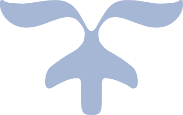


BIG DATA FINAL PROJECT

Mobile price classification and clustering



# Introduction

In 2011 a real-time data streaming technology capable of handling trillions of events per day named Kafka has been created. Initially conceived as a messaging queue, Kafka is based on an abstraction of a distributed commit log and become the industry standard for working with data in motion. [2] In this study, we created a producer from Kafka to allow data movment to uploaded to a dataset in real-time.

By using the special features of Spark library, we created two consumers that will read the real-time data from a topic where the producer uploaded the data. Each one of the consumers will read the real-time data and fulfill one specific task – compare two models of clustering or classification. At each iteration of the consumers, the models will be updated to the new accumulated dataset.

The dataset used for this project is ‘Mobile Price Classification’ available on Kaggle [1]. The dataset contains 2000 samples and 21 features, including the target variable. The features represent different aspects of mobile phones, such as the number of cores of the processor, the total energy a battery can store or the weight of the phone. All the feature values are numeric and don’t required encoding. The target represents the price range of the different phones in a categorical fashion, with 0 being low cost, 1 being medium cost, 2 being high cost and 3 being very high cost.

In this study we use the categorical target to our advantage, using classification methods as well as clustering methods to see which of them forms a better model to represent the data distribution.

There are 3 Hypotheses to this study:

1. Classification vs clustering –

Regarding the fact that a classification model has access to the ground-truth labels and a clustering algorithm does not, we expect the classification models to be more accurate than the clustering models.

1. K-means vs Bisecting K-means –

Bisecting K-means is a clustering algorithm in which the data is placed in one cluster, and that cluster is iteratively divided into two clusters until sum of squared distances between each data point and its nearest cluster centroid is minimized.  This method can increase the parallelism of the fitting process but is less likely to converge to a global minimum. On this basis, we expect the K-means algorithm to outperform the Bisecting K-means algorithm, while still having a longer computation time.

1. Decision tree vs Random Forest –

Regarding the fact that a Random Forest model fits multiple trees while the decision tree fits only one, we expect the decision tree to have a lower accuracy then the Random Forest, while still having a faster computation time.

# Method

To simulate the flow of real time data into a classification/clustering system, a Kafka producer has been created. This producer publishes 100 samples every 5 seconds to a topic named “Data”. In addition, 2 consumers have been created that read from the topic “Data”. The first consumer fits 2 clustering models, K-means and Bisecting K-means, which will be compared both on their accuracy and on their computation time. The second consumer fits 2 classification models, a decision tree and a Random Forest, which will be compared both on the same criteria. The pipeline of built for this project can be seen in Figure 1.

Graphical user interface, diagram

Description automatically generated

*Figure 1 - Pipeline from Kafka server to clustering and classification models.*

In the first consumer we fit the clustering models by hiding the target variable and fitting K-means and Bisecting K-means. K has been chosen to be 4 because the original dataset has 4 classes. In the case of clustering, we can calculate the error directly over the whole dataset using the labels we set aside at the beginning of the process.

In the second consumer we fit the classification models by first splitting the data to 70:30 between the training set and the test set respectivly, and then we fit a decision tree and a Random Forest model on the training set. In this case, the error is calculated only on the test set.

In both consumers all features are standardized to ensure good quality clustering and classification in the different models. The standardization can also lead to greater efficiency of the algorithms.

In this study we test not only the accuracy of the models but also their computation time. We use Spark dataframes to store the data, thus utilizing the parallel computing characteristics of all 16 cores of our allocated machine. In correspondence to hypotheses 2 and 3, we expect Bisecting K-means to have a faster computing time than K-means and the decision tree to have a faster computing time than the Random Forest.

# Results

## Clustering Results

The first consumer has been created to compare two models of clustering: K-means and Bisecting K-means. To evaluate both, a comparison of the calculation time and accuracy has been applied. Because the producer publishes 100 samples every 5 seconds, the consumer has 20 iterations, in each iteration the consumer reads the data from the topic, parallelizes it using a Spark dataframe, and then using Pyspark machine learning library, fits both models to the data collected so far. The accuracy and computation time are calculated for every iteration and a comparison can be seen in Figure 2 and Figure 3.

These figures show that in contrast to our second Hypothesis, Bisecting K-means has a better accuracy than K-means, and also K-means turned out to have a shorter computation time.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

## Classification Results

The second consumer has been created to compare two models of classification: the decision tree and a Random Forest model. To evaluate both, a comparison of the calculation time and accuracy has been applied over the test set. Because the producer publishes 100 samples every 5 seconds, the consumer has 20 iterations, in each iteration the consumer reads the data from the topic, parallelizes it using a Spark dataframe, and then using Pyspark machine learning library, fits both models to the data collected so far. The accuracy and computation time are calculated for every iteration and a comparison can be seen in Figure 4 and Figure 5.

These figures show that as suggested in the third Hypothesis, the Random Forest model does take more time to fit, although the accuracy of both models seem to be very much alike at most iterations.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

# Discussion

In this project, a Kafka server has been created with a topic “Data”, a producer that simulates real-time data flow and 2 consumers that cluster and classify the data using Spark to parallelize the fitting process. The dataset chosen for this pipeline is the ‘Mobile Price Classification’ dataset, available on Kaggle.

The first consumer compared the K-means and Bisecting K-means algorithms and showed that surprisingly, the Bisecting K-means gave higher accuracy values than the original K-means. Several reasons why K-means did not perform as expected and Bisecting K-means had a higher accuracy can be:

1. Dataset - The performance of clustering algorithms heavily depends on the characteristics of the data. K-means assumes that clusters are spherical in shape and have equal variances, while Bisecting K-means can handle more complex shapes and variances.
2. Initialization - K-means is sensitive to the initialization of the centroids, and if the initial centroids are not chosen well, it can lead to poor convergence and lower accuracy. Bisecting K-means overcomes this limitation by iteratively dividing the data into smaller and smaller clusters, thus reducing the impact of poor initial centroids.

However, the computation time of the Bisecting K-means is higher than that of the K-means algorithm. Bisecting K-means starts with all the data points in one cluster and splits it into two smaller clusters in each iteration, repeating the process until the desired number of clusters is reached. This iterative process requires multiple runs of K-means and can take longer than a single run of K-means. From these results we can reject the second Hypothesis of this study. Also, due to the low accuracy received in both models, we can infer that the clustering is not the ideal algorithm for this specific dataset.

The second consumer compared a Decision Tree and Random Forest model and showed that in general, as the number of data points increases, the algorithm may be able to find the underlying patterns in the data more efficiently, resulting in greater accuracy as well as faster convergence time. This theory also explains an unexpected phenomenon that is the decrease in computation time while the number of samples increased. In addition, Spark ML is designed to take advantage of distributed computing and parallel processing, which can also contribute to faster training times as the data size increases.

As for the comparison between the 2 classification algorithms, the training time of the Random Forest Algorithm is higher than the training time of the Decision Tree Algorithm. The reason for this is that in general, Random Forest algorithm is slower to train than a single Decision Tree because it creates multiple trees and aggregates their results. In general, the training time of a Random Forest can be higher due to the number of trees, ensemble process and features sampling. In respect to the accuracy, the Random Forest model shows similar results to the decision tree, this might be because of several reasons:

1. If the decision tree is not overfitting the data and is giving a good representation of the data, then even a random forest, which is an ensemble of decision trees, may not be able to perform better.
2. The performance of a random forest depends on the number of trees in the forest. In this study no hyper-parameter optimization has been done on the Random Forest model because it was not the main purpose of the study.

Considering this, we can say that the second hypothesis is partly correct – the Random Forest model does not outperform the decision tree, but it does take more time to fit, even with parallel computation.

In conclusion, considering the results of all models, we can say that the first hypothesis has been proved to be correct, with the classification models peaking at an accuracy of 85%, while the clustering algorithms peak at less than 60% accuracy.

# References

[1] <https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification?select=train.csv>

[2] Kreps, J., Narkhede, N., & Rao, J. (2011, June). Kafka: A distributed messaging system for log processing. In *Proceedings of the NetDB* (Vol. 11, No. 2011, pp. 1-7).‏