Evaluating classification algorithms on a multi-class single feature problem

1Devharsh Trivedi
1PhD student
1Stevens Institute of Technology

Abstract - **Machine learning is studying computer algorithms that improve automatically through experience and data use and is seen as a part of artificial intelligence. Classification is a task that depends on machine learning algorithms. Classification is the process of identifying and grouping objects into categories. This paper presents Fizz Buzz as a single feature supervised multi-class classification problem and evaluates the performance of various popular classification algorithms based on this problem.**

keywords - **Machine learning, Classification, Evaluation.**

I. INTRODUCTION

Machine learning and Artificial Intelligence (AI) are the most used buzzwords today. The simple definition of AI requires the agent to "see," "hear," or "read" and take singular decisions based on it. The complex definition of AI requires the agent to replicate general human intelligence. However, the minimum intelligence already provides far higher efficiency for most business contexts than in current processes.

Machine Learning is a domain of computer science with its base in computational mathematics and statistics. The machine is fed a ton of data, and it learns the pattern in the data to make future predictions, recognize new patterns, or suggest different classes to the data. Machine Learning algorithms are of three types [1] -

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

When the machine is supervised while "learning," the training type is supervised learning. But what does supervising a machine mean? It means that we provide the device with a ton of information about a case and the outcome. The outcome is called the labeled data, while the rest of the information is used as input features. For example, we show the machine cases when customers defaulted on a loan and cases where customers did not default. Here default / not default is the outcome and hence the labeled data. At the same time, all the other characteristics like age, salary, loan amount, outstanding amount, different loan history, etc., are the input features. The labeled data supervise the machine to learn the relationships and dependencies between default and the borrower information.

There are two other categories into which we can divide supervised learning: regressions and classifications, and each have its own set of use cases and merits. Standard supervised learning algorithms include Linear regression, Naïve Bayes, Nearest Neighbours, Decision Trees, Support Vector Machines, and Neural Networks. We use the regression technique to predict the target values of continuous variables, like indicating an employee's salary. In contrast, we use the classification technique for predicting the class labels for given input data. We design the classifier model in classification, then train it using input train data, and then categorize the test data into multiple class labels present in the dataset.

The classification task with supervised learning always involves two steps [2] -

1. Training (with the assessment) – this is where we discover what features are helpful for classification by looking at many pre-classified examples.
2. Classification (with the assessment) – We look at new examples and assign them to classes based on the features we have learned during training.

As the name suggests, there is no help from the user for the computer to learn in unsupervised learning. In the lack of labeled training sets, the machine identifies patterns in the data that are not obvious to the human eye. So, unsupervised learning is beneficial to recognize patterns in data and help us make decisions. For example, if we did not know which customers defaulted on loans and fed the borrower information to the machine, it would pick out similar patterns among the different borrowers and grouped them in 3-4 buckets or clusters. Unsupervised learning is also often used for anomaly detection, like to uncover fraudulent transactions or payments. The most common use of unsupervised learning is clustering problems,

with the most talked-about algorithms being k-means and hierarchical clustering. However, other algorithms like Hidden Markov models, Self-Organizing Maps, or Gaussian Mixture models are often used.

Reinforcement learning is the closest to how we as humans learn. In this case, the algorithm or the agent learns continually from its environment by interacting with it. It gets a positive or a negative reward based on its action. Let us consider the same example of customers with bank loans. A Reinforcement Learning algorithm looks at a customer's information and classifies him/her as a high-risk customer. When the customer defaults, the algorithm gets a positive reward. If the customer does not default, the agent receives negative compensation. The prize in both cases helps the agent understand the problem and the environment better and thus helps make better decisions on our behalf. Standard algorithms include Q-Learning, Temporal Difference, and Deep Adversarial Networks. Reinforcement Learning is the hardest to execute yet in a business environment but has been commonly used for self-driving cars or the famous Alpha Go chess match trials.

Probability mass function (pmf) for discrete variables gives the probability that a discrete random variable is equal to some value. Probability density function (pdf) for continuous variables is a function whose value at any given sample in the sample space can be interpreted as providing a relative likelihood that the value of the random variable would equal that sample. [3]

Bernoulli Distribution [4]

Consider you are flipping a coin with a probability of 0.4 of turning up heads and a 0.6 of turning up tails. The most straightforward probability distribution is the Bernoulli distribution which just asks for the possibility of the coin turning up heads after one toss, that is, 0.4. The Bernoulli distribution is about the probability of a random event with two possible outcomes occurring after one trial. It's used to reason about the likelihood that a random variable will succeed, such as a coin turning up heads.

The probability mass function for the Bernoulli distribution is:

$f(x; p) = p \hat{} x * (1 - p) \hat{} (1 - x)$

Where "*" is multiplication and "^" is power, "p" is the probability of success and "x" is the number of successes desired, which can be either 1 or 0. When "x" is 1, you find the probability of success, while when it is 0, you see the likelihood of a failure. Since there are only two outcomes, if "p" is the probability of success, then the possibility of a non-success is "1-p". Notice that this function is quite silly as it is just a closed-form expression for choosing either "p" or "1-p" depending on what "x" is. The number you're interested in is raised to 1, while the number you're not interested in is raised to the power of 0, turning it into 1, and then the two results are multiplied together.

Binomial Distribution [4]

One generalization we can make upon the Bernoulli distribution is to consider multiple trials instead of just one and then ask about the probability of several successes. This is called the binomial distribution. For example, the possibility that the coin described above results in precisely two heads after tossing it three times. There are many ways how this result can come about. You can get [head, head, tail], or [head, tail, head], or [tail, head, head]. For this reason, we need to consider the number of different ways the required total can be achieved, which is found by using the combination formula.

The probability mass function for the Binomial distribution is:

$f(x; n, p) = n! / (x! * (n-x)!) * p \hat{} x * (1-p) \hat{} (n-x)$

Where "*" is multiplication and "^" is power, "p" is the probability of success, "n" is the number of trials to try out, and "x" is the number of desired successes out of the "n" trials.

Multinoulli Distribution [4]

Whereas the binomial distribution generalizes the Bernoulli distribution across the number of trials, the multinoulli distribution generalizes it across the number of outcomes, rolling dice instead of tossing a coin. The multinoulli distribution is also called the categorical distribution.

The probability mass function for the multinoulli distribution is:

$f(x_s; p_s) = p_{s1} \hat{} x_{s1} * p_{s2} \hat{} x_{s2} * \ldots * p_{sk} \hat{} x_{sk}$

where "*" is multiplication and "^" is power, "k" is the number of outcomes (6 in the case of dice, 2 for a coin), "p_s" is the list of "k" probabilities where "p_{si}" is the probability of the i^{th} outcome resulting in success, "x_s" is a list of numbers of successes where "x_{si}" is the number of successes desired for the i^{th} outcome, which can be either 1 or 0 and where there can only be precisely a single "1" in "x_s."

Multinomial Distribution [4]

Finally, the most general generalization of the Bernoulli distribution is across both the number of trials and the number of outcomes, called the multinomial distribution. To be more available, this distribution also allows specifying the number of successes desired for each product, rather than just of one product like the multinoulli distribution. This lets us calculate the probability of rolling a dice six times and getting 3 "2" s, 2 "3" s, and a "5". Since we want it to be compatible with the binomial distribution, these outcomes can include arrival in any order; that is, it doesn't matter whether you get [5,3,2,3,2,2] or [2,2,2,3,3,5]. For this reason, we need to use the combination function again.

The probability mass function for the multinomial distribution is:

$$f(x_s; n, p_s) = n! / (x_{s1}! * x_{s2}! * \dots * x_{sk}!) * p_{s1} \wedge x_{s1} * p_{s2} \wedge x_{s2} * \dots * p_{sk} \wedge x_{sk}$$

where "*" is multiplication and "^" is power, "k" is the number of outcomes (6 in the case of dice, 2 for a coin), "p_s" is the list of "k" probabilities where "p_{si}" is the probability of the i^{th} outcome resulting in success, "x_s" is a list of numbers of successes where "x_{si}" is the number of successes desired for the i^{th} outcome, the total of which must be "n."

II. CLASSIFICATION

Classification is a task that requires using machine learning algorithms that know how to assign a class label to samples from the problem domain. An easy-to-understand example is classifying emails as "spam" or "not spam." There are many diverse types of classification tasks that you may encounter in machine learning. Specialized approaches to modeling may be used for each job. The classification predictive modeling process involves assigning a class label to input examples. Classification refers to a predictive modeling problem in machine learning where a class label is predicted for a given input data example.

Examples of classification problems include:

- Given recent user behavior, classify as churn or not.
- Given a handwritten character, classify it as one of the known characters.
- Given a model, classify if it's spam or not.

Classification requires a training dataset with many examples of inputs and outputs to learn. A model will use the training dataset and calculate how to best map examples of input data to specific class labels. The training dataset must be sufficiently representative of the problem and have many examples of each class label. Class labels are often string values, e.g., "spam," "not spam," and must be mapped to numeric values before being provided to an algorithm for modeling.

This is often referred to as label encoding, where a unique integer is assigned to each class label, e.g., "spam" = 0, "no-spam" = 1. There are many diverse types of algorithms for modeling classification predictive modeling problems. There is no good theory on how to map algorithms onto problem types; instead, it is recommended that a practitioner use controlled experiments and discover which algorithm and configuration result in the best performance for a given task.

Table 1 Taxonomy of Classification

Classification	Description	Example
Categorical (Nominal)	Type of entities into categories.	That thing is a dog.
Ordinal	Classification of entities in some ordered relationship.	You are stronger than him.
Adjectival (Predicative)	Classification based on some quality of an entity.	That car is fast.
Cardinal	Classification based on a numerical value.	He is six feet tall.

Classification predictive modeling algorithms are evaluated based on results. Accuracy is a popular metric used to assess the performance of a model based on the predicted class labels. It is not perfect but is a good starting point for many classification tasks. Instead of class labels, some jobs may require predicting a probability of class membership for each example. This provides additional uncertainty in the prediction that an application or user can then interpret. A popular diagnostic for evaluating predicted probabilities is the ROC Curve. [5]

When we solve a classification problem with only two class labels, it becomes easy for us to filter the data, apply any classification algorithm, train the model with filtered data, and predict the outcomes. However, when we have more than two class instances in input train data, it might get complex to analyze the data, train the model, and predict relatively accurate results. To handle these multiple class instances, we use Multi-class Classification. Multi-class classification allows us to categorize the test data into multiple class labels present in trained data as a model prediction.

There are four main types of classification tasks that you may encounter; they are:

- Binary Classification
- Multi-Class Classification
- Multi-Label Classification
- Imbalanced Classification

Binary Classification [6]

Binary classification refers to predicting one of two classes, and Multi-class Classification involves predicting one of more than two classes. We wish to classify data into one of two binary groups for Binary Classification - these are usually represented as 0's and 1's in our data. Examples include:

- Email spam detection (spam or not)
- Churn prediction (churn or not)
- Conversion prediction (buy or not)

Typically, binary classification tasks involve the normal state and abnormal state. For example, "not spam" is the normal state, and "spam" is abnormal. Another example is "cancer not detected," which is the normal state of a task that involves a medical test, and "cancer detected" is the abnormal state. The class for the normal state is assigned the class label 0, and the course with the abnormal condition is given the class label 1. Generally, a binary classification task uses a model that predicts a Bernoulli probability distribution for each example. The Bernoulli distribution is a discrete probability distribution covering a case where an event will have a binary outcome as either a 0 or 1. For classification, the model predicts a probability of an example belonging to class 1 or the abnormal state. Popular algorithms that can be used for Binary Classification include:

- Naïve Bayes
- Logistic Regression
- k-Nearest Neighbors
- Decision Trees
- Support Vector Machine

Some algorithms are designed explicitly for Binary Classification and do not natively support more than two classes; examples include Logistic Regression and Support Vector Machines.

Multi-class Classification [6]

For Multi-class classification, we wish to group an outcome into one of multiple (more than two) groups. While Binary Classification alone is beneficial, there are times when we would like to predict data that has more than two classes. Many of the same algorithms can be used with slight modifications. Additionally, it is common to split data into training and test sets. This means we use a particular portion of the data to fit the model (the training set) and save the remaining amount from evaluating the fitted model's predictive accuracy (the test set). There is no written rule to follow when deciding on a split proportion, though you would want 70% for the training set and 30% for the test set.

Examples include:

- Plant species classification
- Optical character recognition
- Face classification

The multi-class classification does not have the notion of normal and abnormal outcomes. Instead, examples are classified as belonging to one among a range of classes. The number of class labels may be vast on some problems. For example, a model may predict a photo as belonging to one among tens of thousands of faces in a face recognition system. Problems such as text translation models that involve predicting a sequence of words may be considered a particular type of Multi-class Classification. Each word in the line of words to be expected involves a multi-class classification. The vocabulary size defines the number of classes and could be hundreds of thousands of words in length. It is common to model a multi-class classification task with a model that predicts a Multinoulli probability distribution for each example. The Multinoulli distribution is a discrete probability distribution covering a case where an event will have a definite outcome. This means that the model predicts the probability of an example belonging to each class label. Many algorithms used for Binary Classification can be used for Multi-class Classification. Popular algorithms that can be used for Multi-class Classification include:

- Decision Trees
- Naive Bayes
- K-nearest Neighbors
- Random Forest
- Gradient Boosting

Algorithms that are designed for Binary Classification can be adapted for use for multi-class problems. This involves using a strategy of fitting multiple binary classification models for each class vs. all other types (called one-vs-rest) or one model for each pair of classes (called one-vs-one).

- One-vs-Rest: Fit one binary classification model for each class vs. all other courses.
- One-vs-One: Fit one binary classification model for each pair of classes.

Binary classification algorithms that can use these strategies for Multi-class Classification include:

- Support Vector Machine

- Logistic Regression

Multi-label Classification

Multi-label classification refers to those classification tasks with two or more class labels, where one or more class labels may be predicted for each example. It involves predicting one or more classes for each example, and Imbalanced Classification refers to classification tasks where the distribution of samples across the classes is not equal. Consider the example of photo classification, where a given photo may have multiple objects in the scene, and a model may predict the presence of numerous known things in the picture, such as "bicycle," "an apple," "person," etc.

A single class label is predicted for each example in Binary Classification and Multi-class Classification. It is common to model multi-label classification tasks with a model that predicts multiple outputs, with each production being expected as a Bernoulli probability distribution. This is a model that makes numerous binary classification predictions for each example. Multi-label Classification cannot directly use classification algorithms used for Binary or Multi-class Classification. Specialized versions of standard classification algorithms can be used, so-called multi-label versions of the algorithms, including:

- Multi-label Random Forests
- Multi-label Gradient Boosting
- Multi-label Decision Trees

Imbalanced Classification

Classification tasks where the number of examples in each class is unequally distributed is referred to as Imbalanced Classification, which is typically binary classification tasks where most instances in the training dataset belong to the regular class. A minority of models belong to the abnormal class. Examples include:

- Fraud detection
- Medical diagnostic tests
- Outlier detection

These problems are modeled as binary classification tasks, although they may require specialized techniques. Specialized techniques may change the composition of samples in the training dataset by under-sampling the majority class or oversampling the minority class. Examples include:

- SMOTE Oversampling
- Random Undersampling

While fitting the model on the training dataset, specialized modeling algorithms pay more attention to the minority class, such as cost-sensitive machine learning algorithms. Examples include:

- Cost-sensitive Decision Trees
- Cost-sensitive Support Vector Machines
- Cost-sensitive Logistic Regression

Alternative performance metrics may be required as the classification accuracy may be misleading. Examples include:

- Precision
- Recall
- F-Measure

In statistics, three different analysis techniques exist. These are –

- Univariate analysis
- Bivariate analysis
- Multivariate analysis

The selection of the data analysis technique depends on the number of variables, types of data, and focus of the statistical inquiry.

Univariate Analysis [7][8]

Univariate analysis is where the data being analyzed contains only one variable. Since it is a single variable, it does not deal with causes or relationships. The primary purpose of the univariate analysis is to describe the data and find patterns. Think of the variable as a category that data falls into. An example of a variable in the univariate analysis might be "age." Another might be "height." The univariate analysis would not look at these two variables simultaneously, nor would it look at the relationship between them.

Patterns found in univariate data include examining mean, median, range, variance, quartiles, maximum, minimum, and standard deviation. Additionally, some ways to display univariate data include frequency distribution tables, bar charts, histograms, frequency polygons, and pie charts.

Univariate analysis is the most basic form of the statistical data analysis technique. A Univariate analysis technique is used when the data contains only one variable and does not deal with causes or effect relationships. For instance, in a classroom survey, the researcher may be looking to count the number of boys and girls. In this instance, the data would reflect the number, i.e., a single variable and its quantity. The key objective of Univariate analysis is to describe the data to find patterns within the data. This is done by examining the mean, median, mode, dispersion, variance, range, standard deviation, etc.

Univariate analysis is conducted in several ways, which are primarily descriptive.

- Frequency Distribution Tables
- Histograms
- Frequency Polygons
- Pie Charts
- Bar Charts

Bivariate Analysis [7][8]

Bivariate analysis is used to find a relationship between two different variables. Plotting one variable against another on a Cartesian plane and creating a scatter plot can give you a picture of what the data is trying to tell you. If the information fits a line or curve, there is a relationship between the two variables— caloric intake versus weight.

Bivariate analysis is more analytical than Univariate analysis. When the data set contains two variables, and the goal is to compare the two data sets, then Bivariate analysis is the right technique. For example, in a classroom survey, the researcher may analyze the ratio of students who scored above 85%, corresponding to their genders. There are two variables, gender = X (independent variable) and result = Y (dependent variable).

1. Correlation coefficients

Correlations are a statistical association technique where the strength of the relationship between two variables is observed. It is rated on a scale of –1 to 1, where 1 is a perfect direct correlation, –1 is a perfect inverse correlation, and 0 is no correlation.

2. Regression analysis

Regression analysis is used for estimating the relationships between two different variables. It includes techniques for modeling and analyzing several variables when the focus is on the relationship between a dependent and one or more independent variables. It helps to understand how the value of the dependent variable changes when any one of the independent variables is changed. Regression analysis is used for advanced data modeling purposes like prediction and forecasting. There is a range of different regression techniques used depending on the nature of the variable and the type of analysis sought by the research.

For example –

- Linear regression
- Simple regression
- Polynomial regression
- General linear model
- Discrete choice
- Binomial regression
- Binary regression
- Logistic regression

Multivariate Analysis [7][8]

Multivariate analysis is the analysis of three or more variables. There are many ways to perform multivariate analysis depending on your goals. Some of these methods include:

- Additive Tree
- Canonical Correlation Analysis
- Cluster Analysis
- Correspondence Analysis / Multiple Correspondence Analysis
- Factor Analysis
- Generalized Procrustean Analysis
- MANOVA
- Multidimensional Scaling
- Multiple Regression Analysis
- Partial Least Square Regression
- Principal Component Analysis / Regression / PARAFAC
- Redundancy Analysis

Multivariate analysis is a more complex statistical analysis technique used when there are more than two variables in the data set. E.g., A doctor has collected data on cholesterol, blood pressure, and weight. She also collected data on the subjects' eating habits. To investigate the relationship between the three measures of health and eating habits, a multivariate analysis would be required to understand the relationship of each variable with the other.

Commonly used multivariate analysis technique include –

- Factor Analysis
- Cluster Analysis
- Variance Analysis
- Discriminant Analysis
- Multidimensional Scaling
- Principal Component Analysis
- Redundancy Analysis

III. ALGORITHMS

This section provides details about the most popular supervised classification machine learning algorithms. [9]

Logistic Regression

Logistic regression is a Generalized Linear Model (GLM) that uses a logistic function to model a binary variable based on any independent variables. Logistic regression is like linear regression but is used when the dependent variable is not a number but something else (e.g., a "yes/no" response). It's called regression but performs classification based on the regression and classifies the dependent variable into either of the classes. Logistic regression is used for the prediction of output, which is binary, as stated above. For example, if a credit card company builds a model to decide whether to issue a credit card to a customer, it will model whether they will "default" or "not default" on their card.

Support Vector Machines

Support Vector Machines (SVMs) algorithm is more flexible. SVM can do linear classification but can also use other non-linear basis functions. Support vector is used for both regression and classification. It is based on the concept of decision planes that define decision boundaries. A decision plane (hyperplane) separates between a set of objects having different class memberships. It performs classification by finding the hyperplane that maximizes the margin between the two classes with the help of support vectors. The learning of the hyperplane in SVM is done by transforming the problem using some linear algebra. For higher-dimensional data, other kernels are used.

Random Forests

A decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The result is a tree with decision and leaf nodes. It follows the Iterative Dichotomiser 3 (ID3) algorithm structure for determining the split. Random Forests are a type of ensemble learning method which fits multiple Decision Trees on subsets of the data. An ensemble model is a team of models. Technically, ensemble models comprise several supervised learning models that are individually trained, and the results merged in various ways to achieve the final prediction. This result has higher predictive power than the results of any of its constituting learning algorithms independently.

Neural Networks

Neural networks involve fitting many hidden layers to represent neurons connected with synaptic activation functions. Multi-layer perceptron (MLP) is a supervised learning algorithm that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset, where m is the number of dimensions for input and o is the number of sizes for output. Given a set of features $X = x_1, x_2, \ldots, x_m$, and a target y, it can learn a non-linear function approximator for either classification or regression. It is different from logistic regression in that between the input and the output layer; there can be one or more non-linear layers, called hidden layers. These use a very simplified model of the brain to model and predict data. MLP trains using stochastic gradient descent, adam, or L-BFGS. [10]

IV. EXPERIMENT

Fizz buzz is a group word game for children to teach them about division. Players take turns to count incrementally, replacing any number divisible by three with the word "fizz" and any number divisible by five with the word "buzz." Players sit in a circle. The player designated to go first says the number "1", and the players then count upwards in turn. However, any number divisible by three is replaced by the word fizz, and any number divisible by five by the word buzz. Numbers divisible by 15 become fizz buzz. A player who hesitates or makes a mistake is eliminated from the game. [11][12]

For example, a typical round of fizz buzz would start as follows:

1, 2, fizz, 4, buzz, fizz, 7, 8, fizz, buzz, 11, fizz, 13, 14, fizz buzz, 16, 17, fizz, 19, buzz, fizz, 22, 23, fizz, buzz, 26, fizz, 28, 29, fizz buzz, 31, 32, fizz, 34, buzz, fizz, ...

In some versions of the game, other divisibility rules such as seven can be used instead. Another practice that may complicate the game is where numbers containing a digit also trigger the corresponding control (for instance, 52 would use the same rule for a number divisible by 5). Fizz buzz has been used as an interview screening device for computer programmers. Writing a program to output the first 100 fizz buzz numbers is a trivial problem. Still, its value in coding interviews is to analyze fundamental coding habits that may indicate overall coding ingenuity. Fizz buzz is a straightforward programming task used in software developer job interviews to determine whether the job candidate can write code. It was invented by Imran Ghory and popularized by Jeff Atwood.

Here is a description of the task:
Write a program that prints the numbers from 1 to 100. But for multiples of three, print "fizz" instead of the number, and the multiples of five print "buzz." For numbers that are multiples of both three and five, print "fizz buzz."

For this paper, the fizz buzz problem was converted as a multi-class classification problem, and the accuracy of several supervised classification algorithms was compared. The input was a single-feature dataset. For data N, numbers were fed starting from 0 to N-1. Data were classified into four classes: fizz, buzz, fizz buzz, and blank. E.G., For data N = 21, numbers 0,1,2,4,7,8,11,13,14,16,17, and 19 are assigned to class blank, 3,6,9,12, and 18 are assigned to class fizz, 5,10 and 20 are assigned to class buzz, and 15 assigned to class fizz buzz.

V. RESULTS

The experiment was conducted on a 13-inch 2019 model MacBook Pro with a 2.4 GHz Quad-Core Intel Core i5 processor and 8 GB 2133 MHz LPDDR3 RAM with Intel Iris Plus Graphics 655 1536 MB graphics.

Table 2 Accuracy of Classification Algorithms

Data	LR	SVM	RF	MLP
113	0.5517	0.5517	0.1034	0.5517
112	0.4643	0.4643	0.1786	0.4643
111	0.5714	0.5714	0.2143	0.5714
109	0.4643	0.4643	0.1429	0.4643
108	0.5185	0.5185	0.1481	0.5185
104	0.5385	0.5385	0.1154	0.5385
103	0.5385	0.5385	0.1154	0.3462
102	0.6154	0.6154	0.1538	0.6154
101	0.5385	0.5385	0.1154	0.5385
99	0.6400	0.6400	0.1200	0.6400
98	0.5600	0.5600	0.1600	0.5600
50	0.6923	0.6923	0.2308	0.6923
30	0.6250	0.6250	0.2500	0.6250
28	0.7143	0.7143	0.4286	0.8571
27	0.7143	0.7143	0.5714	0.7143
25	0.2857	0.2857	0.1429	0.0000
21	0.5000	0.6667	0.1667	0.0000
15	0.2500	0.5000	0.2500	0.2500
10	0.3333	0.0000	0.6667	0.0000

As shown in the experiment results, Logistic Regression and Support Vector Machine had similar performance, and they performed well compared to Random Forest and Multilayer Perceptron. Logistic regression had a mean accuracy of 0.5324 and a median of 0.5385 across all data points. Support Vector Machine had a mean accuracy of 0.5368 and a median of 0.5517 across all data points. For the case with N=10 data points, Logistic Regression scored 0.3333 accuracies while Support Vector Machine scored 0. For N=15 and N=21, Support Vector Machine performed better than Logistic Regression. For all the other cases, two had the same accuracy. Random Forest scored a mean accuracy of 0.2250 and median of 0.1600, while Multi-layer Perceptron secured a mean accuracy of 0.4709 and a median accuracy of 0.5385, the same as Logistic Regression.

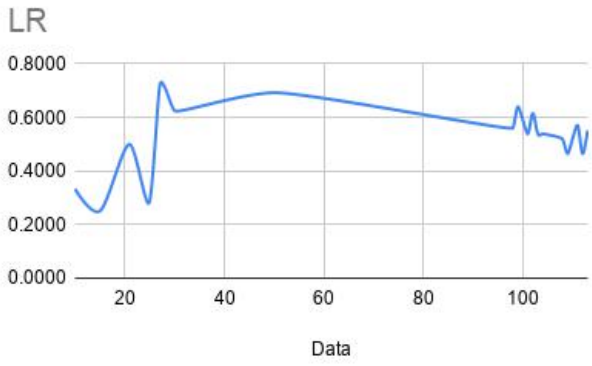

Figure 1 Logistic Regression Accuracy

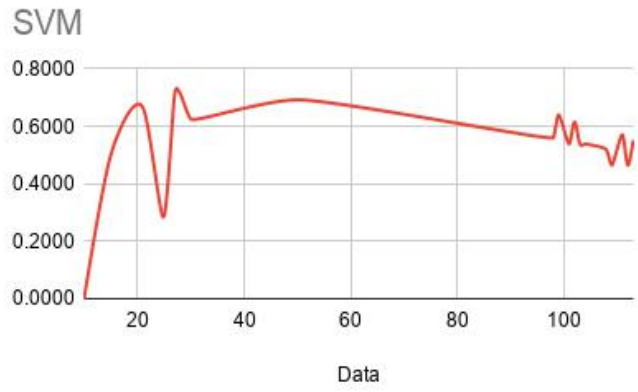

Figure 2 Support Vector Machine Accuracy

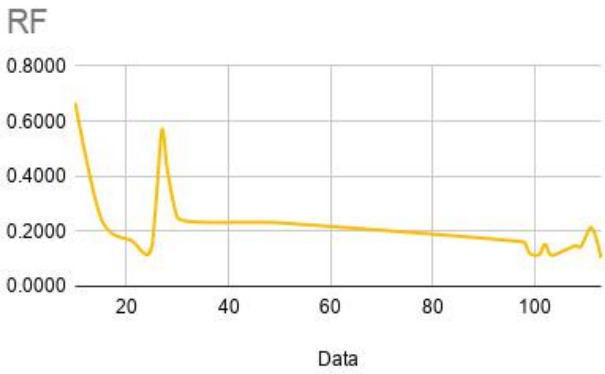

Figure 3 Random Forest Accuracy

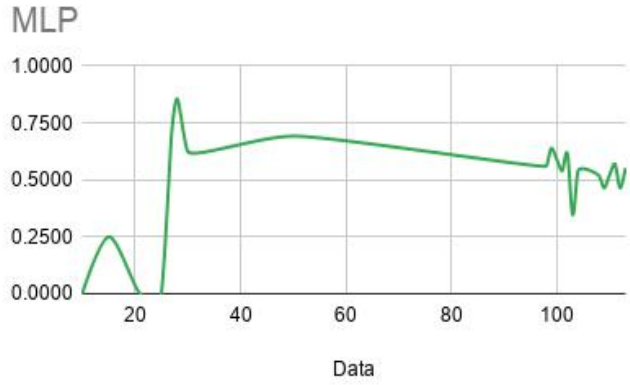

Figure 4 Multi-layer Perceptron Accuracy

VI. Conclusion

This paper explains the classification problem in the machine learning domain with some examples. The taxonomy of classification is presented in detail. Fizz Buzz is converted to a single feature multi-class supervised learning classification problem, and the performance of some popular algorithms is compared. We conclude that Random Forest performed poorly, and more data points did not help increase the accuracy. While Multi-layer Perceptron had zero accuracies for N<27 in most cases, it can give the best results if fine-tuned. Otherwise, Logistic Regression and Support Vector Machine performs well in most cases out of the box. The code for this project is available upon request.

References

[1] https://pioneerlabs.io/insights/the-three-types-of-machine-learning-algorithms/
[2] http://www.improvedoutcomes.com/docs/WebSiteDocs/Classification_and_Prediction/SLAM/An_Introduction_to_Classification.htm
[3] https://en.wikipedia.org/wiki/Probability_density_function
[4] https://geekyisawesome.blogspot.com/2016/12/bernoulli-vs-binomial-vs-multinoulli-vs.html
[5] https://machinelearningmastery.com/types-of-classification-in-machine-learning/
[6] https://stackabuse.com/classification-in-python-with-scikit-learn-and-pandas/
[7] https://hotcubator.com.au/research/what-is-univariate-bivariate-and-multivariate-analysis/
[8] https://www.modernanalyst.com/Careers/InterviewQuestions/tabid/128/ID/4904/Describe-the-difference-between-univariate-bivariate-and-multivariate-analysis.aspx
[9] https://builtin.com/data-science/supervised-machine-learning-classification
[10] https://scikit-learn.org/stable/modules/neural_networks_supervised.html
[11] https://victorqi.gitbooks.io/swift-algorithm/content/fizz_buzz.html
[12] https://en.wikipedia.org/wiki/Fizz_buzz