts and deleted at last, similar as 'Time' to 'Hour' with three parts. Data of 'X' and 'Y' are rounded to the 3rd digit after decimal point. Other features with same names are exactly same as the first feature set.

5.3 Data Imbalance

As seen in data visualization, some categories of crime have tons of data points to train the model but some categories are lack of data points. There are two fundamental approaches to solve data imbalance, oversampling and undersampling. The basic idea of oversampling is to generate extra data according to the primitive data of the classes with fewer data points. Undersampling is and opposite technique, cutting down some data of those classes with excess data. In our project, we use Synthetic Minority Over-sampling Technique (SMOTE) to eliminate the effect of data imbalance. According to Wikipedia[11], the work flow of this method is first to take the vector between one of those k neighbors, and the current data point. Then multiply this vector by a random number x which lies between 0, and 1. Add this to the current data point to create the new, synthetic data point.

5.4 PCA and Normalization

In one test, we try dimensionality reduction through applying principal component analysis (PCA) to feature set 1. This is expected to improve the performance by helping to avoid overfitting. PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components[12]. Before applying PCA, we do normalization first by removing the mean and scaling to unit variance.

6 Evaluation on Results and Discussion

All classification algorithms are evaluated with logarithmic loss test set and with accuracy on validation set. Two feature sets are both tested and the first feature set is additionally tested under another two conditions, with data balanced and with dimensionality reduction (PCA).

6.1 Naive Bayes

Naive Bayes has the advantage of simplicity and efficiency. Table 1 shows the evaluation of Naive Bayes.

Table 1. Evaluation with Naive Bayes

Condition	Logarithmic Loss	Accuracy (%)
Feature Set 1	2.882	13.1
Feature Set 2	2.920	12.1
Feature Set 1 with balanced data	3.171	26.8
Feature Set 1 with PCA	34.002	13.0

6.2 k-Nearest Neighbors

The baseline method k-Nearest Neighbors is always good to try because it doesn't need much adjustment to get reasonable performance. Table 2 is the result of appling kNN when k equals to 200. (Computer fails to carry out the program when choosing parameter k because both the training set and testing set are too large.)

6.3 Logistic Regression

Logistic regression is quite suitable for the output with distribution of probability of different classes. The final evaluation is shwon in Table 3.

Table 2. Evaluation with k-Nearest Neighbors

Condition	Logarithmic Loss	Accuracy (%)
Feature Set 1	3.340	26.0
Feature Set 2	3.756	26.4
Feature Set 1 with balanced data	3.558	37.7
Feature Set 1 with PCA	5.734	25.5

Table 3. Evaluation with Logistic Regression

Condition	Logarithmic Loss	Accuracy (%)
Feature Set 1	2.585	21.5
Feature Set 2	2.640	20.1
Feature Set 1 with balanced data	2.693	32.3
Feature Set 1 with PCA	16.988	23.1

6.4 Discussion

From three tables, it is obviously PCA hurts logarithmic loss very much and also the accuracy in some degree. We have to do normalization before applying PCA and the normalization method we choose is a linear one, which is not appropriate to the geographical data like longitude and latitude.

After balancing data, logarithmic loss decreases a bit but accuracy goes up a lot. As we look closely at the formula, we can see that logarithmic loss and accuracy are evaluating the classification results from different perspectives. Logarithmic loss is a kind of soft measurement of accuracy, which indicates the uncertainty. If some success prediction is under low prediction probabilities, logarithmic loss will show high ,i.e. bad, but accuracy increases. Therefore, balancing data does help to improve the performance of classification but the certainty gets worse.

In general, the first feature set shows a little better performance than the second one in both logarithmic loss and accuracy. Logistic regression shows the lowest logarithmic loss and kNN is the best model of accuracy. This is quite reasonable since the basic idea of logistic regression is to predeict based on the estimation of the probability. Whereas, kNN does direct prediction with regards to the k nearest neighbours, which is more accurate on evaluation of accuracy.

7 Conclusion

In San Francisco crime classification task, we applied three algorithms Naive Bayes, kNN, Logistic Regression to do the classification under two different features and four different conditions. The fianl evluation is based on logarithmic loss formula. The first feature set shows a good prediction on the test set but PCA and balancing data fails to improve the results. Logistic Regression turns

out to be the best one among these three models. The best score we get is 2.585, ranking 48% in Kaggle.

References

1. San Francisco Crime Dataset(2015). Available from: https://www.kaggle.com/c/sf-crime/data
2. San Francisco Crime Classification Evaluation. Available from: https://www.kaggle.com/c/sf-crime#evaluation
3. Competition Description on Kaggle. https://www.kaggle.com/c/sf-crime#description
4. U.M. Fayyad and R. Uthurusamy, Evolving Data Mining into Solutions for Insights. Comm. ACM, Aug.2002,pp.28-31.
5. Chen, Hsinchun, et al., Crime data mining: a general framework and some examples. Computer 37.4 (2004): 50-56.
6. Iqbal, Rizwan, et al., An experimental study of classification algorithms for crime prediction. Indian Journal of Science and Technology 6.3 (2013): 4219-4225.
7. Nath, Shyam Varan, Crime pattern detection using data mining. Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI- IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on. IEEE, 2006.
8. Wikipedia: Naive Bayes classifier, https://en.wikipedia.org/wiki/Naive_Bayes_classifier
9. Wikipedia: k-nearest neighbors algorithm, https://en.wikipedia.org/wiki/K\-nearest_neighbors_algorithm
10. Dreiseitl, Stephan, and Lucila Ohno-Machado. "Logistic regression and artificial neural network classification models: a methodology review." Journal of biomedical informatics 35.5 (2002): 352-359.
11. Wikipedia: Oversampling and undersampling in data analysis, https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis
12. Wikipedia: Principal component analysis, https://en.wikipedia.org/wiki/Principal_component_analysis