Module**:CS6502Applied Big Data & Visualisation**

**Professor: Sarmad Ali**

Project**: Amazon Electronics Reviews Classification Using Apache Spark**

|  |  |
| --- | --- |
| Student Id | Student Names |
| [24102369](mailto:24102369@studentmail.ul.ie) | Ashwin Karthikeyan |
| 23179309 | Anurag Raju Kalamkar |
| 24170763 | Narayanadass Bala Krishna Karra |
| 24195693 | Nitish Kanchakam |
| 24129763 | Kishore Kajendran |

****

**Introduction**

Big data production continues in excessive quantities through digital processes that stem from e-commerce platforms along with social media networks and financial transactions and Internet-of-Things devices. [“Katal, Wazid, & Goudar, 2013”.](https://www.stat.purdue.edu/~doerge/BIOINFORM.D/SPRING16/KatalWazidGoudar_2013.pdf) Coping with Big Data's enormous datasets and executing proper analysis to obtain useful insights presents major hurdles to existing management systems. The big volumes of data require data processing systems beyond their scope because they lack necessary scalability alongside insufficient speed and inflexibility. [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing)

Apache Spark is now considered the top choice for big data processing because it both runs memory-based calculations and distributes processing workload across server clusters. [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing) Apache Spark connects effortlessly with SparkML and SparkSQL which enables it to work as a complete solution for large-scale end-to-end data science operations. The project delivers hands-on experience in dealing with big data through machine learning and Apache Spark operations on the free Databricks Community Edition platform.

The Amazon Electronics Reviews Dataset serves as the chosen dataset for this project to analyze extensive customer-provided reviews of Amazon electronic products. [“McAuley, Targett, Shi, & van den Hengel, 2015”.](https://arxiv.org/abs/1506.04757) The dataset contains structured information about rating scores with unstructured data found within written review texts making it suitable to evaluate numerical and text-based data processing approaches. The main targets of this project consist of:

1. The project uses SparkSQL along with Spark DataFrames to speed up the **data loading process and cleaning operations**.
2. The data requires exploratory analysis through **Exploratory Data Analysis (EDA)** to identify patterns by conducting both descriptive and visual data evaluation.
3. The review texts needed to undergo **machine learning processing** for the creation of relevant features.
4. The project builds a classification model through **Machine Learning implementation** followed by conducting evaluations to predict review sentiments and rating categories.
5. The process involves two parts including **performance enhancement and result interpretation** to obtain actionable findings.

Through the fulfillment of these tasks this project establishes Spark's complete ability to effectively manage and analyse significant datasets at each stage of processing. The implementation models how to manage typical limitations found in massive data science operations by using distributed processing strategies.

A diagram of a process

AI-generated content may be incorrect.

**Figure 1** End-to-End Data Processing and Machine Learning Pipeline for Amazon Electronics Reviews. The pipeline includes dataset ingestion into Spark, data cleaning, exploratory data analysis (EDA), feature engineering of review text and length, and final model training using Random Forest Classifier.

**Image Source:** [**napkin.ie**](https://napkin.ie/)

1. **Data Ingestion & Cleaning (Using Spark SQL and Spark Data Frames)**

The project began with importing the Amazon Electronics Reviews Dataset from its JSON format [“McAuley, Targett, Shi, & van den Hengel, 2015”.](https://arxiv.org/abs/1506.04757) The JSON data set entered Apache Spark through read.json(). Using this particular method let the schema automatically detect itself during the process which reduced the need for manual schema specification. The data fields included reviewerID, asin, reviewerName, helpful, reviewText, overall, summary, unixReviewTime, and reviewTime that form an ideal structure for text and number analysis. A verification of field type recognition by Spark was achieved using the printSchema() command. [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing).

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 2:** Schema and sample records from the Amazon Electronics Reviews dataset loaded into Spark Data Frame.  
**Image Source: Databricks Community Edition\***

The dataset’s composition received preliminary investigation before data ingestion. Five records were shown through the show(5) function allowing users to see the structure along with the content of individual reviews. The count() function delivered information regarding the available record number. The diagnostic exploration validated that the dataset possessed an ample size which would provide sufficient data for both statistical testing and machine learning modeling. The initial number of rows is 1689188 with 9 corresponding columns.

All data cleaning procedures were applied to the dataset before starting analytical and modeling work to achieve improved data consistency. The data cleaning process started with null value treatment; records containing missing information in essential columns reviewText and overall were eliminated through dropna() method application [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). . This check resulted in reassessing the dataset to maintain only whole records. The dropDuplicates() function served to detect and remove duplicate records in the sequence. The accurate elimination of duplicate records during data preparation became essential because their presence would introduce bias and jeopardize both training and evaluation results for the model.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 3: Cleaned sample data from Amazon Electronics Reviews.**  
**Image Source: Databricks Community Edition\***

The project required no outlier removal because the data evaluation showed that specific outlier removal steps were unnecessary. The review texts and ratings components represented the key features of interest because they demonstrated solid resistance to numerical outlier issues. The data types Spark applied to stored fields during ingestion proved suitable without additional change ["Han, Kamber, & Pei, 2011”](https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf)

The primary functional aspect of machine learning was designated to the overall rating field and the reviewText field was chosen as the main input characteristic. The thorough preparation served as an essential base for upcoming stages of exploratory data analysis together with model development.

1. **Exploratory Data Analysis (EDA)**

Exploratory data analysis (EDA) of the Amazon Electronics Reviews data began after the data cleaning stage. Major relationships in the data require identification through EDA operations before modeling steps begin.

Basic descriptive statistics were calculated first by concentrating analysis on the overall rating field. The distribution of customer ratings became accessible through the implementation of SparkSQL aggregation functions [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). The groupBy approach together with the count function showed that electronic product customers mostly provided 4 or 5 rating scores demonstrating positive attitudes.

**Code Executed:**

python

CopyEdit

df.groupBy('overall').count().orderBy('overall').show()

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 4: Distribution of review counts across different overall ratings.**  
**Image Source: Databricks Community Edition\***

In addition to analysing ratings, the characteristics of the review texts were explored. A new feature, reviewTextLength, was created by calculating the character length of each review using the length() function [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). This helped understand the variability in review detail across the dataset.

**Code Executed:**

python

CopyEdit

from pyspark.sql.functions import length

df = df.withColumn('reviewTextLength', length(df['reviewText']))

Summary statistics for review text length were then generated using the describe() method, showing significant variation, with some reviews being concise and others considerably longer.

**Code Executed:**

python

CopyEdit

df.select('reviewTextLength').describe().show()

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 5: Summary statistics of review text length after data cleaning.**  
**Image Source: Databricks Community Edition\***

Finally, a histogram was plotted to visualize the distribution of review text lengths, confirming that most reviews were relatively short, with a small number of exceptionally lengthy entries.

**Code Executed:**python  
CopyEdit  
display(df.select('reviewTextLength'))

A graph of a graph

AI-generated content may be incorrect.

**Figure 6: Review length distribution histogram.**

**Image Source: Databricks Community Edition\***

Overall, the EDA highlighted the prevalence of positive customer sentiment and considerable diversity in review content length. These insights guided the feature engineering and modelling phases that followed.

1. **Feature Analysis**

Preparation of the machine learning model required feature analysis as a vital step to process the dataset["Han, Kamber, & Pei, 2011”](https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf). The system aimed to expand the data by selecting important characteristics while preparing it for predictive model usage. The textual review content combined with overall rating information required both numerical and text-based analysis since these variables contained most primary information.

A new feature called review\_length was created through characterization of review contents by counting each review's character count. The review length measurement system allowed researchers to quantify the amount of information customers included in their review. Prolonged review content was considered as a possible source of extensive information related to customer satisfaction levels [“Mukaka, 2012”](https://pubmed.ncbi.nlm.nih.gov/23638278/).

**Code Executed: (Generated by Chatgpt)**

python

CopyEdit

from pyspark.sql.functions import length

df = df.withColumn('review\_length', length(df['reviewText']))

Following the feature creation, an investigation was conducted to assess whether the review length had any relationship with the overall rating provided by the customer. Using a correlation function, it was found that the correlation coefficient between review length and rating was approximately -0.0959, indicating a very weak negative correlation. In other words, review length did not have a significant influence on the rating score [“Mukaka, 2012”](https://pubmed.ncbi.nlm.nih.gov/23638278/).

**Code Executed:**

python

CopyEdit

df.select('overall', 'review\_length').corr('overall', 'review\_length')

Standardization methods were applied to the numerical features before model preparation. Standardization creates features that occupy zero values and have equal quantity of variances because such transformations collaborate best with models whose performance depends on input data ranges. ["Han, Kamber, & Pei, 2011”](https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf). During vectorization the textual data received processing from SparkML’s feature engineering tools [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). Text reviews became numerical vectors which the machine learning algorithms could manage through the text\_features column.

Together the feature transformations achieved complete optimization of numerical and textual data elements for the upcoming machine learning process.

**A screenshot of a computer

AI-generated content may be incorrect.**

**Figure 7: Summary of feature engineering steps.**

**Image Source: Databricks Community Edition\***

**4. Machine Learning Model Implementation**

The extracted features allowed supervision-driven machine learning to perform classified review assessments through engineered features. The Random Forest Classifier was used because it demonstrated critical features of robustness alongside feature management for large high-dimensional datasets along with compatibility for numeric and textual features [“Breiman, 2001”](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf). Random Forests determine predictive accuracy alongside overfitting control through their decision tree combination method thus creating an effective model for complex real-world datasets like textual reviews.

A classification label binary system was established before the training process began. All reviews with both lengths shorter than 20 characters and ratings at 1 or 5 stars received a 1 classification while the remaining reviews received a 0 classification. The authors designed a classification system to distinguish consistently short and polarized reviews when compared to ordinary review content. The VectorAssembler performed feature assembly to create one features vector from available scaled\_features and text\_features for classifier input. The complete prediction pipeline merged both the VectorAssembler component and the Random Forest model as stations for efficient training operation. [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). Achieving computational effectiveness alongside data representativeness required the selection of a 30% random sample from the total dataset before separating it into 80% training and 20% testing sets. A Random Forest Classifier attained maximum accuracy through ten trees with a maximum depth limitation of 3 to maintain an optimal generalization performance level. The model received assessment through MulticlassClassificationEvaluator which used accuracy as its primary evaluation metric after completion of training. [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing).

The model reached exceptional testing accuracy at 99.91 percent due to multiple characteristics in its operation. The classification goal presented a simplified set of tasks to recognize extremely short and polarized reviews which assisted the model in effective pattern detection. The engineered features successfully picked up substantial data variations that enhanced the model's capability to separate distinct categories. Random Forest effectively prevented overfitting through its ensemble structure due to using trees at a moderate depth. Data sampling served as a method to direct training without creating substantial sampling biases. The accurate performance of a predictive model must be examined because it may occur through natural sampling disproportion between classification categories. The high model performance stands because the review behaviors formed the basis of the designed label structure, and the sampling method maintained conventional patterns. A summary of the leading ten prediction outcomes contains original rating (overall) alongside review length, true rating information, and model prediction results.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 8: Model accuracy and sample predictions using Random Forest Classifier on the test dataset (Image Source: Databricks Community Edition\*)**

**5. Challenges and Improvements**

During data preparation and modeling stages together with feature engineering the project faced multiple difficulties that led to strong outcomes. The initial phase of work required addressing the built-in non-homogeneity and insufficient data density within the Amazon Electronics Reviews dataset [**“He & Garcia, 2009”**](https://ieeexplore.ieee.org/abstract/document/5128907?casa_token=jPkXsTrNKF0AAAAA:eTC-EH765_WGyQaoNd_oaj9IeUopHOj_jgSFKc3SDAjvkXXa52VbRw-CmWwm34Ax_rq6T1o). The review dataset contained extensive information but specific types of reviews including short ones and highly polarized ones were observed rarely among the data samples. The design process for labels required extreme care since it needed to produce meaningful classification problems with adequate training data quantities.

The text vectorization process led to major difficulties in managing its resulting dimensionality challenges. HashingTF enabled an efficient processing workflow for textual data without building an explicit vocabulary although the user needed to determine an optimal number of features to strike a balance between representation strength and computational performance. Standardizing and assembling numerical and textual features proved difficult as it required thorough preprocessing work to be executed through the SparkML pipeline. [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). The selection of suitable tree depth parameters along with tree count values was essential to stop overfitting when dealing with simple binary classification [**“Breiman, 2001”**](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf). The data analysis required sampling strategies to decrease computational resource requirements without sacrificing important patterns that the information provided.

Future work should investigate enhanced natural language processing methods such as TF-IDF vectorization and word embeddings to extract more complex relationships from review texts. The model performance can be enhanced by testing additional complex modeling approaches including Gradient Boosted Trees in combination with deep learning methods to detect delicate patterns. Supplementing the main evaluation criteria with measurements of F1-score and precision and recall would offer a more refined perspective on model performance when dealing with class imbalance diseases. [“Saito & Rehmsmeier, 2015”](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432), [“Mikolov et al., 2013”](https://arxiv.org/abs/1310.4546), [“Friedman, 2001”](https://www.jstor.org/stable/2699986?casa_token=rlWgmrwfiHgAAAAA:h5kdCCegiWWaLpkcVABedWR2GW43tqsz2VmkG5FkmnJQtXGzEu7Ddvbj1wCoBJFRSebQVgwE5ZeXjFBeua3g2VK3bvM9pXp_A3iP6JGskeWdTs_yuA).

Overall, despite these challenges, the project successfully demonstrated the end-to-end application of Spark for large-scale data management, feature engineering, and machine learning, resulting in a highly accurate predictive model.

**Conclusion**

This study proved that Apache Spark can practically process large datasets while engineering features alongside machine learning tasks using actual review data [“Zaharia et al., 2016”](https://www.researchgate.net/publication/310613994_Apache_spark_A_unified_engine_for_big_data_processing). By applying sequential data processing methods and cleansing operations to the Amazon Electronics Reviews dataset it became prepared for predictive modeling use. The combination of engineered meaningful features with Random Forest Classifier and vectorized text data resulted in 99.91% accuracy detection that verified the efficiency of the implemented preprocessing and modeling techniques. [“Breiman, 2001”](https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf)**.** Data handling techniques together with feature design excellence and model selection precision proved necessary to achieve successful outcomes working with extensive datasets. The project demonstrated successful minimization of data imbalance and computational constraints together with feature sparsity usingsampling and robust model structure and feature scaling methods.["Han, Kamber, & Pei, 2011”](https://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf) and [“He & Garcia, 2009”](https://ieeexplore.ieee.org/abstract/document/5128907?casa_token=jPkXsTrNKF0AAAAA:eTC-EH765_WGyQaoNd_oaj9IeUopHOj_jgSFKc3SDAjvkXXa52VbRw-CmWwm34Ax_rq6T1o)

The future project development creates possibilities for complex analysis through enhanced natural language processing and extended evaluation methods surpassing basic accuracy metrics. The project showed both Spark's big data analytics capabilities alongside applicable best practices for working with real-world large datasets in end-to-end data science operations.[“Mikolov et al., 2013”](https://arxiv.org/abs/1310.4546), and [“Saito & Rehmsmeier, 2015”](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0118432).

**References:**

* Katal, A., Wazid, M., & Goudar, R. H. (2013). Big Data: Issues, Challenges, Tools and Good Practices. *International Conference on Emerging Trends and Applications in Computer Science*. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Katal%2C+A.%2C+Wazid%2C+M.%2C+%26+Goudar%2C+R.+H.+%282013%29.+Big+Data%3A+Issues%2C+Challenges%2C+Tools+and+Good+Practices.+International+Conference+on+Emerging+Trends+and+Applications+in+Computer+Science.&btnG=)
* Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., ... & Stoica, I. (2016). *Apache Spark: A Unified Engine for Big Data Processing*. Communications of the ACM. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Zaharia%2C+M.%2C+Chowdhury%2C+M.%2C+Das%2C+T.%2C+Dave%2C+A.%2C+Ma%2C+J.%2C+McCauley%2C+M.%2C+...+%26+Stoica%2C+I.+%282016%29.+Apache+Spark%3A+A+Unified+Engine+for+Big+Data+Processing.+Communications+of+the+ACM.&btnG=)
* McAuley, J., Targett, C., Shi, Q., & van den Hengel, A. (2015). *Image-Based Recommendations on Styles and Substitutes*. Proceedings of the 38th International ACM SIGIR Conference [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09McAuley%2C+J.%2C+Targett%2C+C.%2C+Shi%2C+Q.%2C+%26+van+den+Hengel%2C+A.+%282015%29.+Image-Based+Recommendations+on+Styles+and+Substitutes.+Proceedings+of+the+38th+International+ACM+SIGIR+Conference&btnG=)
* Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Han%2C+J.%2C+Kamber%2C+M.%2C+%26+Pei%2C+J.+%282011%29.+Data+Mining%3A+Concepts+and+Techniques+%283rd+ed.%29.+Morgan+Kaufmann.&btnG=)
* Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Breiman%2C+L.+%282001%29.+Random+Forests.+Machine+Learning%2C+45%281%29%2C+5%E2%80%9332.&btnG=)
* Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., ... & Stoica, I. (2016). *Apache Spark: A Unified Engine for Big Data Processing*. Communications of the ACM. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Zaharia%2C+M.%2C+Chowdhury%2C+M.%2C+Das%2C+T.%2C+Dave%2C+A.%2C+Ma%2C+J.%2C+McCauley%2C+M.%2C+...+%26+Stoica%2C+I.+%282016%29.+Apache+Spark%3A+A+Unified+Engine+for+Big+Data+Processing.+Communications+of+the+ACM.&btnG=)
* He, H., & Garcia, E. A. (2009). *Learning from Imbalanced Data*. IEEE Transactions on Knowledge and Data Engineering. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09He%2C+H.%2C+%26+Garcia%2C+E.+A.+%282009%29.+Learning+from+Imbalanced+Data.+IEEE+Transactions+on+Knowledge+and+Data+Engineering&btnG=)
* Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). *Distributed Representations of Words and Phrases and their Compositionality*. Advances in Neural Information Processing Systems. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Mikolov%2C+T.%2C+Sutskever%2C+I.%2C+Chen%2C+K.%2C+Corrado%2C+G.S.%2C+%26+Dean%2C+J.+%282013%29.+Distributed+Representations+of+Words+and+Phrases+and+their+Compositionality.+Advances+in+Neural+Information+Processing+Systems&btnG=)
* Saito, T., & Rehmsmeier, M. (2015). *The Precision-Recall Plot is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets*. PLOS ONE. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Saito%2C+T.%2C+%26+Rehmsmeier%2C+M.+%282015%29.+The+Precision-Recall+Plot+is+More+Informative+than+the+ROC+Plot+When+Evaluating+Binary+Classifiers+on+Imbalanced+Datasets.+PLOS+ONE.&btnG=)
* Friedman, J.H. (2001). *Greedy Function Approximation: A Gradient Boosting Machine*. Annals of Statistics, 29(5), 1189-1232. [Link](https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=%E2%80%A2%09Friedman%2C+J.H.+%282001%29.+Greedy+Function+Approximation%3A+A+Gradient+Boosting+Machine.+Annals+of+Statistics%2C+29%285%29%2C+1189-1232.&btnG=)