**Report**

****

University of Sydney S1 2023 COMP5318

Machine Learning and Data Mining

Compiled and Prepared by A2 Group 21

Guangbin Fu & Zicong Chen

**Abstract**

This project aims to learn further about machine learning for classification by building analytical models. To do this, we will use the sentiment140 dataset, which contains data collected from Twitter. This dataset contains 1,600,000 tweets extracted using the Twitter API. These tweets have been annotated (0 = negative, 4 = positive) and can be used to detect sentiment.

After a series of cleaning, data processing, and visualization of this data in a word cloud, we will build a model. This model's goal will be to classify positive and negative tweets in terms of sentiment correctly.

Throughout the project, we will use each result to understand the data further, extract insights and information from it and learn how to improve our model. From the types of words used in the tweets containing sentiment to the diversity of words in each case and how often these words occur, to the concept of overfitting and the importance of capturing the data when building a specific model, and finally, learning further machine learning from it.

**Table of Contents**

[**1. Project Introduction 1**](#_heading=h.8mfic3hs93hl)

[**2. Market Research 2**](#_heading=h.y6j9ysv0g1z0)

[2.1 Existing Products and Services 2](#_heading=h.mmcfzudgfxzn)

[2.2 Demand Analysis 2](#_heading=h.39a0214sk66f)

[2.2.1 Customer 2](#_heading=h.62067t4zs2tu)

[2.2.2 Delivery Company 2](#_heading=h.ug2y2a9v3l29)

[2.2.3 Government 3](#_heading=h.z5y56rbcghj)

[2.3 Market Gaps 3](#_heading=h.kmcusgxokdc1)

[2.4 Opportunities for Growth 4](#_heading=h.22mrku8o1ezb)

[**3. Competitive Advantages 5**](#_heading=)

[3.1 Competitors’ Use of ITS 5](#_heading=h.reckxzsslvc8)

[3.1.1 Zipline’s ITS 5](#_heading=h.pv6q89ox11wc)

[3.1.2 Zipline’s Limitation 6](#_heading=h.1wezfx1v8bsj)

[3.1.3 Swoop Aero’s ITS 6](#_heading=h.yi20kiyquevo)

[3.1.4 Swoop Aero’s Limitation 7](#_heading=h.7k3oroj2dmx2)

[3.2 Different ITS from Medical Drone 7](#_heading=h.43zhhhg159ib)

[3.2.1 Solar Panel Technology 7](#_heading=h.w8tbh1wmtuxc)

[3.2.2 Vertical Integration Information System 8](#_heading=h.ecspp6o5mk3m)

[3.2.3 Decentralization 8](#_heading=h.zidv2txphnpz)

[3.3 Medical Drone ITS implementation 8](#_heading=h.o6db59nz4hjg)

[3.3.1 Medical Drone’s External ITS 8](#_heading=h.vpz7fda6x9yq)

[3.3.2 Medical Drone’s Internal ITS 9](#_heading=h.oigsrkw6txoz)

[**4. Unique Value Proposition 11**](#_heading=)

[4.1 Customer Segments 11](#_heading=h.g30d4c2u1ghh)

[4.2 Minimum Viable Product 11](#_heading=h.yjjbwy6x8has)

[4.3 Value Provided to Stakeholders 12](#_heading=h.m5gzdsd9l9s1)

[4.4 Benefits and Advantages of the Product 12](#_heading=h.7aawrjxbald)

[**5. Revenue Streams 14**](#_heading=)

[5.1 Sources of Revenue 14](#_heading=h.q9kzz3q4wpw2)

[5.2 Revenue Model 14](#_heading=h.q9kw3oq21zmv)

[5.3 Cost of Alternatives 15](#_heading=h.oir78pejzsiu)

[5.4 Reasons for Customers Paying for the Product 16](#_heading=h.of24bnkjdx60)

[**6. Privacy and Trust 17**](#_heading=)

[6.1 Protect Consumer Privacy 17](#_heading=h.d9z41j46pq7w)

[6.2 How It Will Achieve Consumer Trust 17](#_heading=h.5nu9gvxuc3fc)

[6.3 Individual Reflections and Recommendations 17](#_heading=h.621twa4njizx)

[**7. Conclusion 20**](#_heading=h.x4l4n7i26rxk)

[**References 21**](#_heading=h.m8b6a9qv564e)

## 

## 1. Introduction

Medical Drone is set in the market of healthcare services, which is expected to grow rapidly in the coming years. Medical Drone disrupts the industry by offering a series of specialized services, which have high demand and social value. Medical Drone also contributes to society and the country’s economy by creating jobs, saving lives, reducing emissions, and advancing innovation.

## 2. Methodology

### 2.1 Data cleaning

In order to be able to feed the tweet text data into the classification model, we first need to clean it. This prevents such words from interfering with the model by removing emotionally irrelevant words such as usernames, special nouns, links and numbers from the text data.

We then need to tokenize the cleaned text. Tokenize is the process of dividing paragraphs into sentences and sentences into individual words. In this project we will use word\_Tokenizer; a tokenizer provided by the nltk library. The text is then stored for further processing. The text will be stored for further processing after it has been split.

### 2.2 Data processing

The cleaned text data is further processed to produce a word frequency matrix, i.e. a dictionary of all words in the document after word separation. In this project, we will use two different text feature extraction function CountVectorizer and TfidfVectorizer for this purpose. In fact TfidfVectorizer is the more reliable CountVectorizer and TfidfVectorizer is equivalent to the combined use of CountVectorizer and TfidfTransformer, so CountVectorizer will be present as a control group for increasing the data sample.

#### 2.2.1 Vectorization with CountVectorizer

CountVectorizer is a text feature extraction method that belongs to the common class of feature value calculation. For each training text, it only considers how often each word appears in that training text.

CountVectorizer converts the words in the text into a word frequency matrix and it calculates the number of occurrences of each word using the fit\_transform function.

The CountVectorizer class has many parameters and is divided into three processing steps: preprocessing, tokenizing, and n-grams generation.

The CountVectorizer converts the words in the text into a word frequency matrix using the fit\_transform function. The elements of the matrix, a[i][j], represent the frequency of word j under the text. That is, the number of occurrences of each word. The keywords of all texts can be seen by get\_feature\_names() and the results of the word frequency matrix can be seen by toarray().

#### 2.2.2 Vectorization with Term Frequency - Inverse Document

TF-IDF stands for Term Frequency Inverse Document Frequency. It is the very common algorithm that converts text into a meaningful numerical representation for adapting machine algorithms for prediction. tfidfVectorizer() is based on the tf-idf algorithm. The algorithm consists of two parts, tf and idf, which are multiplied together to obtain the tf-idf algorithm.

The tf algorithm counts the number of times a word appears in a given training text and is calculated as follows: *tf(t) = (No. of times term ‘t’ occurs in a document) / (Frequency of most common term in a document)*

The idf is a measure of how common or rare a term is in the overall literature base and is calculated as follows: *idf(t) = log e [ n / df(t) ]*

### 2.3 Data Visualizing

A word cloud is one way of visualizing the frequency of words in a text document. Essentially what it does is to generate an image of the frequently occurring words in a text file that represent the frequency or importance of each word, with the most frequent words displayed in a larger font size and the less frequent words displayed in a smaller font size. By importing WordCloud, the data can be presented in a more vivid and visual way.



### 2.4 Classification

Classification belongs to the class of supervised learning, which is an important area of research in data mining, machine learning and data science. Classification models are similar to human learning in that they learn an objective function from historical data or training sets, and then use this objective function to predict unknown properties of new data sets. The project uses four main algorithms - Random Forest, Naive Bayes, Decision Tree and Logistic Regression - to analyze the dataset.

It consists of two main steps:

Training.: Given a dataset, each sample contains a set of features and a category of information, and then the classification algorithm is called to train the classification model.

Prediction: The generated model or function is used to make a classification prediction on a new dataset (test set) and to determine its post-classification result, which is displayed in a visual plot.

The data set is divided into a disjoint training set, which is used to construct the classification model, and a test set, which is used to check how many classes of labels are correctly classified.

#### 2.4.1 Classification with Random Forest

Random Forest is a supervised learning algorithm for classification and regression problems. It can be used in areas such as data mining, computer vision, natural language processing, etc. Random forests are built on the basis of decision trees. An important feature of Random Forest is that it reduces the overfitting of decision trees due to overfitting of data, thus improving the performance of the model.

A random forest is an integrated model consisting of many decision trees. Its core idea is that when training data is fed into the model, instead of building one large decision tree from the entire training dataset, Random Forest uses different subsets and feature attributes to build multiple smaller decision trees, which are then combined into a more powerful model. By combining the results of multiple decision trees, Random Forests can enhance the model.

RandomForest has several advantages: 1. Training can be highly parallelized, which is advantageous for the training speed of large samples in the era of big data. Personally, I think this is the main advantage.

2. Since the nodes of the decision tree can be randomly selected to divide the features, the model can still be trained efficiently when the dimensionality of the sample features is very high.

3. After training, the importance of each feature to the output can be given

4. Due to the use of random sampling, the variance of the trained model is small and the generalization ability is strong.

5. The RF implementation is relatively simple compared to the Boosting family of Adaboost and GBDT.

6. It is not sensitive to the absence of some features.

Random Forest also has several significant drawbacks, including

1. The RF model is prone to overfitting in some noisy sample sets.

2. Features with more divided values tend to have a greater impact on the RF decision, thus affecting the effectiveness of the fitted model.

In this project we can use the RandomForest provided by sklearn to classify the dataset using a random forest model.

#### 2.4.2 Classification with Logistic Regression

Logistic regression is a machine learning method used to solve classification problems. It is a predictive analysis technique based on probabilistic ideas. The classification algorithm Logistic regression is used to predict the likelihood of a categorical dependent variable. Logistic regression models are similar to linear regression models, except that logistic regression uses a more complex cost function called a 'Sigmoid function' or "logistic functions" rather than linear functions.

logistic function is represented by the following formulas: *Logit(pi) = 1/(1+ exp(-pi))*

In this project we can use the LogisticRegression provided by sklearn to classify the dataset using a logistic regression model.

#### 2.4.3 Classification with Naive Bayes

The Naïve Bayes classifier is a supervised machine learning algorithm for classification tasks, such as text categorization. It is also part of a family of generative learning algorithms, meaning that it seeks to model the input distribution of a particular class or category. Unlike discriminative classifiers such as logistic regression, it does not learn which features are most important in distinguishing between different categories.

Naïve Bayes is also known as a probabilistic classifier because it is based on Bayes' theorem. This theorem allows us to 'invert' the conditional probabilities. Bayes' theorem can be expressed by the following formulas:

or 

Plain Bayes has several advantages:

1. Ordinary Bayesian models have stable classification efficiency.

2. performs well on small data sizes, can handle a wide range of classification tasks, and is suitable for incremental training, especially when the amount of data exceeds memory, and can be trained in increments.

3. Less sensitive to missing data and simpler algorithms, often used for text classification.

It also has several disadvantages:

1. The prior probability needs to be known, and the prior probability often depends on the hypothesis, and there can be many kinds of hypothesis models, so in some cases, the prediction result is not good due to the assumed prior model.

2. It is sensitive to the form of expression of the input data (discrete, continuous, minimum, etc.).

In this project we can use the MultinomialNB provided by sklearn to classify the dataset using a Multinomial Naive Bayes model.

#### 2.4.4 Classification with Decision Tree

Decision trees: are a basic classification and regression method. In classification problems, it represents the process of classifying instances based on features and can be thought of as a collection of if-then or as conditional probability distributions defined over the feature space and class space.

Decision trees usually have three steps: feature selection, decision tree generation, and decision tree pruning. Classification with decision trees: starting from the root node, a feature of the instance is tested, and according to the test result the instance is assigned to its child nodes, at which point each child node corresponds to a value taken for that feature, and so on recursively, the instance is tested and assigned until it reaches a leaf node, and finally the instance is assigned to the class of the leaf node.

Decision trees allow the problem of constructing a decision tree model based on a given training dataset so that it can correctly classify instances and select the optimal decision tree in the sense of a loss function. It is essentially the induction of a set of classification rules from a training set, or the estimation of a conditional probability model from a training dataset.

The advantages of decision trees are their low computational complexity, easy to understand output, insensitivity to missing intermediate values and the ability to handle uncorrelated feature data.

The disadvantage of a decision tree is that it can create overfitting problems.

In this project we can use the DecisionTreeClassifier provided by sklearn to classify the dataset using a Multinomial Decision Tree model.

## 3. Experimental and Discussions

This project chose a classified dataset and adopted four machine learning algorithms for the dataset: random forest, logistic regression, plain Bayesian and decision tree. The data was cleaned and processed from the native dataset, a training set and a test set were distinguished, and the accuracy, precision, recall and confusion matrix of each model was calculated by machine learning the models and visualized in a graphical presentation

### 3.1 Development Environment

Operating system: Windows 11

CPU: 3.2 GHz AMD Ryzen 7 5800H with Radeon Graphics

Memory: 16GB RAM

Graphics: NVIDIA GeForce 3070

DirectX version: 11

Storage: 4 GB of free space required

Software: PyCharm 2022.3.3/Jupyter notebook

### 3.2 Dataset

The dataset uses the Sentiment 140 dataset and contains 300,000 pieces of data. This is the Sentiment140 dataset. It contains 300,000 tweets extracted using the twitter api. These tweets have been annotated and can be used to detect sentiment.

It contains the following 6 fields:

**sentiment**: the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive)

**id**: The id of the tweet ( 2087)

**date**: the date of the tweet (Sat May 16 23:58:44 UTC 2009)

**query**: The query (lyx). If there is no query, then this value is NO\_QUERY.

**username**: the user that tweeted (robotickilldozr)

**text**: the text of the tweet (Lyx is cool)

In the project, we further reduced the sample of the dataset by taking 16,000 positive and 16,000 negative tweet each for initial model learning and comparing the results obtained with the results learned from the full sample size.

### 3.3 Experimental results

In this project, there are four samples, each with the same four models. The four samples are: a sample of 16,000 tweets based on CountVectorizer classification, a sample of 16,000 tweets based on TfidfVectorizer classification, a sample of 300,000 tweets based on CountVectorizer classification and a sample of 16,000 tweets based on TfidfVectorizer classification. based tweets, a sample of 300,000 tweets based on CountVectorizer classification and a sample of 300,000 tweets based on TfidfVectorizer classification. CountVectorizer and TfidfVectorizer were used to compare the differences between similar text feature extraction functions.

#### 3.3.1 Accuracy

Accuracy is the most common evaluation metric we use, and it is easy to understand that it is the number of samples classified correctly divided by the number of all samples, and generally speaking, the higher the correct rate, the better the classifier.

Accuracy is indeed a good and intuitive evaluation metric, which is one of two measures of measurement error. The formula for the accuracy rate is

But sometimes a high accuracy rate does not mean that an algorithm is good. An unthinking classifier that classifies the category as 0 for every test case may achieve 99% accuracy, but the classifier is unaware of the real earthquake, and the loss from this classification is huge. But a 99% accurate classifier is not what we want, because here the data is unevenly distributed and there is so little data in category 1 that a complete misclassification of category 1 could still achieve a high accuracy rate but ignore what we are looking at. Therefore, relying solely on accuracy to evaluate an algorithmic model is far from scientific and comprehensive.

The accuracy of the 4 samples was compared and according to the results obtained, the larger the sample size, the higher the accuracy of the training. 300,000 sample sizes have basically a 2%-4% increase in accuracy compared to 32,000 sample sizes. At 300,000 sample size, the TFIDF-based logistic regression had the highest accuracy rate of 77.22%. The CountVectorizer-based decision tree was the least accurate, at 70.18%.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | classification with CountVectorizer | | | | classification with Tfidf | | | |
| Random Forest | Logistic Regression | Naive Bayes | Decision Tree | Random Forest | Logistic Regression | Naive Bayes | Decision Tree |
| 32000 | **73.48%** | **73.66%** | **73.66%** | **68.75%** | **73.67%** | **74.73%** | **73.16%** | **68.66%** |
| 300000 | **75.65%** | **76.87%** | **76.03%** | **70.18%** | **76.35%** | **77.22%** | **75.56%** | **70.24%** |

#### 3.2.2 Precision

Precision is the other indicator of measurement error, which is is how close the measurements are to each other. In other words, precision is a description of random errors, a measure of statistical variability. Precision can be expressed by the following formulas:

Precision describes how many of the positive examples predicted by the binary classifier are true positive examples from the perspective of prediction results, i.e. how many of the positive examples predicted by the binary classifier are accurate.

The precision of these four samples was compared and according to the results obtained, an increase in sample size also increases the precision rate. Compared to a sample size of 32,000, the random forest and logistic regression based on a sample size of 300,000 had a 2% improvement in precision, but the difference between the plain Bayesian and decision tree based on a sample size of 300,000 was essentially within 1%. At 300,000 sample sizes, the TFIDF-based plain Bayesian had the highest precision rate of 76.93%. The TFIDF-based decision tree had the lowest precision, at 70.13%.

**Precision**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | classification with CountVectorizer | | | | classification with Tfidf | | | |
| Random Forest | Logistic Regression | Naive Bayes | Decision Tree | Random Forest | Logistic Regression | Naive Bayes | Decision Tree |
| 32000 | **74.09%** | **73.29%** | **76.27%** | **69.22%** | **74.43%** | **74.77%** | **75.97%** | **68.69%** |
| 300000 | **76.01%** | **75.66%** | **76.93%** | **70.47%** | **76.75%** | **76.25%** | **76.51%** | **70.13%** |

#### 3.2.3 Recall

Unlike Precision, Recall describes how many of the true positive examples in the test set were selected by the dichotomous classifier from the perspective of true results, i.e. how many of the true positive examples were recalled by that dichotomous classifier. Recall can be expressed by the following formulas:

A cross-sectional comparison of the recall rates for these four samples was carried out. The recall rate for logistic regression was significantly higher than the other three models, while the recall rate for decision trees was also significantly lower than the other three models. With a large sample size, based on a sample size of 300,000, both the TFIDF-based and CountVectorizer-based logistic regressions achieved a recall rate of over 79%, while the decision tree with the same conditions only had a recall rate of around 70%. The largest difference between large and small samples was achieved by plain Bayes, at 6%.

**Recall**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | classification with CountVectorizer | | | | classification with Tfidf | | | |
| Random Forest | Logistic Regression | Naive Bayes | Decision Tree | Random Forest | Logistic Regression | Naive Bayes | Decision Tree |
| 32000 | **73.01%** | **75.24%** | **69.42%** | **68.62%** | **72.89%** | **75.39%** | **68.49%** | **69.67%** |
| 300000 | **75.34%** | **79.56%** | **74.71%** | **70.0%** | **75.94%** | **79.41%** | **74.13%** | **71.05%** |

### 3.3 Running time

When training the model, it was divided into a training set based on the TFIDF text feature extraction function and a training set based on the CountVectorizer text feature extraction function. The running times of the two text feature extraction functions at 300,000 sample sizes were: 3982.7 ± 20 seconds and 4605.5 ± 20 seconds respectively; and at 32,000 sample sizes were: 100 ± 10 seconds and 108 ± 10 seconds respectively.

**Running time**

|  |  |  |
| --- | --- | --- |
|  | classification with CountVectorizer | classification with Tfidf |
| 32000 tweet | **108.02025079727173** | **100.59158134460449** |
| 300000 tweet | **4605.4628784656525** | **3982.6770946979523** |

It is clear that the data set based on the TfidfVectorizer function is more friendly to the training model, and the principle of the TfidfVectorizer function is also based on the CountVectorizer function, so the TfidfVectorizer function has a better performance for machine learning purposes than the TfidfVectorizer function has better performance than the CountVectorizer function for machine learning purposes. There are of course some non-objective factors, including the use of computer hardware by other programs, but the results of several training sessions have shown that the TfidfVectorizer function is still the superior text feature extraction function.

### 3.4 Cross-sectional comparison of data classification with CountVectorizer

图表, 条形图

描述已自动生成

### 3.5 Cross-sectional comparison of data classification with TfidfVectorizer

图表, 条形图

描述已自动生成

## 4. Conclusion and Future Work

Overall, this report covers the following aspects of the project:

* Cleaning and processing of the dataset, including interference word exclusion, text feature extraction to further enhance the validity of the dataset and help the dataset cater to the model. Datasets of different sizes are also provided for cross-sectional comparisons. The market research, which analyzes the existing products and services, the demand analysis, the market gaps, and the opportunities for growth in the industry.
* Multiple models were taken for cross-sectional comparison, using both Random Forest, Logistic Regression, Parsimonious Bayes and Decision Trees, to analyze the performance of the dataset by comparing accuracy, Precision, recall and confusion matrix for training on this dataset. The unique value proposition, which identifies the customer segments, the minimum viable product, the value provided to stakeholders, and the benefits and advantages of the product.
* The data was visualized and the cleaned and processed dataset was mapped as a word cloud. Accuracy, Precision and Recall are graphically visualized for each model for a visual cross-sectional comparison. The privacy and trust, which discusses how the project protects consumer privacy and how it achieves consumer trust.

The report concludes that the project implemented classification processes for specific datasets, pre-processing the datasets, by using different machine learning algorithms, and by specifying task metrics to evaluate and compare the performance differences of the different models. The report suggests that this project could be further improved by pre-processing the dataset in such a way that the dataset can be trained to produce better models, for example by using Gensim-trained Word2Vec and pre-trained Global Vectors (GloVe) to process the dataset. In addition to this, the code processing could be made more efficient, as the training time for the existing model on a 300,000 sample size dataset is still a little over an hour.

## References

Grand view research, *Commercial Drone Market Size, Share & Trends Analysis Report By Product (Fixed-wing, Rotary Blade, Hybrid), By Application, By End-use, By Region, And Segment Forecasts*, 2023 – 2030, April 2023

Konstantinos Dalamagkidis , Kimon P. Valavanis , Les A. Piegl, *On Integrating Unmanned Aircraft Systems into the National Airspace System*, 2012

ZipLine (2016). *Zipline - Lifesaving Deliveries by Drone*. [online] Flyzipline.com. Available at: <https://www.flyzipline.com/>

Swoop Aero. (n.d.). *Swoop Aero - Transforming access to healthcare*. [online] Available at: <https://swoop.aero/>

‌