A Big Data Approach for Customer Behavior Analysis in Telecommunication Industry

Hong-Phuc Vo[1,2], Khoa-Gia-Cat Nguyen[1,2], Kim-Loc Nguyen[1,2], and Thanh-Van Le[1,2](✉)

[1] Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh, Vietnam
ltvan@hcmut.edu.vn

[2] Vietnam National University Ho Chi Minh City (VNU-HCM), Ho Chi Minh, Vietnam

Abstract. The evolution of telecommunications has led to a profound transformation in the realm of communication, revolutionizing how this industry mines customer behavior for their business outcomes. The analysis of user historical activities promoted paramount importance in driving strategic decision-making to enhance customer experiences and recommend ways to attract customers more effectively. While the demand is growing, some telecom data analytics either use small datasets or provide a high abstract level of analysis result. When the number of customers increases significantly, it becomes impractical to customize service for each customer under the same approach. This paper provides a comprehensive examination of the challenges, needs, and solutions associated with the analysis of user data within the telecom domain. We focus on three key user data analysis problems: user clustering, user classification, and revenue prediction derived from user insights. With Florus - our proposed big data framework, we have carried out the telecom customer behavior analysis with a large dataset. The experiment result demonstrates the promising performance and its potential for long-term use.

Keywords: Customer behavior analysis · big data framework · regression · clustering

1 Introduction

The contemporary telecommunications sector is currently experiencing an unprecedented increase in user-generated data, which shows no signs of abating. This rapid expansion of telecom data introduces a substantial degree of intricacy, presenting a spectrum of challenges and prospects for stakeholders within the industry. The sheer volume and complexity of the data being generated are fundamentally reshaping the way businesses operate and engage with their customers. Every interaction represented by voice calls, text messages, internet usage, or location tracking, contributes to the vast reservoir of data that is being amassed. What adds further complexity is the diverse range of data formats, spanning from structured, semi-structured to unstructured, further augmenting the intricate nature of the telecom data landscape.

N. Thai-Nghe et al. (Eds.): ISDS 2023, CCIS 1950, pp. 91–105, 2024.
https://doi.org/10.1007/978-981-99-7666-9_8

When seeking a business enhancement strategy, the corporations found that customer behavior analysis is the key to maintaining and developing their relationship with customers, who bring income. Undoubtedly, the e-commerce industry has a long-time experience in mining customer spending. In [11], the customer spending score predictions can be inferred through the regressions method. This work utilizes of customer spending budget and increases sale value. However, the sample of the research is only 200 customers, a small number compared to the large scale of the telecom company. Despite the simplicity, this technique may not be applicable when the volume and complexity are significantly different.

Considering the customer behavior analysis in the aforementioned domain, to win a larger market share than competitors, telecom research is mainly focused on the churn proportion, these analysis [3,9,15] expose reasons and alert the provider about the tendency of the user to terminate service, require action on services to maintain the loyalty in this sector [2], not all the potential clients are alike [11]. Relative to the focused study sample, based on telecom customers' usage, scientists can predict their demographic information [4]. In this paper, we will concentrate directly on the revenue of users from 3 primary services - service subscription, phone call, and internet usage - that most telecom company offer.

Another related research about telecom user revenue is Forecast and Analyze the Telecom Income based on ARIMA Model [14]. This research was conducted on the revenue of the provinces in China, in the format of a time series of Yen income for the company. This study represents the total revenue in general, but can not provide the granular forecast or specify the marketing strategy for each consumer due to the high abstraction level of data aggregation.

Under the large scale of dataset, the telecom industry demands a detailed, low bias, low error solution for every customer. Not only about providing insights to enhance the customer experience but also providing a general conclusion for a higher level of customer management on strategy motivation. The study about telecom customer behavior has not yet focused on their spending, which mainly contributes to the revenue of the company, for every of millions of users. In this paper, we propose a method for predicting future charged amounts for each user based on their usages. To tackle the imbalance, the regression analysis will be conducted on groups of segments, where the segment is a cluster of users. In application, to feature the decision support system, we provided a classification model, with its outcome will invoke the corresponding revenue prediction model.

As the processing data amount raising, the traditional implementation for machine learning suffers the bottleneck or exceeds the upper bound of the memory. In our experiment, the algorithm of scikit-learn can only handle a table with 2 million rows × 4 columns, much smaller than the size of this industry data. Three solutions to keep working with this approach are sampling the dataset to the lower number of records, or scaling up the system, or scaling out the system. While sampling risks the value of unmanipulated information, scaling up faces the infeasibility of physical computing resources. Therefore, scaling out the system and using the big data framework to construct a stronger cluster seems the best approach. Specifically, we will set up an Apache Spark cluster and use its algorithm library API to support our analysis process. The implementation of these setups is part of the Florus - a framework for handling large datasets.

The paper is organized in the following way. Section 2 describes the background of the Apache Spark framework. Section 3 proposes a pipeline for analysis of telecom user behavior based on big data techniques. Section 4 represents the obtained model and chosen parameters for each stage. In particular, Sect. 4.3 again evaluates the application process and testing on the unknown dataset. Finally, Sect. 5 summarizes all the conducted results and describes future work.

2 Apache Spark Framework

2.1 Architecture

Apache Spark is a multi-language engine for executing data engineering, data science, and machine learning on a single-node machine or a cluster. It is an open-source architecture that implements the MapReduce model along with Hadoop Distributed File System (HDFS). While HDFS stores the result of each MapReduce phase in the disk, Spark maintains it in memory when possible [10].

A spark application architecture is also designed to follow the Master/Slave concept where the master is called driver and the slave is called executor. When an application is started, the driver first creates a Spark Context, which acts like a gateway to access all functionalities of Spark, to connect to its cluster manager such as Yarn, Meros or Kubernetes depending on how a Spark cluster is deployed. Then, it will request the cluster manager resources and allocate some executors in the worker node [10,12].

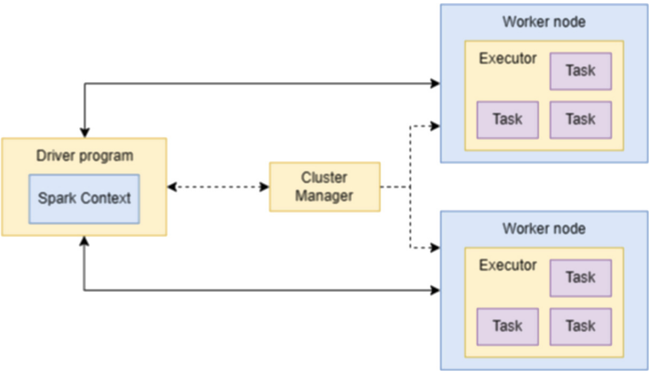


Fig. 1. Architecture of a Spark application

Figure 1 above shows the architecture of a Spark application where the dashed line shows the process to request resources, allocate the executor, and the solid lines show the process of passing data through the driver and executor as tasks.

2.2 Spark MLlib

MLlib, Spark's open-source distributed machine learning library, provides efficient functionality for a wide range of learning settings and includes several underlying statistical, optimization, and linear algebra primitives. Shipped with Spark, MLlib supports several languages and provides a high-level API that leverages Spark's rich ecosystem to simplify the development of end-to-end machine learning pipelines [8]. The MLLib provides 3 types of model algorithm API, namely: Classification, Regression, and Clustering.

Cluster Analysis. Cluster analysis or simply clustering is the process of partitioning a set of data objects (or observations) into subsets [5]. Each of these subsets contains similar objects, whose similarities are different from the other groups. A K-means model or any other clustering analysis can be evaluated by the most common metrics, which are the Silhouette score and Inertia score:

- **Inertia**: Inertia measures how internally coherent clusters are.

  $$Inertia = \sum_{k=1}^{K} \sum_{x_i \in C_k} distance(x_i, c_k)^2 \tag{1}$$

  Where:
  - C_k: is the K^{th} cluster
  - x_i: is the i^{th} point in the C_k
  - c_k: is the centroid of C_k
- **Silhouette**: Silhouette score is used to evaluate the quality of clusters created using clustering algorithms in terms of how well samples are clustered with other samples that are similar to each other.

  $$s(x_i) = \frac{b_i - a_i}{max(a_i, b_i)} \tag{2}$$

  Where:
  - a_i is the mean distance from x_i to others point in cluster x_i.
  - b_i is the mean distance from x_i to all points in the nearest cluster of x_i

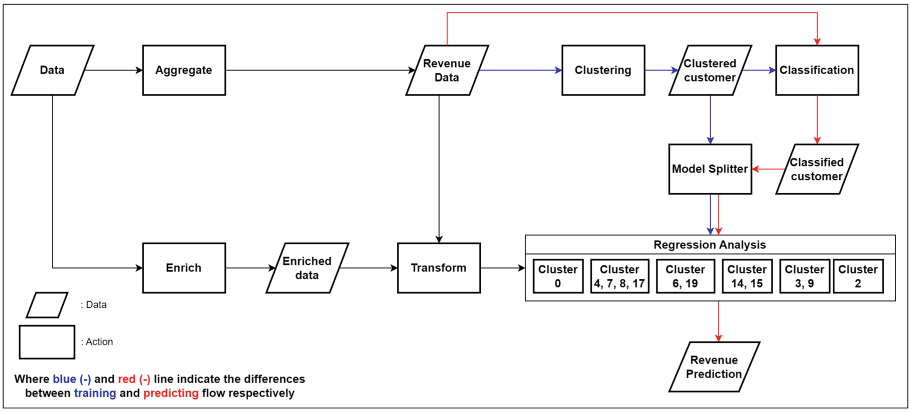
Classification. Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts [7]. The prerequisites include training data and test data. They all require a label feature to predict when missing this label. Three experimented algorithms in this paper are Logistic Classification, Gradient-Boosted Trees, and Decision Tree, and they are evaluated using accuracy, precision, and recall.

Regression. Regression analysis is a statistical methodology that is most often used for numeric prediction, hence the two terms tend to be used synonymously [6]. This method visualizes the distribution trends of data. The specified Linear regression or Gradient-boosted tree regression APIs can be evaluated for the application in regression.

3 Proposed Approach

User behavior analysis is vital for the management of the telecommunication industry, being considered one of the most important factors contributing to business success. The *revenue* represents the charged amounts on phone calls and internet usage of customers, two main services of this industry. This study aims to predict the spending of each user based on their recorded activities in the nearest 2 months period. Even though high benefits would come for customer relationship management, inaccurate analysis can be adverse to the marketing and sale strategy. Especially when businesses witness the number potential of customers with high revenue contributions much lower than the others. Consequently, the recommendations for this group do not satisfy these customers, not even enhance the profits but cost money and effort to lose customers.

We propose serial steps from customer clustering to revenue prediction so that the enhancement could be suitable for both the high revenue and high quantity customer groups. The approach will have 2 flows: the training process for building appropriate models and predicting process in the Decision Support System (DSS). Figure 2 illustrates the stages inside and the differences between them.



Where blue (-) and red (-) line indicate the differences between training and predicting flow respectively

Fig. 2. Data flow of the proposed approach

In our experiment, we use over 550 million anonymized and daily aggregated records from telecom customer usages. The main focus of this dataset is package subscriptions, phone calls, internet charged fees, and top-up amounts in 3 months. There is no demographics data included to reveal user identities, as well as any raw format of user activities. Due to the large volume, we will conduct the process with our Florus framework for all relevant tasks in collecting and analyzing data.

3.1 Florus - A Framework for Handling Large Dataset

Florus is a Lakehouse architecture framework designed for handling large datasets. We design this system to support end users to ingest data from multiple sources, then process and visualize the stored data into graphics. For machine learning demand, Florus provides the interface to read data, train models, and save the result in our environment.

The system includes the user interface, a set of micro-services, and the infrastructure. This framework can apply to any dataset and does not require redesigning the architecture of Fig. 3. In general, there are 5 layers, each holds a specific function in the system.

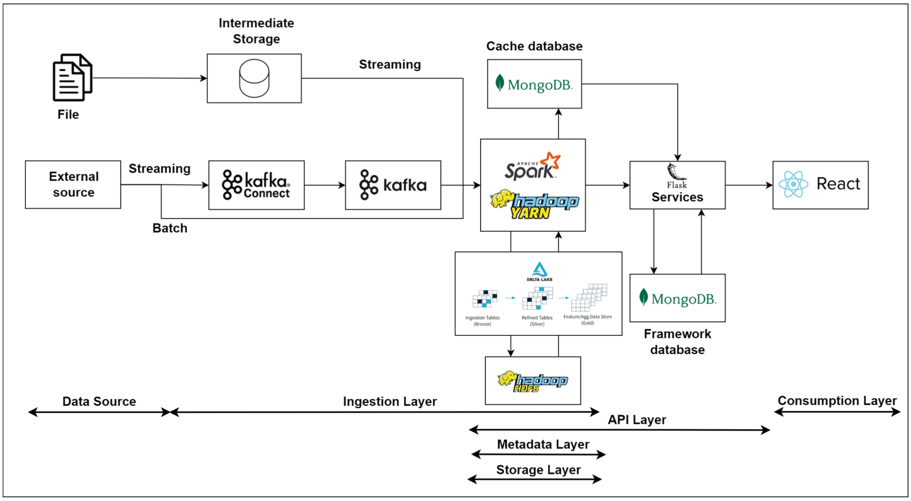


Fig. 3. Architecture of Florus framework

- Ingestion layer: This first layer retrieves data from different data sources and transport it to the Storage Layer. This layer can unify both batch and streaming processing.
  - Batch: Scheduling an event to read from the database.
  - Streaming: Connecting to the source database to stream newly appended data by Kafka and Kafka Connector. Later, data will be proceeded to streaming into HDFS by Spark Structured Streaming. Users can also upload files to our intermediate storage before starting streaming files.
- Storage layer: The Storage layer is made up of two distinct components that belong to two separate purposes:
  - Using HDFS to store data received from the Ingestion layer. The Medallion architecture supports data management in this layer by 3 levels of cleanliness: Bronze (raw data), Silver (validated, enriched), and Gold (refined, aggregated data).

  - Using MongoDB to store data about the framework, including user information, project details, metadata of data sources, tables, and other framework related metadata.
- Metadata layer: This layer provides Metadata of the HDFS and furnishes management features:
  - ACID Transaction: Delta Lake framework ensures consistency as multiple parties concurrently read or write data.
  - Caching: MongoDB is used for caching to store the content of the analyzed result to speed up the query on HDFS.
- API layer: The core of this layer is the Flask component, which hosts the API endpoints. The processing can associate Spark and SparkML to read/write on HDFS or to operate the machine learning model.
- Consumption layer: ReactJS is used to provide the user interface for their interaction with the system such as setting data sources, processing data files, training machine learning models, ... In addition, Apache ECharts will help the user represent visualization for aggregated result.

3.2 Cluster Settings

Besides, the implementation under the Spark application in the above framework also requires setting up the infrastructure and configuring resources. In our experiments, we conducted the work on a Spark cluster with **one master node and three slave nodes** by the instruction of [1]. Each computing node has 32GB RAM, 1TB storage and is running the Ubuntu 22.04.1 LTS version of the operating system. Table 1 lists the important parameters in our Spark nodes.

Table 1. Configuration for Spark application

Configuration	Key	Value
Number of executors	spark.executor.instances	14
Number of cores for drivers	spark.driver.cores	3
Number of cores for each executors	spark.executor.cores	3
Size of memory used for driver	spark.driver.memory	3 GB
Size of memory used for each executors	spark.executor.memory	3 GB
Number of partitions in RDDs returned by transformations	spark.default.parallelism	126
Number of partitions to use when shuffling data for join or aggregation	spark.sql.shuffle.partition	126

3.3 Customer Behavior Analysis

For tackling our objectives, we have carried out multiple steps to develop a set of models and their correspondences to give a throughout the analysis. In order to enhance the accuracy and handle the analysis on a large number of users, the following tasks will be integrated into the pipeline:

1. Preparation:
   The user's revenue is first aggregated from data-using, calling using log by month and by week. In addition, the user's revenue is affected by their behavior, which is usually related to their subscriptions and many other relevant uses. Each plan allows users to purchase a fixed amount of non-charge services within a specific period. Customer segments represent a set of users with similar charge fees, counted calls, and internet services,... By generating multiple features, the dataset then be enriched. Additionally, the enriched dataset also requires transformation before entering the final stage by splitting into 3 subsets as follows:
   - Train set: Use for training model
   - Validate set: Use for model selection and tunning hyperparameter
   - Test set: Use for testing each step of the pipeline and the whole pipeline
2. Clustering:
   Clustering user datasets can bring numerous benefits to businesses. Firstly, grouping customers with similar purchasing behaviors helps businesses gain a better understanding of the needs and preferences of each customer segment. Secondly, this analysis allows businesses to save time and cost in customer management, and focus on high-potential customer segments to increase sales. In this approach, the users are then clustered into groups to their different charged usage. However, the cluster size does not correlate to the revenue contribution, which is the cause of the bias in the simple approach. The purpose of this stage is to find the segments and their spending distributions, which later will be the vital metrics for reducing imbalance.
3. Classification:
   In general, user classification will help businesses understand their customer base better and, as a result, develop appropriate business strategies, marketing tactics, and customer care approaches tailored to different user groups. Regarding our suggested process, the classification model uses the outcome of the clustering stage to train and aims to classify the users into equivalent segments based on their summary usages.
4. Regression:
   Due to the difference in size and revenue of these above clusters, we extract them into multi-independent models to avoid their interference in the process of other clusters. The classification model acts as the gateway for the regression stage, so the splitting rule should take into account the miss-classified behavior of this model. Based on the confusion matrix, clusters are grouped if they have a miss classified into one another in the group of more than 25% of the predicted number. Finally, the bias is ceased by the combination of multiple regression to increase the accuracy.

In applications, the classification module helps us to determine which revenue model should be used. In case the whole proposed pipeline has not been retrained, their historical usage is recorded but has never been clustered, this pipeline is still able to achieve the revenue prediction output.

We proposed an additional scheme to evaluate when applying the model in industrial-scale scope:

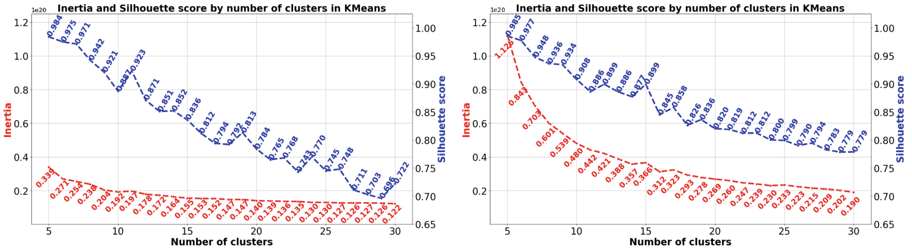
1. Preparation: similar to the preparation stage of the training process.
2. Classification: the input will be 1 feature lower than the clustering training set used to predict the segments of customers (from clustering). The prediction from this model will be enriched with other features and passed to the next stage.
3. Regression: using the label classified by the previous stage, the splitter will invoke the regression model trained for this segment to generate revenue prediction.

4 Experiment

While working on the models, the entire dataset will be used for the clustering. However, in the later stages, only 80% of the dataset is included in the training data and the other 10% will serve as testing data for both regression and classification models. Finally, we will perform the evaluation stage for the pipeline application on the remaining dataset.

4.1 User Clustering

Dataset Pre-processing and Hyperparameter Tuning. In the user clustering tasks, we used the *AVG3MONTHS* attribute as a feature for elements within the clusters. This attribute represents the average revenue from users in 3 consecutive months. For 2 types of time windows, we clustered customers by the aggregation of their expenses. Both of them will have a total of 26 KMeans models with cluster numbers ranging from 5 to 30 to cluster the data. The Inertia and Silhouette score based on the cluster numbers are as follows:

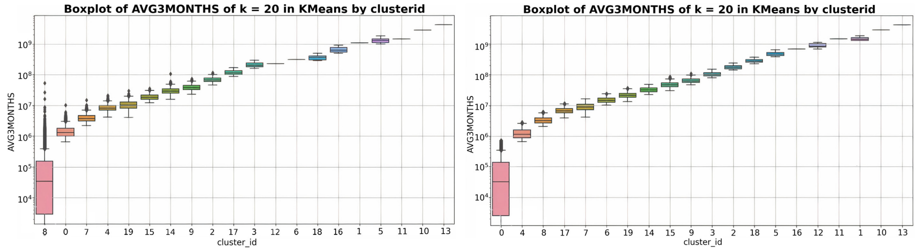


(a)

(b)

Fig. 4. Inertia and Silhouette scores by cluster numbers. (a) weekly revenue dataset. (b) monthly revenue dataset.

For each dataset, the Inertia will be measured on different cardinality of features, which lead to the dissimilar value range. The number of cluster then be selected by narrowing down the range to reach more stable changes in Inertia and Silhouette score. After selecting k for each dataset, the distribution of *AVG3MONTHS* for each cluster then be visualized in Fig. 5.



(a) (b)

Fig. 5. Average revenue in 3 months from each user across the clusters derived from clustering model (a) trained by weekly dataset. (b) trained by monthly dataset.

Result Analysis. As in Fig. 4, the overall trend of Inertia and Silhouette score are decrease. For the weekly dataset, from $k = 12$, the improvement between two consecutive Inertia measures become less significant but Silhouette score still fluctuating, and hold the average of 0.779, which is approximate the value for 20 clusters. For the monthly dataset, the changes in both metrics marginally slower when k reach 20. Therefore, choosing ***k=20*** is deemed to be an appropriate selection and these labels are added into our set of features.

Regarding clustering based on weekly revenue data, particularly with 20 clusters, based on the boxplots presented in the Fig. 5a, it can be observed that some clusters like clusters 8, 0, 7, etc., are suffering the outliers who overlap the other clusters' boxplot. Compared to using weekly revenue data from users, using monthly revenue data yields better clustering results in clusters with smaller range of outliers and higher Silhouette scores. Therefore, this result will be chosen as the baseline for user classification and revenue prediction tasks.

However, the other models require a sample with enough quantity to train and test, some clusters also contain outliers, which adversely affect the overall results.

- Remove the cluster whose cluster size is lower than 100.
- Remove the outliers of revenue by using interquartile range.

4.2 User Classification

Dataset. In the user classification problem, we will use the revenue of the two last months in 2022 to classify users into clusters. Their segment labels are inferred from the result of previous stage with $k = 20$ based on the clustering evaluation metrics.

Result Analysis. Using three algorithms: Softmax Regression, Decision Tree, and Gradient-Boosted Trees, the summarize classification results of the validation set using the key metrics of Accuracy, Precision, and Recall for each label in the table.

Table 2. Evaluation of classification algorithms

	Softmax regression		Decision Tree		Gradient-Boosted Trees	
	Accuracy: 0.9587		Accuracy: 0.9569		Accuracy: 0.9609	
Label	Precision	Recall	Precision	Recall	Precision	Recall
0	0.9882	0.9941	0.9863	0.9965	0.9862	0.9979
2	–	–	–	–	0.8788	0.9063
3	0.3279	0.3175	–	–	0.7344	0.7460
4	0.9037	0.9139	0.9119	0.9019	0.9177	0.9059
6	0.7651	0.8157	0.7244	0.8119	0.7687	0.8666
7	0.4147	0.1301	0.3355	0.1242	0.4942	0.1341
8	0.8004	0.7508	0.7758	0.7366	0.8119	0.7398
9	–	–	0.3924	0.7879	0.8188	0.7879
14	0.5995	0.5635	0.5000	0.7842	0.7825	0.8801
15	0.2678	0.1034	–	–	0.6383	0.4138
17	0.7648	0.8275	0.7545	0.7949	0.7670	0.8680
19	0.3685	0.2795	–	–	0.5990	0.3315

Overall, the accuracy of the algorithms is quite high. However, when considering the Precision and Recall metrics for the classes, it can be observed that class 0 is well-classified, while the majority of the other classes are not. This can be attributed to the imbalanced nature of the data, where class 0 dominates the dataset. In the boxplot Fig. 5b, the data in class 7 has a wide distribution and overlaps with the neighboring classes (class 17 and 6). As a result, this class also exhibits poorer classification performance compared to other classes with a high number of instances.

Based on the evaluation, it can be observed that for the classes with fewer instances (class 2, 3, 19, 15), the Softmax Regression and Decision Tree algorithms yield poor results (even failing to recognize any elements in some classes, presented by "-" in the Table 2), whereas the Gradient Boosted Tree algorithm provides better evaluation measures.

The reason why the ensembled algorithms (Gradient-Boosted Trees) perform the best is the unsuitable for handling imbalanced dataset of the others. Their poor predictive accuracy over the minority class come from the tendency to favor the samples from the majority class. On the other hand, Gradient Boosted Trees belong to the Boosting group of algorithms, which are known to handle imbalanced data well [13]. As a result, these algorithms produce more promising results compared to the others.

4.3 Regression Analysis

Baseline Result. The most elementary approach for this problem is using a single regression model for all the users. This baseline model is only one single regression without enhancement stages conducted. The result achieved by Gradient-Boosted Trees Regression recorded by Table 3.

Table 3. Performance of the simple regression approach

R^2	**MAE**
67.81%	500,318

The overall performance is acceptable, however, this result suffers from the imbalance ratio of income contribution and cluster size. In contrast to the biggest size of cluster 0 (81.69% customers), they only hold 8.59% of all cluster revenue. Since this characteristic can lead to a model designed for only the biggest cluster, the user segmentation result will be manipulated in the Model splitting step.

Model Splitting. Having chosen the group of 20 clusters by evaluating Silhouette and Inertia value, some of them then be removed or excluded outliers before applying the grouping logic. The final combination of these clusters will be group A includes cluster 0, group B (4, 7, 8, 17), group C (6, 19), group D (14, 15), group E (3, 9), and group F of cluster 2, ordered by cluster size (Table 4).

Table 4. Cluster size and average revenue in 3 months of each cluster group

Cluster	Total cluster size	AVG3MONTHS
0	5,159,001	79,084
4, 7, 8, 17	1,123,860	2,320,169
6, 19	24,670	16,766,213
14, 15	5,601	35,120,355
3, 9	1,933	73,260,771
2	277	172,481,584
Total	**6,315,342**	**789,318**

Data Enrichment and Preprocessing. The enriched dataset table contains 79 features about 6.9 million individual users. At the size of rows and columns, the dataset has 1 column of identify key for customers, 3 features of the revenue (with one as the dependent variable), 20 features about phone call history, 20 features of subscription, and 5 internet services columns. All data source is anonymized to ensure the customer's privacy.

Result Analysis. Altogether the regression model is the combination of 6 separated Gradient-boosted trees regression models. Cluster 0 requires the highest attributes for training and also transformation steps added. Table 5 presents the performance of each model on the test dataset (accounting for 10% total).

Table 5. Evaluation of regression models on test set

Cluster	MAE
0	70,010
4, 7, 8, 17	1,183,768
6, 19	6,515,053
14, 15	12,807,727
3, 9	23,188,716
2	34,743,690
Overall	**397,781**

According to this table, we can state that the MAE value of most models is lower than their average revenue. Even though MAE correlate to the revenue, the ratio between them is much lower for the high-spending clusters namely (3,9) or 2. In other words, this model can perform well on both the largest customer segmentation as well as the others.

The overall result is the evaluation of the predicted value of atom models with their actual value. While that may indicate a huge difference in the metrics, this is related to the varying of each model's sample size. This combined model performance also yields the MAE at **397,781**. This means the metric is reduced by about 20% compared to the base model.

Evaluation of Pipeline in Telecom's DSS. Using the trained models and 10% unused of the dataset, we test this pipeline through the classification and the invoked regression process.

The overall MAE in the application is about **422,502 VND**, approximate to the MAE of the regression evaluation result. Even though these metrics do not gain perform as well as the training stage, this prediction pipeline still achieves an improvement of **15.56%** in the MAE of the base model.

5 Conclusion

In this paper, we proposed a big data based model for analyzing telecom users' behavior. The research contribution is to enable telecom companies to understand their customers' actions, giving a closer look at what benefits they will get and may support a future recommendation system of appropriate service to each user. As the dataset is millions of people and records, the solution is implemented in a Spark cluster and takes advantage of this framework's machine learning library.

In the first analysis stage, we clustered all the people and found 20 customer segments. The clustering result was then used to build the training and testing dataset of the classification tasks to predict a cluster label for a new customer for recognizing the potential one. Acting as the classifier for the regression model, a Gradient-boosted Tree Classification is chosen to identify the segment from the customer's monthly charge.

Next, the regression analysis model set includes 6 atom models, divided by the size and their miss classified rates. They all require the enriched data of user usage summaries to pass through the preprocessing stage of feature selection and data normalization. These models will be activated by the predicted value from the classification model.

Based on the evaluation result, this research can feature in the Telecom Decision Support System with the purpose to promote the business operation. To further increase the accuracy, we recommend implementing an additional layer of churn prediction before entering the regression model. In order to improve model performance, we continue collecting the dataset over a longer period of time.

Acknowledgement. The authors acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

References

1. Alapati, S.R.: Expert Hadoop administration managing, tuning, and securing spark, YARN, and HDFS. Addison-Wesley Professional (2016)
2. Chen, C.M.: Use cases and challenges in telecom big data analytics. APSIPA Trans. Signal Inf. Process. **5**, e19 (2016). https://doi.org/10.1017/ATSIP.2016.20
3. Dalvi, P.K., Khandge, S.K., Deomore, A., Bankar, A., Kanade, V.A.: Analysis of customer churn prediction in telecom industry using decision trees and logistic regression. In: 2016 Symposium on Colossal Data Analysis and Networking (CDAN), pp. 1–4 (2016). https://doi.org/10.1109/CDAN.2016.7570883
4. Felbo, B., Sundsøy, P., Pentland, A., Jørgensen, S.L., Montjoye, Y.A.: Modeling the temporal nature of human behavior for demographics prediction. Lect. Notes Comput. Sci. **10536**, 140–152 (2017). https://doi.org/10.1007/978-3-319-71273-4_12
5. Han, J., Kamber, M., Pei, J.: 10 - cluster analysis: basic concepts and methods. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining. The Morgan Kaufmann Series in Data Management Systems, 3rd edn, pp. 443–495. Morgan Kaufmann, Boston (2012). https://doi.org/10.1016/B978-0-12-381479-1.00010-1

6. Han, J., Kamber, M., Pei, J.: 6 - mining frequent patterns, associations, and correlations: basic concepts and methods. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining. The Morgan Kaufmann Series in Data Management Systems, 3rd edn, pp. 243–278. Morgan Kaufmann, Boston (2012). https://doi.org/10.1016/B978-0-12-381479-1.00006-X
7. Han, J., Kamber, M., Pei, J.: 8 - classification: basic concepts. In: Han, J., Kamber, M., Pei, J. (eds.) Data Mining. The Morgan Kaufmann Series in Data Management Systems, 3rd edn, pp. 327–391. Morgan Kaufmann, Boston (2012). https://doi.org/10.1016/B978-0-12-381479-1.00008-3
8. Meng, X., et al.: Mllib: machine learning in apache spark (2015)
9. Olle, G.: A hybrid churn prediction model in mobile telecommunication industry. Int. J. e-Educ. e-Bus. e-Manag. e-Learn. (2014). https://doi.org/10.7763/IJEEEE.2014.V4.302
10. Shaikh, E., Mohiuddin, I., Alufaisan, Y., Nahvi, I.: Apache spark: a big data processing engine. In: 2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM), pp. 1–6 (2019). https://doi.org/10.1109/MENACOMM46666.2019.8988541
11. Sharma, P., Chakraborty, A., Sanyal, J.: Machine learning based prediction of customer spending score. In: 2019 Global Conference for Advancement in Technology (GCAT), pp. 1–4 (2019). https://doi.org/10.1109/GCAT47503.2019.8978374
12. Sleeman, W.C., IV., Krawczyk, B.: Multi-class imbalanced big data classification on spark. Knowl. Based Syst. **212**, 106598 (2021). https://doi.org/10.1016/j.knosys.2020.106598
13. Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., Asadpour, M.: Boosting methods for multi-class imbalanced data classification: an experimental review. J. Big Data **7**(1), 1–47 (2020). https://doi.org/10.1186/s40537-020-00349-y
14. Wang, M., Wang, Y., Wang, X., Wei, Z.: Forecast and analyze the telecom income based on arima model. Open Cybernet. Syst. J. **9** (2015)
15. Win, N.A.S., Thwin, M.M.S.: Comparative Study of Big Data Predictive Analytics Frameworks. Ph.D. thesis, MERAL Portal (2017)