CASE STUDIES WITH PYSPARK AND MACHINE LEARNING

a project report submitted by

JOSWIN V JAISON (URK18CS097)
in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

under the supervision of

Prof. Shibin David

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING AND TECHNOLOGY

KARUNYA INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed-to-be-University under Sec-3 of the UGC Act, 1956)
Karunya Nagar, Coimbatore - 641 114, India.

APRIL 2022

ACKNOWLEDGEMENT

First and foremost, we praise and thank **ALMIGHTY GOD** for giving us the will power and confidence to carry out our project.

We are grateful to our beloved founders Late **Dr. D.G.S. Dhinakaran**, **C.A.I.I.B, Ph.D.,** and **Dr. Paul Dhinakaran**, **M.B.A., Ph.D.,** for their love and always remembering us in their prayers.

We extend our thanks to **Dr. P. Mannar Jawahar, Ph.D.,** our honorable vice chancellor, **Dr. E. J. James, Ph.D.,** and **Dr. Ridling Margaret Waller, Ph.D.,**our honorable Pro-Vice Chancellor(s) and **Dr. R. Elijah Blessing, Ph.D.,** our respected registrar for giving us the opportunity to carry out this project.

We are thankful to **Dr. G. Prince Arulraj, M.E., Ph.D.,** Dean (Engineering & Technology) for his support and encouragement.

We would like to place our heart-felt thanks and gratitude to **Dr. J. Immanuel Johnraja, Ph.D.,** HOD, Department of Computer Science and Engineering for his encouragement and guidance.

We are grateful to our guide, **Mr. Shibin David,** Assistant Professor, Department of Computer Science and Engineering for his valuable support, advice and encouragement.

We also thank all the staff members of the Department for extending their helping hands to make this project work a success.

We would also like to thank all my friends and my parents who have prayed and helped me during the project work.

BONAFIDE CERTIFICATE

Certified that this project report **“CASE STUDIES WITH PYSPARK AND MACHINE LEARNING”** is the bonafide work of

“JOSWIN V JAISON [URK18CS097] ” who carried out the project work under my supervision.

SIGNATURE

Dr. J. Immanuel Johnraja

Head of the Department

Department of Computer Science and Engineering

SIGNATURE

Mr. Shibin David

Supervisor

Assistant Professor
Department of Computer Science and Engineering

Submitted for the Project Viva Voce held on……23 April 2022………………….

Examiner

Karunya INSTITUTE OF TECHNOLOGY AND SCIENCES
(Declared as Deemed to be University under Sec. 3 of the UGC Act, 1956)
A CHRISTIAN MINORITY RESIDENTIAL INSTITUTION
AICTE Approved & NAAC Accredited
Karunya Nagar, Coimbatore - 641 114, Tamil Nadu, India

ARISE AND SHINE

Department of Computer Science and Engineering

Date: 21 April 2022

Undertaking

I, __Joswin V Jaison___ (URK18CS097_) have started my internship cum project work at Cognizant Technology Solutions on January 27 2022 . As on date, I have completed the module number 4 and started module number 5. I declare that the above information is true and I have enclosed a proof of internship cum project or employment offer (whichever is later) to this report.

Name of the student : JOSWIN V JAISON

Register Number : URK18CS097

Signature with date

Joswin

ABSTRACT

This report presents the compendia of my learning experience at Cognizant Technology Solutions I have started my internship from January 27 2022. I have been assigned with the business unit named ‘Data’. And within data business unit am working under big data technology. This report mainly portrays my insights as well as mini project experiences on various data science technologies at Cognizant Technology Solutions. The primary importance of data at organizations is that it improves the quality of life for its clients. The domain of data science is growing at a rapid rate so that the companies can get valuable analytical insights for its client. Big data analytics enables data scientists to examine large and complex varieties of data using predictive modeling, statistics and other analytics to uncover hidden patterns, market trends, customer preferences, unknown correlations and other useful information to help organizations improve their decision-making. It enables the companies to understand gigantic data from multiple sources and derive valuable insights to make smarter data driven decisions. There are various reasons why Data Science is important in business. Data Science enables enterprises to measure, track, and record performance metrics for facilitating enterprise-wide enhanced decision making. Companies can analyze trends to make critical decisions to engage customers better, enhance company performance, and increase profitability. Data Science models use existing data and can simulate several actions. Thus, companies can devise the path to reap the best business outcomes. Data Science helps organizations identify and refine target audiences by combining existing data with other data points for developing useful insights. Data Science also helps recruiters by combining data points to identify candidates that best fit their company needs. According to one study, the global Data Science market is expected to grow to $115 billion by 2023. As a result data plays a major role in the working of organizations. And the demand of skilled data scientists or big data experts is also increasing in a greater manner.

TABLE OF CONTENTS

List of figures

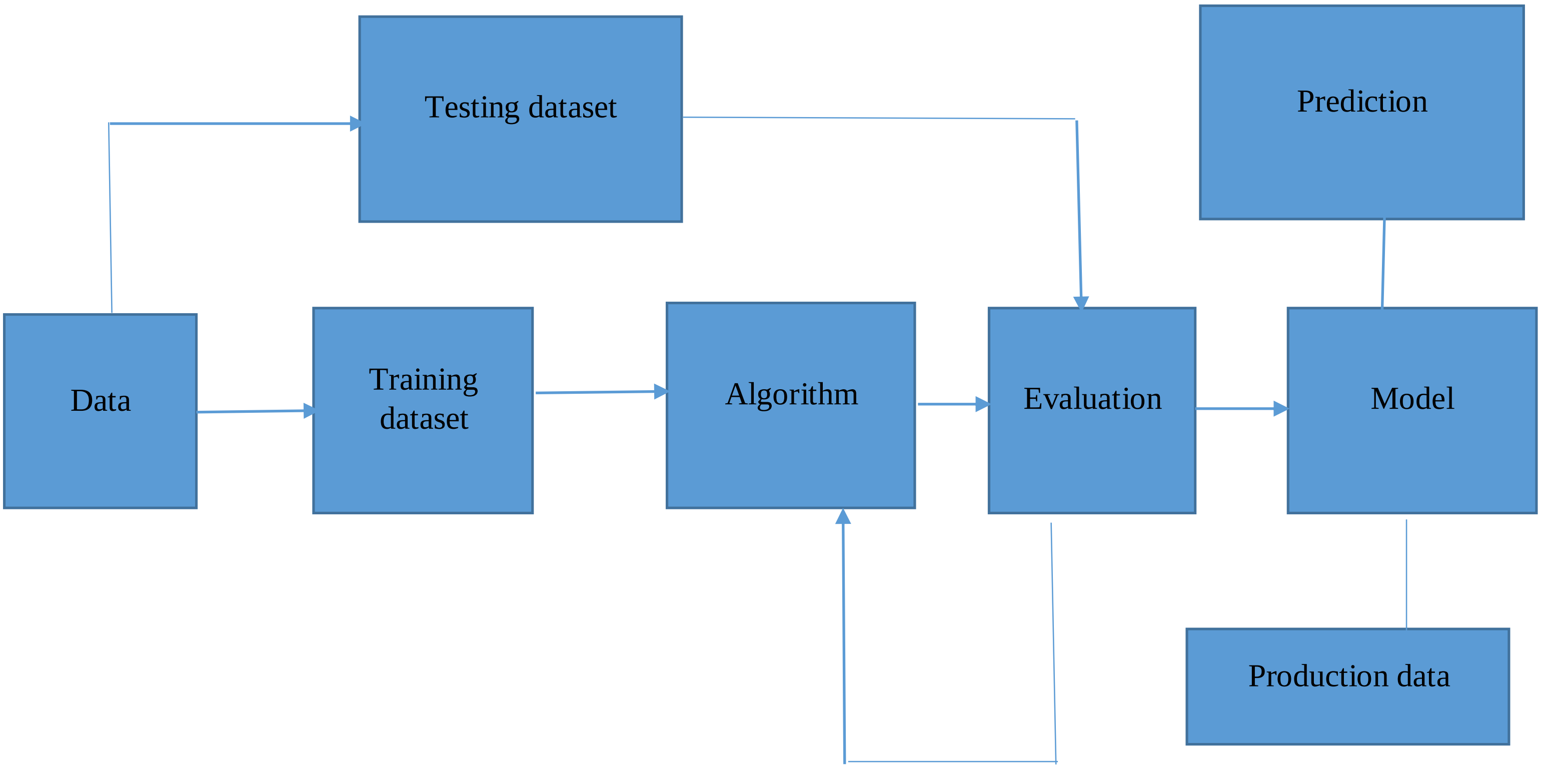


Figure 1 Machine Learning Work flow

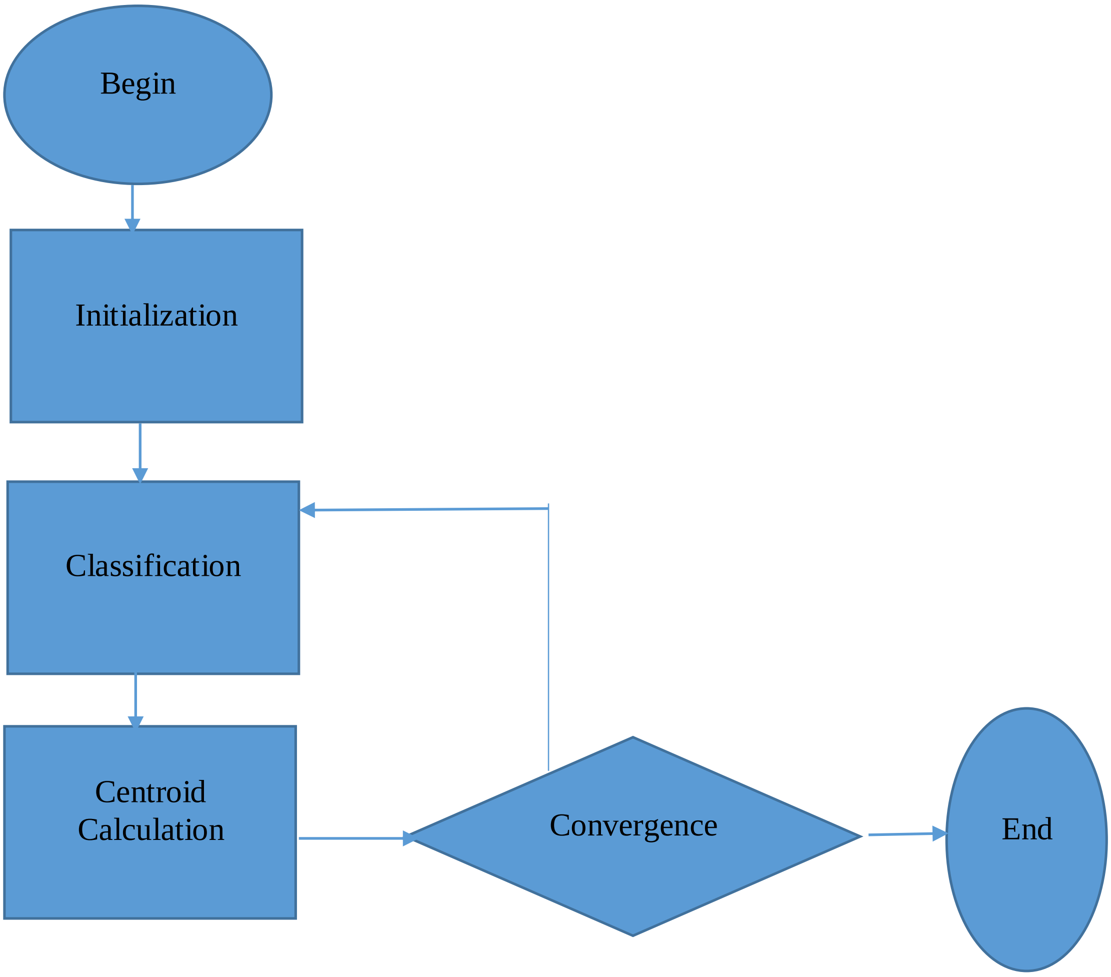


Figure 2 K Mean Clustering work flow

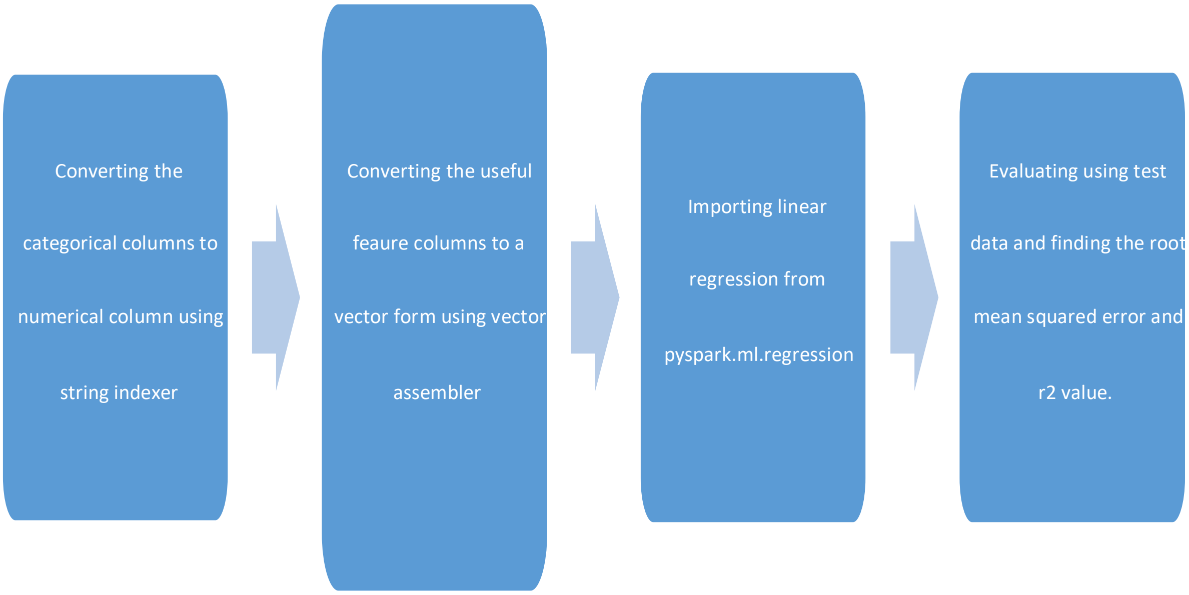

Converting the categorical columns to numerical column using string indexer
Converting the useful feaure columns to a vector form using vector assembler
Importing linear regression from pyspark.ml.regression
Evaluating using test data and finding the root mean squared error and r2 value.

Figure 3 Linear Regression Case Study

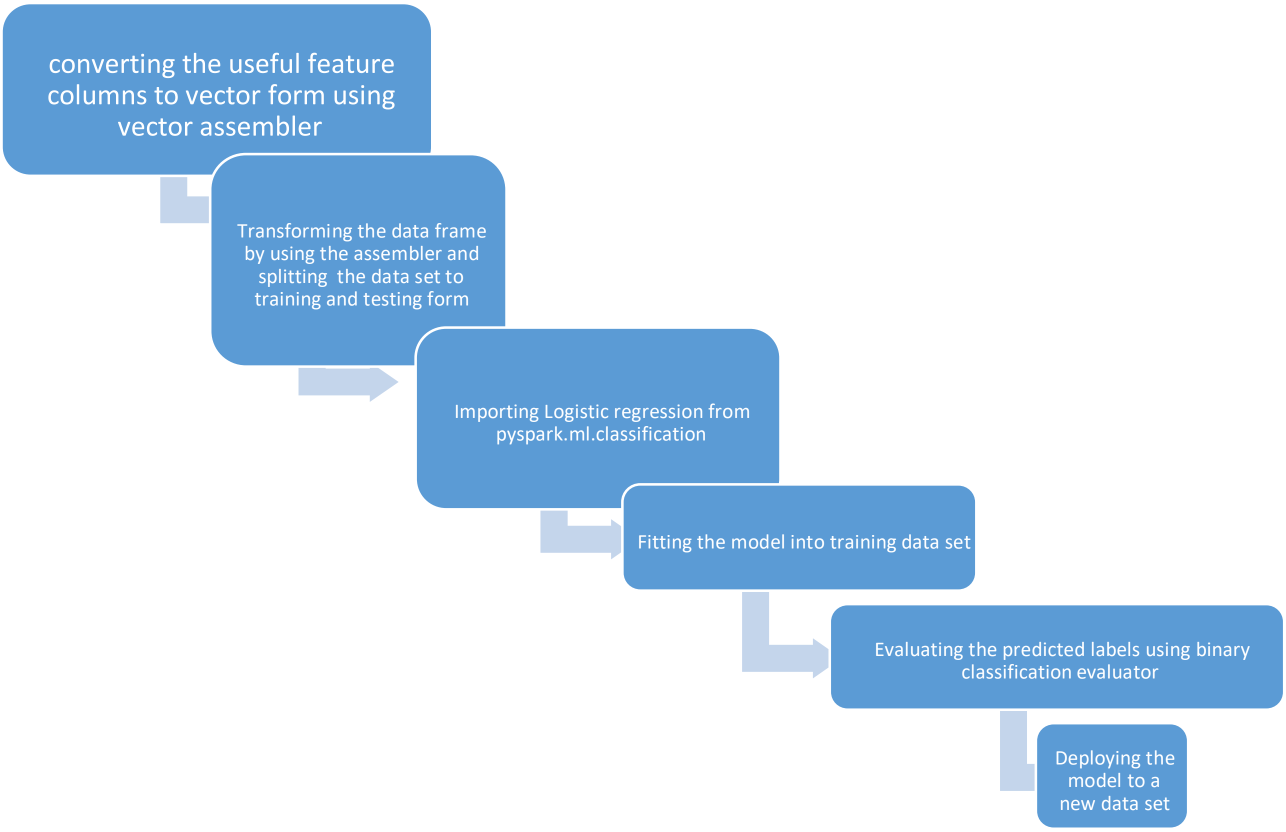

converting the useful feature columns to vector form using vector assembler
Transforming the data frame by using the assembler and splitting the data set to training and testing form
Importing Logistic regression from pyspark.ml.classification
Fitting the model into training data set
Evaluating the predicted labels using binary classification evaluator
Deploying the model to a new data set

Figure 4 Logistic regression case study

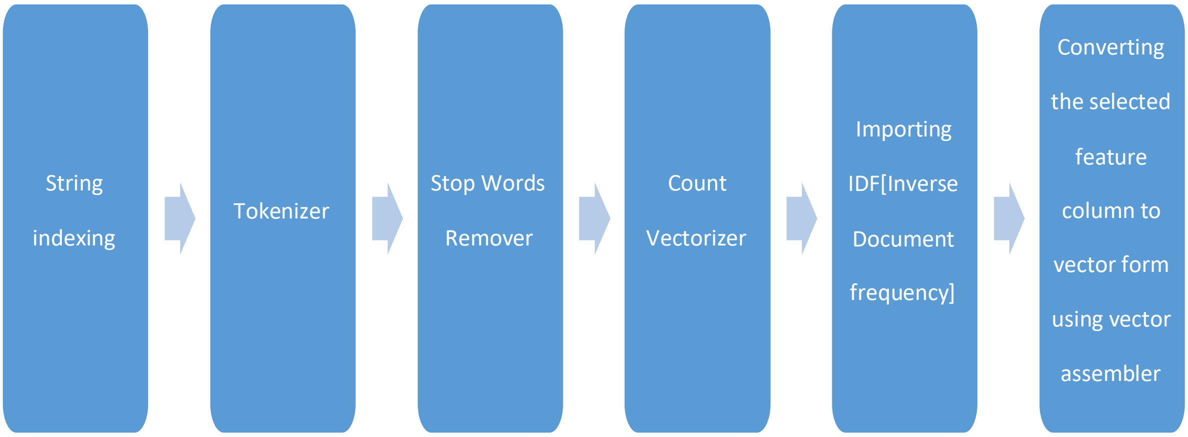

String indexing
Tokenizer
Stop Words Remover
Count Vectorizer
Importing IDF[Inverse Document frequency]
Converting the selected feature column to vector form using vector assembler

Figure 5 Stages in pipeline created for developing a spam filter using natural language processing

LIST OF TABLES

Table 1 Evaluation Metrics

Name of case study	Machine learning algorithm used	Evaluation metrics and results
Predicting the number of crew members needed for a ship company	Linear Regression	R square:0.8547 Root mean square error :1.4223
Customer churn prediction for a marketing agency	Logistic Regression	Area under curve:0.740
Predicting whether a university is private or not	Random forest classifier,Decision tree classifier,Gradient boost trees.	Area under curve RFC:0.9913 GBT:0.973 DTC:0.922 Accuracy when using random forest classifier:0.96137 F1 score when using random forest classifier:0.9611
Predicting which preservative cause heavy spoilage of dog food	Random forest classifier	Used feature Importances score to calculate the importance of each feature Feature A : 0.0174 Feature B: 0.012 Feature C:0.94 Feature D:0.0221
Creating a movie recommendation engine	Collaborative filtering. Pyspark has ALS algorithm for collaborative filtering	Root mean square error: 1.788

Analyzing the patterns in hacking	K Means clustering	Haven’t used any evaluation metrics since there is no splitting into testing and training dataset
Creating a spam filter	NLP techniques and Naïve Bayes approach	Accuracy : 0.929

Table 2 : Equations ,Formula and abbreviations

Accuracy	(TP+TN)/TP+TN+FP+FN
Precision	TP/(TP+FP)
Recall	TP/TP+FN
F1	(2*Precision*Recall)/(Precision +Recall)
TP	True positive
TN	True negative
FP	False positive
FN	False negative
AUC[Area under the curve]	∫ab f(x)dx
Root mean square error	$RMSE = \sqrt{(f - o)^2}$
R square	R2=1−(Unexplained variation/Total variation)
Linear regression line	Y=a+bx x is the explanatory variable Y is the dependent variable

CHAPTER 1

INTRODUCTION

Cognizant internship is a pre requisite skill development program which makes the selected candidates ready for employment . The successful completion of internship forms a critical part of employment with respective company .The internship program consists of learning curriculum as per the learning track assigned. The learning path will include in-depth sessions, hands on exercise and project work. There will also be a series of webinars, quizzes, mentor connects, leader connect ,code challenges, assessments etc. to accelerate learning. The outcomes during the Internship would be monitored through formal evaluations. My internship at cognizant started from January 27 2022. It will be a 4 month internship . The interns are designated as programmer Analyst trainee at cognizant. They should follow the standard procedures of cognizant including daily and weekly attendance submission. The internship timing is from 9 am to 7pm from Monday to Friday. Each candidate will be mapped to batches based on the domain assigned. They will be further mapped into cohorts. Each cohort will be assigned with a coach. My cohort consists of 44 members. Each cohort will be assigned with a technical trainer who helps the candidates in clearing their doubts. There will be behavioral sessions which focus mainly on soft skills , do 's and don' t's in industry etc. The domain I have got is big data.I have done various case studies using a tool called pysark. Big data is a voluminous and diverse collection of data from a variety of sources that is too complicated to be handled by traditional database management applications or people. The key difference between big data and "normal data" is big data's capacity to organize and store complex and vast amounts of data. There are various business benefits related to big data. It helps organizations create new growth opportunities and entirely new categories of products and services by combining and analyzing any possible source of data. Information about the offerings, buyers and suppliers, and consumer preferences can be captured and analyzed to optimize business processes. For example, retailers can easily optimize their stock, based on predictive models generated from social media data, web search trends and weather forecasts. Big Data Analytics is much more objective than the older methods and companies can make the correct business decisions using data insights. There was a time when companies could only interact with their customers on one in stores. And there was no way to know what individual customers wanted on a large scale. But that has all changed with the coming of Big Data Analytics. Now companies can directly engage with each customer online personally and know what they want . Due to this huge demand the need of skilled big data experts are increasing .This internship at Cognizant is mainly concerned with making the candidate ready for a job related to big data. Pyspark is one of the tool that I have been working on..

CHAPTER 2
SYSTEM ANALYSIS

PySpark is the Python API for Apache Spark, an open source, distributed computing framework and set of libraries for real-time, large-scale data processing. The primary objective of the case studies that I have performed is to get familiarized with various machine learning algorithms as well as Pyspark.I have performed seven case studies using Pyspark and various machine learning algorithms.The first case study deals with linear regression.It is the process of building a predictive model for a ship company to predict the number of crew members.The features given are Ship name,cruise line,age,tonnage,number of passengers,length,number of cabins etc. By creating the particular model it will help given ship company to predict how many crew members they need if given various features. The second case study deals with creating a predictive model for a marketing agency to predict whether the client will churn or not.If they churn they can be assigned to an account manager.This model is primarily based on the concept of logistic regression . It is a classification algorithm. The features are name of the client,age of the client,total purchase,whether an account manager is assigned or not,total years as a customer,number of websites that use the service,onboarding date,client headquarters address, name of client company. The label column is churn .It can be 0 or 1.The value 0 indicate that customer will not churn and 1 indicate that customer will churn. The third case study deals with creating a model to predict whether a university is private or not. The objective of the particular case study is to predict whether a particular university is private or public based on various features.The features are name of university,number of applications received ,number of applications accepted,number of new students enrolled,how many percent of students come from top 10 percent of high school class,how many percent of students come from top 20 percent of the high school class. The label column is private.It is a categorical column that has 2 values Yes or No.Here we have created model using decision tree classifier ,random forest classifier and gradient boost trees. The models created using different algorithms are compared.The fourth case study I have performed is creating a model based on dog food dataset. The primary objective of the case study is to predict which chemical contributes most to the spoiling of dog food. Here I have used random forest to predict which feature has the most predictive power.There are 4 features. Percentage of chemical A, percentage of chemical B,percentage of chemical C,percentage of chemical D.The target column is a column named Spoiled which indicate whether a food has been spoiled or not. It can be useful in cases where we need to create a model to predict which feature is the most powerful.The fifth case study I have performed is creating a movie recommendation engine using collaborative filtering approach using pyspark.The datset used here is movie lens dataset.It consists of 3 columns userId,movieId and rating.The algorithm with which I have implemented collaborative filtering is ALS. Apache Spark ML implements alternating least squares (ALS) for collaborative filtering, a very popular algorithm for making recommendations. The sixth case study deals with creating a clustering model for getting information about the involvement of a particular hacker related to a security attack.The datset include various features like how long the hacking session lasted in minutes,number of bytes transferred during session,whether or not the hacker used Kali linux,number of servers corrupted during attack,number of pages illegally accessed , location where attack come from ,the estimated typing speed . One significant fact related to the case study is that the total number of hacks done by each of the hacker will be almost equal or they are evenly distributed. By taking the particular clue as input we should predict the involvement of a particular hacker. The next case study deals with building a spam detection filter using the concepts of natural language processing. Data consists of volunteered text messages from a study in Singapore and spam text from a UK reporting site.

CHAPTER 3

SYSTEM DESIGN

The project mainly deals with solving various cases using Pyspark with the help of machine learning algorithms.Pyspark helps to analyze large amount of data as compared to pandas.I have worked with 7 different case studies. The design of the particular system using different machine learning algorithms are mentioned below:

3.1. LINEAR REGRESSION MODEL TO PREDICT CREW MEMBERS FOR A SHIP COMPANY

Linear regression is one of the most well known and well understood algorithms in statistics and machine learning.Linear regression is a linear model, e.g. a model that assumes a linear relationship between the input variables (x) and the single output variable (y). More specifically, that y can be calculated from a linear combination of the input variables (x).When there is a single input variable (x), the method is referred to as simple linear regression. When there are multiple input variables, literature from statistics often refers to the method as multiple linear regression. We can import Linear Regression from Ml library of pyspark.

3.2. LOGISTIC REGRESSION MODEL TO PREDICT WHETHER CUSTOMER WILL CHURN OR NOT

Logistic regression is a classification problem. Here the label has 2 values either 0 or 1.So logistic regression is preferred. Logistic Regression is used when the dependent variable(target) is categorical.This type of statistical analysis (also known as logit model) is often used for predictive analytics and modeling, and extends to applications in machine learning. In this analytics approach, the dependent variable is finite or categorical: either A or B (binary regression) or a range of finite options A, B, C or D (multinomial regression). It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help us to predict the likelihood of an event happening or a choice being made.We can import Logistic regression from Ml library of pyspark.There are mainly three types of logistic regression.

a.Binary logistic regression

The label has only 2 values. In this case I have used binary logistic regression .The output belong to 2 categories either ‘the customer will churn’ or ‘the customer will not churn’.

b.Multinominal logistic regression

The label belongs to 3 or more categories.For example predicting which type of food is preferred to a particular person

c.Ordinal logistic regression

Here the labels will be ordered.For example predicting the rating of a movie.

3.3.MODEL TO PREDICT WHETHER UNIVERSITY IS PRIVATE OR NOT

The primary objective of the model is to compare the working of 3 different algorithms random forest,decision trees and gradient boost as well as to create a model to predict whether the university is private or not.From pysapark.ml.classification we can import Decision Tree classifier,random forest classifier and GBT classifier.

a.Decision tree classifier:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a treestructured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. Since the desired output is discrete value decision tree classifier is used.

b.Random forest classifier:

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.It takes less training time as compared to other algorithms.It predicts output with high accuracy, even for the large dataset it runs efficiently.It can also maintain accuracy when a large proportion of data is missing.

c.GBT classifier:

It is a technique of producing an additive predictive model by combining various weak predictors, typically Decision Trees.Gradient Boosting Trees can be used for both regression and classification.Here I have used classifier since the label is discrete.It often provides predictive scores that are far better than other algorithms.It can handle missing data without any imputation.

3.4. RANDOM FOREST MODEL TO FIND OUT WHICH PRESERVATIVE CAUSE HEAVY SPOILAGE OF FOOD

The main objective of the case study is to find out which feature contributes to the most in label prediction.I have used random forest classifier for the particular case study. The feature importance (variable importance) describes which features are relevant. It can help with better understanding of the solved problem and sometimes lead to model improvements by employing the feature selection.We can import random forest classifier from pyspark.ml.classification. The biggest advantage of Random forest is that it relies on collecting various decision trees to arrive at any solution. This is an ensemble algorithm that considers the results of more than one algorithms of the same or different kind of classification.

3.5. CREATING MOVIE RECOMMENDATION ENGINE USING COLLABORATIVE FILTERING

There are primarily 2 types of recommendation systems.Content based recommendation system and collaborative filtering based recommendation system.Content based recommendation system focus on the attributes of the items and gives us recommendation based on similarities between them.Collaborative filtering approach is more commonly used than content based recommendation system because it usually gives better results.Spark.ml currently supports model based collaborative filtering in which users and products are described by a small set of latent factors that can be used to predict missing entries. Collaborative filtering aggregates the past behaviour of all users. It recommends items to a user based on the items liked by another set of users whose likes (and dislikes) are similar to the user under consideration. This approach is also called the user-user based CF.Apache Spark ML implements alternating least squares (ALS) for collaborative filtering, a very popular algorithm for making recommendations. ALS recommender is a matrix factorization algorithm that uses Alternating Least Squares with Weighted-Lamda-Regularization (ALS-WR). The basic idea is to decompose a matrix in smaller parts in the same way we can do for a number.

For instance, we can say that the number four can be decomposed in two times two (4 = 2 x 2). In the same way, we can do a decomposition of a matrix.We can import ALS from pyspark.ml.recommendation.By default we need to specify userCol,ratingCol and itemCol in ALS algorithm. The architecture of the recommendation engine is dependent on the business domain and the attributes of the dataset at one’s disposal. For instance, customers on eBay frequently provide ratings for products scaling on 1 (unhappy) to 5 (very happy). Spotify holds information about the gender of music one listens to. Uber eats should know what our favourite type of food is. Instagram has user patterns of likes in images. Such data sources document interactions between users and products (items). In addition, the platform may access personal information from users, such as their location, age, sex and so on.We need to predict by how much percent does the user will like the particular movie.

3.6. ANALYZING THE PATTERNS IN HACKING USING K MEANS CLUSTERING APPROACH

The main idea behind this case study is to analyze the various patterns in a particular dataset related to hacking.The problem statement is to predict the involvement of a particular hacker during the security attack.The algorithm used here is KMeans clustering approach.It is an un supervised learning algorithm. K-Means Clustering groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.The k-means clustering algorithm mainly performs two tasks: a.Determines the best value for K center points or centroids by an iterative process b. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.From pyspark.ml.clustering we can import KMeans.The final model gives a table which shows the count of total no of attacks done by each cluster which is identified as a hacker. It is based on the value of k I have chosen.

3.7 SPAM FILTER USING NATURAL LANGUAGE PROCESSING

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There’s a good chance we have interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-totext dictation software, customer service chatbots, and other consumer conveniences. But it also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processesThe case study deals with creating a spam filter using the concepts of natural language processing.To perform the particular case study I have imported Tokenizer,StopWordsRemover, CountVectorizer,IDF,StringIndexer. Tokenization is breaking the raw text into small chunks. Tokenizor breaks the raw text into words, sentences called tokens.Stop words remover removes the stop words from the particular token list. Count Vectorizer is used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text. TF-IDF (Term Frequency - Inverse Document Frequency) is a handy algorithm that uses the frequency of words to determine how relevant those words are to a given document.String Indexer helps to convert class column into a numerical form.

CHAPTER 4
SYSTEM IMPLEMENTATION

4.1 IMPLEMENTATION OF LINEAR REGRESSION MODEL FOR A SHIP COMPANY

To build the particular model I have created the spark application.Then the csv file is read using spark.read.csv.The feature named cruise line is categorical .We can’t drop the particular feature since it is useful.So we are converting the particular categorical feature into a numerical column called categories.I am fitting and transforming the particular string indexer to my dataset.The dataset now contains a new column called categories. We can convert the particular useful features into a new vector so that it can be useful for our analysis.The next step is to split the final data containing features and crew members into training and testing dataset.The percentage used here is 70 percentage and 30 percentage .The next stage is to import LinearRegression from pyspark.ml.regression.And crew is designated as the label.Then we are fitting the linear regression model to our training data and evaluating on the test data.The particular r2 value obtained is 0.85 and the particular root mean squared error is 1.42.Then the correlation between 2 features are obtained.The selected 2 features are crew and passengers.The correlation value is 0.91.It implies that the total number of crew members are highly dependent on factor called total number of passengers.

4.2 IMPLEMENTATION OF LOGISTIC REGRESSION MODEL FOR CHURN PREDICTION

The dataset includes features like names,age,total purchase,account manager assigned or not,onboarding date ,location etc.Vector Assembler is imported from pyspark.ml.feature to convert all the useful features into a vector form.The useful features include age,total purchase made by the client,whether the account manager is assigned or not,years,num_sites.The particular output column which contains the vector is features.Then we are transforming the data frame using the assembler .And the new data frame will contain a new feature called features.Then we are converting the new data frame into training and testing dataframe based on the ratio 70% and 30%.Then logistic regression is imported from pyspark.ml.classification.Here the label is assigned as churn.The next step is to fit and transform the logistic regression model into the final data frame.The area under the curve is 0.74.Then a new dataset is read and the predictions are evaluated on the new data frame.

4.3 IMPLEMENTATION OF A MODEL TO PREDICT WHETHER THE UNIVERSITY IS PRIVATE OR NOT

The objective of the case study is to predict whether the particular university is private or not.VecorAssembler is imported to group all the necessary features.These features are grouped into a new column called features.The label column is named as private.It is categorical .In order to perform the analysis efficiently we are converting the categorical column into numerical using String indexer.It can have two values .Either 0 or 1.Then I am fitting and transforming the particular indexer into the data frame and a new data frame is obtained. The new data frame is being split into training and testing data.From pyspark.classification I have imported Decision Tree classifier,Random forest classifier and GBT classifier.The private index column which is the transformed column of private column is considered as the label column. The feature column which consists of various features is designated as the feature column in algorithms.Then I have fitted this model on testing data and performed transformation on testing data.Binary classification evaluator and Multi class classification evaluator is imported and the metrics of various models are compared.

4.4 ANALYZING DOG FOOD SPOILAGE USING RANDOM FOREST MODEL

The particular dataset contains various features named A,B,C,D which represents various preservatives. The column named Spoiled is considered as label which has 2 values either 0 or 1.0 signifies that the food is not spoiled and 1 signifies that the food is spoiled.Vector Assembler is imported from pyspark.ml.feature which combines various features so that it will be helpful for our analysis.And I have transformed the assembler into the dataset.Since it is a classification problem I have imported random forest classifier from pyspark.ml.classification.A data frame consisting of two features called features and Spoiled is created.Then I have fitted the random forest classifier into the new data frame. The new model consists of an attribute called feature Importance which shows the importance of every preservative .As a result it is concluded that chemical C is of high importance and it contributes the most in food spoilage.

4.5 IMPLEMETATION OF A MOVIE RECOMMENDATION ENGINE USING COLLABORATIVE FILTERING

The case study deals with creating a movie recommendation engine.I have imported ALS from

Pyspark.ml.recommendation.And regression evaluator is imported from pyspark.ml.evaluation

The particular data is split into training data and testing data.There are various parameter to specify in ALS algorithm.It includes maxIter,regParam,userCol,ratingCol and itemCol.Then the particular model is fitted into the training data and the test data is transformed using the model.When we are calling the new data frame a new column called prediction is obtained.Then the regression evaluator is called.Here rating column is specified as the labelCol and column named prediction is designated as the predictionCol.The metric specified is rmse.From the test data I have selected a new data frame which consists of movieId belonging to a particular user.Then I have transformed the new data frame using the particular model.It produce a new column called prediction which shows by how much percent the user is likely to enjoy the movie.

4.6 ANALYZING PATTERNS IN HACKING USING K MEANS CLUSTERING

The primary objective of the case study is to analyze the hacking patterns using K Means clustering and investigating the involvement of a particular hacker.To implement the particular case I have imported KMeans clustering from pyspark.ml.clustering.The vector assembler is imported from pyspark.ml.feature.It will group the necessary features together into a new column called features.

The data frame is transformed by using the particular assembler.These features are in different range.It is scaled to a standardized form using Standard Scaler.The scaler object is fitted into the training data and the data is transformed using the scaler model created.The new data frame consists of a new feature named scaled features which displays the features in a scaled form so that it will be helpful for our analysis.Now the kmeans clustering object is created two times where the specified k value is 2 and

3.The new data frame consists of column prediction which shows the name of cluster belonging to a particular hacking activity. The data frame in which k value is 3 shows the number of hacks done by each hacker if there were 3 hackers.The case study have given a hint that the number of hacks done by each hacker is evenly distributed.Since the count of hacks done by each hacker is different when k values is 3,it implies that the third hacker is not involved in the hacking attack.

4.7 SPAM FILTER USING NATURAL LANGUAGE PROCESSING

The particular data set consists of 2 column .The first column includes label and second column include the particular message.They are named as c0 and c1 by default.I have renamed them to class and message respectively.The length function is imported from pyspark.sql.functions.A new column called length is created which computes the length of each message.The class is grouped by mean of length where the mean length of spam is 138.67 and mean length of ham is 71.45.I have imported Tokenizer,Stop words remover,Count Vectorizer,IDF and String Indexer from pyspark.ml.feature.I have created a vector assembler which where the input columns are tf_idf and output column is length.The classification algorithm imported is naïve bayes.A pipeline is also imported since various stages are used.The fitting and transformation is done on the data.From the new data frame I have selected label and features.And it is split into training and testing dataset.Multi class classification evaluator is imported from pyspark.ml.evaluation.The accuracy is 0.92.

CHAPTER 5

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1.Week 1-3 : January 27 to February 11

The first 2 weeks of my internship was mainly for induction activities. It involves the completion of onboarding formalities, completion of courses, participation in introductory sessions and leadership connects at cognizant.

- Onboarding : My onboarding was on 27th January 2022.
- Introductory sessions: The introductory sessions were conducted on topics like importance of professional etiquette in corporate environment.
- Security Awareness training: The mission of corporate security training is to provide comprehensive world class security and risk management capabilities to protect and enable Cognizant's global business while creating client value and competitive differentiation. This particular session helped me in getting an overview on security best practices in Cognizant. This practices include protecting our credentials and data, protecting our assets and network systems, protecting our workspace. This session also gave an idea on phishing attacks and the process to recognize a phishing attempt .A brief overview on cognizant policies, procedures and standards were also given during this session.
- Leader Connects : Directors of various fields in cognizant took session on first week. Mr Saranjit Deb conducted a session on the different areas available to explore in cognizant .Various platforms like Genc learn, Cognizant Udemy platform were introduced during this connect. Another one session was conducted by Mrs Meenakshi who is the insurance products practice lead at Cognizant. She shared her experience of working in cognizant for 11 years. We learnt more about the phases which company had undergone.
- Boot camp and Courses : Candidates should complete some mandatory courses during the first 2 weeks regardless of the domain .I was assigned with 4 courses during my first 2 week

  1.Course on Zero waste in 30

  This online course is designed to gain knowledge , resources and guidance and start driving change at personal level at your families and communities.

The course duration was 2.5 hours . I had to take the course through genc Learn platform. This course provided an insight on various waste management practices that we need to follow.

2. Courses assigned by Cognizant ethics and compliance team.

The particular courses which comes under Cognizant ethics and compliance category familiarize associates with the code of ethics ,global policies and procedures and decision making process in company.

a. Course on code of ethics and acceptable use

This course provides a brief idea on rules and regulations existing within the company. The behavior of employees play a crucial role in a particular organization. It familiarizes us with Cognizant's updated Code of Ethics and key global corporate policies (such as Acceptable Use Policy, Anti-Corruption Policy, Conflicts of Interest Policy, Global Privacy Policy, and Record Retention Policy) and procedures, and considers the responsibilities each of us has to act in ways that promote an ethical culture of mutual trust and respect, recognize and address risks to the company, and advance the business goals the right way. b. Course on data security.

We live up to our responsibilities. Our commitment to doing business ethically includes respecting privacy, protecting information, and safeguarding assets. The volume of information that our business receives, creates, and stores is significant and increasing. With the increase in ransom ware attacks, phishing attempts, and data protection regulations, Cognizant has been refreshing and strengthening its approach to security. A key component to that is better data privacy and management across the company. This course helps the candidates at Cognizant to be aware of various security related attacks as well as provides a motive to act against it.

c. Course on prevention of sexual harassment at workplace .

This course familiarizes associates in India with forms of sexual harassment, protections, and redress procedures under the Prevention, Prohibition and Redressal Act (POSH Act).

2. Week 3 : February 14-February 21

Database fundamentals

The learning objective of the particular week includes database design,SQL basics with DML and DDL statements, data modelling ,querying database. I have come across operators ,aggregate functions ,string functions ,date functions etc in the particular module. I have done the following courses during this week of data base fundamentals

1.Relational Database design

This course taught me how to create an effective relational database design using proven concepts and industry knowledge. Effective database designs will help make systems faster, improve data quality, and ensure future changes are easier.This course include helped in understanding normalization and type of normal forms. It helped to learn how to identify tables and how to create relationships.It also gave me an insight on the naming convention we need to follow while designing tables.

2. Understand SQL using the MySQL database. Learn Database Design and Data Analysis with Normalization and Relationships

This course helped me in getting a detailed knowledge and understanding of using MySQL, one of the most widely used databases in the world.This course provide an in depth knowledge on data base design .For this I have installed mySql workbench .The initial phase of this course focus mainly on data definition language,data manipulation language etc. Then it is shifted to aggregate functions , sub queries ,functions etc.This course helped me to understand and apply SQL with MySQL, including Database Design and Data Analysis.Here I have created a coffee store database and cinema booking data base . I have performed various SQL operations within this particular databases which helped me in enhancing my database skills as it is vital for data scientists and big data experts.

3.Data Modelling and relational database design using Erwin

This is mainly a theory oriented course which help us to learn how to develop data models and maintain them using the data modelling tool called Erwin. It helped me in creating entity relationship diagrams by identifying entities ,attributes, relationships and constraints from a set of requirements.

Soft skills

During internship candidates are required to complete some courses related to soft skills as mandatory requirement on the basis of domains .These are the soft skill courses I was assigned during this particular week

1.Communications and soft skills

Communication is the foundation of all human relationship, it facilitates the spread of knowledge and forms relationships between people. For this reason organizations are entirely reliant on it. This course helped me in becoming more confident and effective when speaking ,writing and listening.

2.Better virtual meetings how to lead effective meetings

This course gives us insights on how to run meetings effectively, how to take helpful meeting minutes during meetings, what to do before during and after each meeting, when to schedule meetings (best days and times) to minimize the number of interruptions, how to schedule meetings for team members in different time zones (using a free tool) etc. This course helped me to learn simple concepts that helps in improving our meeting facilitation and communication skills, and become a better team leader through easy meeting tactics.

3.Active Listening Master Class

Active listening is the key to increasing leadership equity, unlocking employee retention, increasing workplace efficiency .This course helps us to transform our leadership, increase workplace efficiency, understand the negative impact of multitasking while listening etc.

4.Effective email communications enhancing your voice at work

Email is a primary method of communication; however, many are not aware of appropriate skills and strategies for communicating effectively. The particular course will help in building knowledge and understanding of email communication with the intent of improving both clarity and effectiveness. The course includes topics like understanding the nature of email communication , considering the nature of an intended audience etc.

5. Communicate for business ,write email, close the loop

Clear communication isn't always as easy as it sounds. This course will help us to express our ideas in traditional genres as well as digitally. I have learnt how to communicate to create lasting change in our organization: successfully engage the people who have to respond to the initiative.
This course will give insights on choosing words for their precision and ability to engage our target audience , tailoring our emails based on our purpose and recipient etc.

3.Week 4 -6 : February 21- March 7

Java Programming

This week on java programming focus mainly on basic java concepts,object oriented programming in java, advanced java concepts ,collections in java etc. I was assigned with a 31 hours java course which helped me in enhancing my java skills and helped me in learning advanced java concepts .I have used Eclipse platform to do various hands on exercises. Java is one of the most popular programming languages. Java offers both object-oriented and functional programming features. The particular course helped me in working with both these concepts. In that I was new to functional programming. And by the end of course I solved many problems in functional programming with Java. During the initial phase I have come across basic java concepts like conditional statements , primitive types ,loops etc. Then the next stage was mainly on object oriented programming. I have implemented the concepts like inheritance , abstraction etc in java.

I have also come across object composition concept in Java which will help in programming .The next phase was mainly on collections . In that I have learnt about implementing collections like List interface, Set interface ,Map interface etc. I have also implemented lists like arrayList,sets like HashSet,LinkedHashSet,TreeSet etc. This course also helped me in implementing maps like HashMap,LinkedHashMap and Tree Map.I also gained insights on queue interface and implemented priority queue in java. The next phase was based on functional programming and multithreading. In functional programming I had worked with lambda expressions and stream operations like sort,distinct,filter and map. This course also provides an overview of how these functions are written in Java. In multithreading I have learnt about creating thread,placing priority requests in thread,thread utility methods,executor service etc. The next phase was mainly based on exception handling .In that I have come across basics of exception handling ,creating our own exception,throwing a checked exception etc. This course also provides an overview of spring framework

4.Week 6-8: March 7-March 14

Data warehouse fundamentals

This week mainly focus on data warehouse fundamentals and Unix commands.Here I have completed a course on data warehousing and completed the hands on exercises on Unix.I have come across topics like data warehousing architecture ,ETL,dimension tables,fact table,star schema,snow flake etc in data ware house architecture. During Unix phase I have worked on areas including directory creation,copying file,grep ,tail command,redirect commad,pattern printing etc.

5.Week 8: March 14-March 21

Python programming

This week was subjected to develop python programming skills. I have completed a course on python programming where I refreshed my python skills. I was also required to complete hands on exercises in python which is mainly based on python data types, functions ,collections in python, operators ,modules ,file handling etc. During python collection stage I have worked on collections like list , tuple and dictionary . Comparison between java collections and python collections was done at this age. Creating modules in python will allow large programs to break down into small chunks of code.It also helps in program modifications.

6.Week 9: March 21-present

Big data and cloud fundamentals

Since I am in the initial stage of big data phase , I was assigned with a course on Big data with spark. Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching, and optimized query execution for fast analytic queries against data of any size.I have set up Ubuntu using oracle VM virtual box. In that I have installed apache spark.

Currently I am running spark stand alone with the help of jupyter notebook.During the starting phase I have worked on a case study with Walmart stock dataset. Firstly I have learnt about data frame basics.In that stage I worked with group by,order by,head,collect,describe etc. I also had hands on experience on date and timestamps in spark which is helpful in data analysis.Then I moved on to machine learning .In machine learning module I have started with linear regression.

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

Data is the lifeblood of all business. Data-driven decisions increasingly make the difference between keeping up with competition or falling further behind. Machine learning can be the key to unlocking the value of corporate and customer data and enacting decisions that keep a company ahead of the competition.Many studies have shown that data driven decision are more effective and more efficient than human-generated decisions. Big Data allows organizations to detect trends, and spot patterns that can be used for future benefit. It can help to detect which customers are likely to buy products, or help to optimize marketing campaigns by identifying which advertisement strategies have the highest return on investment. It is easy to see that organizations that 'know' more than their competitors, will outperform their peers in the long run. As a result big data is one of the best technology to learn.The tool with which I have been working is Pyspark. PySpark is very well used in Data Science and Machine Learning community as there are many widely used data science libraries written in Python including NumPy, TensorFlow also used due to their efficient processing of large datasets. PySpark has been used by many organizations like Walmart, Trivago, Sanofi, Runtastic, and many more.It can work on a huge sets of data as compared to pandas.The major algorithms with which I have been worked on is linear regression,logistic regression,k means clustering , collaborative filtering , random forest classifier and decision tree classifier .The first case study that is predicting the crew members for a particular ship company mainly deals with linear regression.The second case study mainly focus logistic regression model to predict whether the particular customer will churn or not.The third case study focus more on comparison between algorithms like decision trees and random forest classifier based on a case study to predict whether the particular university is private or not .The fourth case study mainly deals with the prediction of the importance of each feature in a dataset using random forest classifier.The case study with which I have worked is analyzing dog food spoilage using random forest classifier.The next case study with which I have worked is building a movie recommendation system using collaborative filtering .The sixth case study mainly focus on analyzing hacking activity using k means clustering .The primary aim of case study is to investigate the presence of a particular hacker in the security attack.The next case study mainly focus on natural language processing.The primary aim of the case study is to build a spam filter using nlp techniques. I have also worked with various NLP tools . The future scope of the project involves performing case studies with various other algorithms like support vector machines ,k nearest neighbors etc.

APPENDIX A

SOURCE CODE

1.Predicting the number of crew members for a ship company

```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('ship').get
OrCreate()
df=spark.read.csv('cr.csv',inferSchema=True,hea
der=True) df.printSchema() df.describe() for s in
df.head(5):

   print(s)

df.groupBy('Cruise_line').count().sh
ow() from pyspark.ml.feature
import StringIndexer

indexed=StringIndexer(inputCol='Cruise_line',outputCol='c
ategories') indexednew=indexed.fit(df).transform(df)
type(indexednew)

pyspark.sql.dataframe.DataFrame
indexednew.show()

from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import
VectorAssembler

assembler=VectorAssembler(inputCols=['Age','Tonnage','passengers','length','cabi
ns','passen ger_density','categories'],outputCol='features') indexednew.columns

['Ship_name',
'Cruise_line',

 'Age',
 'Tonnage',

 'passengers',

 'length',

 'cabins',

 'passenger_density',
 'crew',

 'categories']

output=assembler.transform(indexednew)
output.select('features','crew','Ship_name').sh
ow()
finalData=output.select(['features','crew'])
traindata,testdata=finalData.randomSplit([0.7,
0.3]) from pyspark.ml.regression import
LinearRegression
lr=LinearRegression(labelCol='crew')
trainedModel=lr.fit(traindata)
testModel=trainedModel.evaluate(testdata)
type(trainedModel)

pyspark.ml.regression.LinearRegressionModel
testModel.r2

0.8547645832700974
testModel.rootMeanSquaredError
1.4223068077716097

from          pyspark.sql.functions          import          corr
df.select(corr('crew','passengers')).show()

+--------------------+

|corr(crew, passengers)|

+--------------------+
|    0.9152341306065384|
```

2.Churn Prediction Using Logistic Regression

```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('churn').getOrCreate()
df=spark.read.csv('customer_churn.csv',inferSchema=True,hea
der=True) df.printSchema() df.columns

from pyspark.ml.feature import VectorAssembler

assembler=VectorAssembler(inputCols=['Age','Total_Purchase','Account_Manager','Years','
Num_Sites'],outputCol='features')
output=assembler.transform(df)
finaldata=output.select('features','Churn')
finaldata.show()

traind,testd=finaldata.randomSplit([0.7,0.3])
from pyspark.ml.classification import
LogisticRegression
lrchurn=LogisticRegression(labelCol='Churn')
fittedmodel=lrchurn.fit(traind)
trainingsummary=fittedmodel.summary
trainingsummary.predictions.describe().show()

from pyspark.ml.evaluation import
BinaryClassificationEvaluator
predlabels=fittedmodel.evaluate(testd)
predlabels.predictions.show()

churnevaluator=BinaryClassificationEvaluator(labelCol='Churn',rawPredictionCol='
predictio n')

auc=churnevaluator.evaluate(predlabels.predi
ctions) auc

0.7402717127607957
finalmodel=lrchurn.fit(finaldata)
```

```
newcustomers=spark.read.csv('new.csv',inferSchema=True,header=True)
newcustomers.printSchema()

testnewcustomers=assembler.transform(newcustomers)
testnewcustomers.show()

finalresults=finalmodel.transform(testnewcustomers)
finalresults.columns

finalresults.select('Names','prediction').show()
```

3.Private/Public University Comparison for various models

```
From pyspark.sql import SparkSession

spark=SparkSession.builder.appName('college').getOrCr
eate() data
=spark.read.csv('college.csv',inferSchema=True,header=
True) data.printSchema()
data.head(1)

from pyspark.ml.feature import VectorAssembler
data.columns

from pyspark.ml.feature import VectorAssembler
assembler=VectorAssembler(inputCols=['Apps',

 'Accept',

 'Enroll',

 'Top10perc',

 'Top25perc',

 'F_Undergrad',

 'P_Undergrad',

 'Outstate',
 'Room_Board',

 'Books',

 'Personal',

 'PhD',

 'Terminal',

 'S_F_Ratio',

 'perc_alumni',

 'Expend',

 'Grad_Rate'],outputCol='features')
output=assembler.transform(data)
from pyspark.ml.feature import
StringIndexer

indexer=StringIndexer(inputCol='Private',outputCol='privatein
dex') outputfixed=indexer.fit(output).transform(output)
outputfixed.collect()[1]

finaldata=outputfixed.select('features','privateindex')
finaldata.show()

traindata,testdata=finaldata.randomSplit([0.7,0.3])

from pyspark.ml.classification import
(DecisionTreeClassifier,RandomForestClassifier,GBTClassifier)
from pyspark.ml import Pipeline

dtc=DecisionTreeClassifier(labelCol='privateindex',featuresCol='features')

rfc=RandomForestClassifier(numTrees=150,labelCol='privateindex',featuresCol=
'features') gbt=GBTClassifier(labelCol='privateindex',featuresCol='features')
dtcmodel=dtc.fit(traindata) rfcmodel=rfc.fit(traindata)
gbtmodel=gbt.fit(traindata) dtcpreds=dtcmodel.transform(testdata) type(dtcpreds)
```

```
pyspark.sql.dataframe.DataFrame
rfcpreds=rfcmodel.transform(testdata)
gbtpreds=gbtmodel.transform(testdata)

from pyspark.ml.evaluation import
BinaryClassificationEvaluator
bevt=BinaryClassificationEvaluator(labelCol='privatein
dex') print(bevt.evaluate(dtcpreds))
print(bevt.evaluate(gbtpreds)) 0.9731813610459268

print(bevt.evaluate(rfcpreds))
0.9913677505866575
gbtpreds.printSchema()

from pyspark.ml.evaluation import MulticlassClassificationEvaluator

acc=MulticlassClassificationEvaluator(metricName='accuracy',labelCol='privateindex')
accuracyevaluator=acc.evaluate(rfcpreds)

f1evaluator=MulticlassClassificationEvaluator(metricName='f1',labelCol='privateindex')
f1evaluator.evaluate(rfcpreds)
```

4.Analyzing dog food spoilage using random forest model

```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('treeconsult').getOrCreate()
data=spark.read.csv('dog_food.csv',inferSchema=True,header=True)
data.head(1)
from pyspark.ml.feature import VectorAssembler

assembler=VectorAssembler(inputCols=['A','B','C','D'],outputCol='features')
output=assembler.transform(data)

from pyspark.ml.classification import
RandomForestClassifier
rfc=RandomForestClassifier(labelCol='Spoiled',featuresCol=
'features') finaldata=output.select('features','Spoiled')
```

```
finaldata.show() rfcmodel=rfc.fit(finaldata)
rfcmodel.featureImportances
```

5. Implementation of movie recommendation system using collaborative filtering

```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('reco').get
OrCreate() from pyspark.ml.recommendation
import ALS from pyspark.ml.evaluation import
RegressionEvaluator

data=spark.read.csv('movielens_ratings.csv',inferSchema=True,hea
der=True) data.show() data.describe().show()

trainingdata,testdata=data.randomSplit([0.8,0.2])

als=ALS(maxIter=5,regParam=0.01,userCol='userId',ratingCol='rating',itemCol='
movieId') model=als.fit(trainingdata) predictions=model.transform(testdata)
predictions.show()

evaluator=RegressionEvaluator(labelCol='rating',predictionCol='prediction',metricNa
me='rm se')

rmse=evaluator.evaluate(predict
ions) rmse

singleuser=testdata.filter(testdata['userId']==6).select('movieId','userId')
singleuser.show()

recommendations=model.transform(singleuser)

recommendations.orderBy('prediction',ascending=False).show()
```

6. Analyzing the patterns in hacking using K Means Clustering

```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('hack1').getOrCreate()
data=spark.read.csv('hack_data.csv',inferSchema=True,header=True)
data.head(5)
```

```
from pyspark.ml.clustering import
KMeans from pyspark.ml.feature
import VectorAssembler data.columns

feature_columns=['Session_Connection_Time',

 'Bytes Transferred',

 'Kali_Trace_Used',

 'Servers_Corrupted',

 'Pages_Corrupted',

 'WPM_Typing_Speed']

assembler=VectorAssembler(inputCols=feature_columns,outputCol=
'features') finaldata=assembler.transform(data)
finaldata.printSchema()

from pyspark.ml.feature import StandardScaler

scaler=StandardScaler(inputCol='features',outputCol='scale
dfeatures') scalermodel=scaler.fit(finaldata)
type(scalermodel)

clustereddata=scalermodel.transform(f
inaldata) clustereddata.head()
clustereddata.printSchema()

kmeans2=KMeans(featuresCol='scaledfeatures',k=2)
kmeans3=KMeans(featuresCol='scaledfeatures',k=3)
type(kmeans2)

kmeans2model=kmeans2.fit(clustereddata)
kmeans3model=kmeans3.fit(clustereddata)

kmeans2model.transform(clustereddata).groupBy('prediction').count().show()
kmeans3model.transform(clustereddata).groupBy('prediction').count().show()
```

7. Spam filter using Natural Language processing

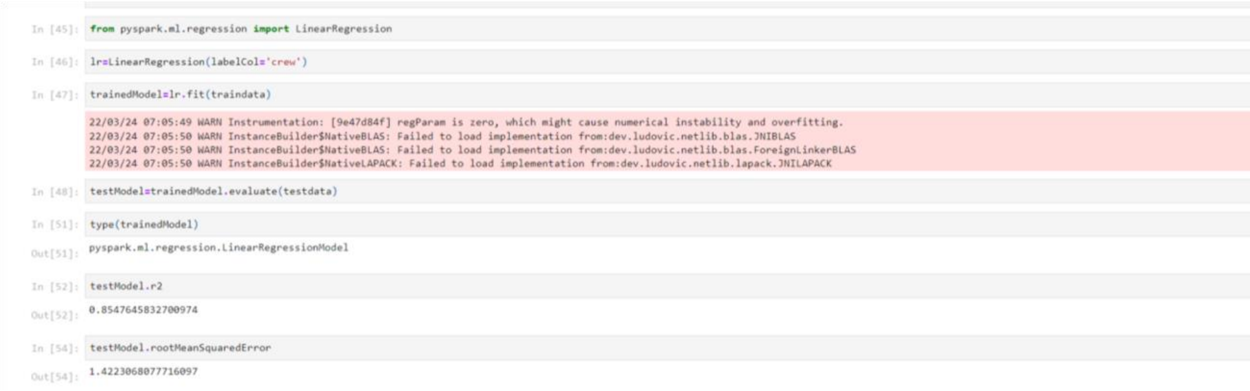
```
from pyspark.sql import SparkSession

spark=SparkSession.builder.appName('spamdetection').get
OrCreate()
data=spark.read.csv('spam',inferSchema=True,sep='\t')
df.show()

data=data.withColumnRenamed('_c0','class').withColumnRenamed('_c1','message')
data.show()

from pyspark.sql.functions import length

data=data.withColumn('length',length(data['message']))
data.show()

data.groupBy('class').mean().show()

from pyspark.ml.feature import
Tokenizer,StopWordsRemover,CountVectorizer,IDF,StringIndexer
tokenizer=Tokenizer(inputCol='message',outputCol='tokens_text')
stopwords=StopWordsRemover(inputCol='tokens_text',outputCol='stop_token')
countvec=CountVectorizer(inputCol='stop_token',outputCol='c_vec')
idf=IDF(inputCol='c_vec',outputCol='tf_idf')

hamspamtonumeric=StringIndexer(inputCol='class',outputCol='l
abel') from pyspark.ml.feature import VectorAssembler

cleanup=VectorAssembler(inputCols=['tf_idf','length'],outputCol=
'features') from pyspark.ml.classification import NaiveBayes
nb=NaiveBayes()

from pyspark.ml import Pipeline

datapreppipe=Pipeline(stages=[hamspamtonumeric,tokenizer,stopwords,countvec,idf,cleanup
])

cleaner=datapreppipe.fit(data)
cleanerdata=cleaner.transform(data)
cleanerdata.collect()[0]
```

```
cleanerdata=cleanerdata.select('label','features')
cleanerdata.show()

training,test=cleanerdata.randomSplit([0.7,0.3])
spamdetector=nb.fit(training)
testresults=spamdetector.transform(test)
testresults.show()

from pyspark.ml.evaluation import
MulticlassClassificationEvaluator
acceval=MulticlassClassificationEvaluator()
acc=acceval.evaluate(testresults) acc
```

APPENDIX B

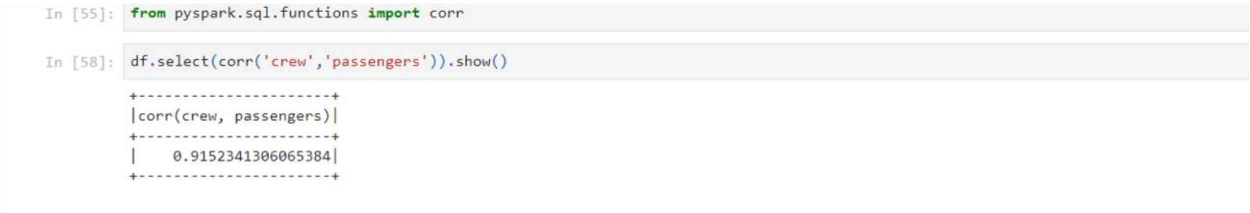
SCREENSHOTS

1. Predicting the number of crew members for a ship company



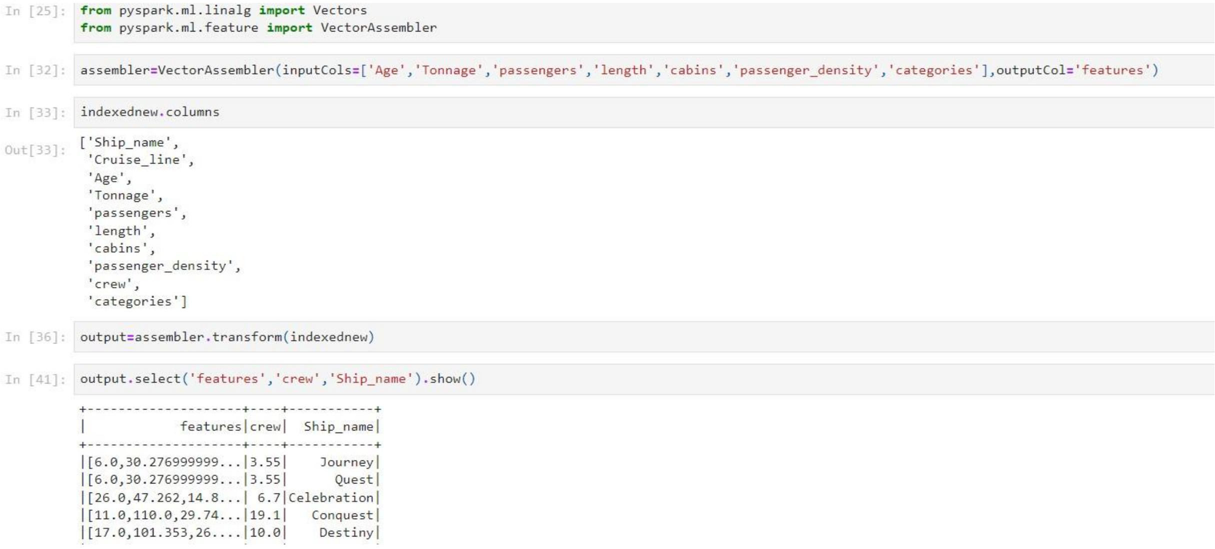
```
In [45]: from pyspark.ml.regression import LinearRegression

In [46]: lr=LinearRegression(labelCol='crew')

In [47]: trainedModel=lr.fit(traindata)

         22/03/24 07:05:49 WARN Instrumentation: [9e47d84f] regParam is zero, which might cause numerical instability and overfitting.
         22/03/24 07:05:50 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
         22/03/24 07:05:50 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:dev.ludovic.netlib.blas.ForeignLinkerBLAS
         22/03/24 07:05:50 WARN InstanceBuilder$NativeLAPACK: Failed to load implementation from:dev.ludovic.netlib.lapack.JNILAPACK

In [48]: testModel=trainedModel.evaluate(testdata)

In [51]: type(trainedModel)

Out[51]: pyspark.ml.regression.LinearRegressionModel

In [52]: testModel.r2

Out[52]: 0.8547645832700974

In [54]: testModel.rootMeanSquaredError

Out[54]: 1.4223068077716097
```

Importing linear regression .Performing fitting.



```
In [55]: from pyspark.sql.functions import corr

In [58]: df.select(corr('crew','passengers')).show()

         +----------------------+
         |corr(crew, passengers)|
         +----------------------+
         |    0.9152341306065384|
         +----------------------+
```

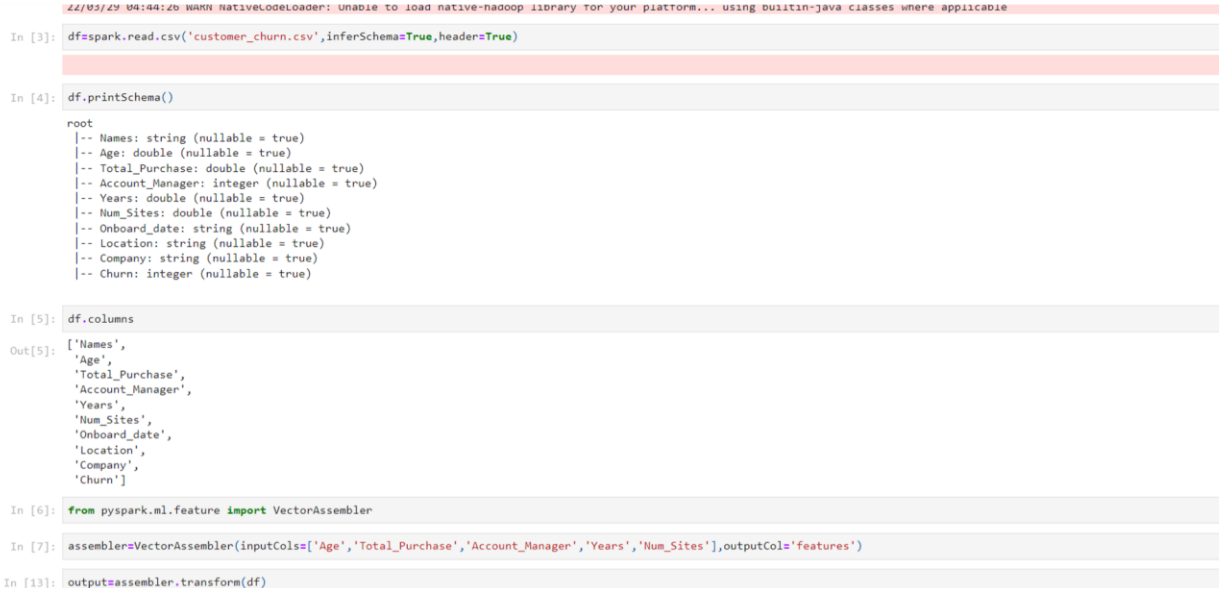
Correlation between 2 features crew and passengers



```
In [25]: from pyspark.ml.linalg import Vectors
         from pyspark.ml.feature import VectorAssembler

In [32]: assembler=VectorAssembler(inputCols=['Age','Tonnage','passengers','length','cabins','passenger_density','categories'],outputCol='features')

In [33]: indexednew.columns

Out[33]: ['Ship_name',
          'Cruise_line',
          'Age',
          'Tonnage',
          'passengers',
          'length',
          'cabins',
          'passenger_density',
          'crew',
          'categories']

In [36]: output=assembler.transform(indexednew)

In [41]: output.select('features','crew','Ship_name').show()

         +--------------------+----+-----------+
         |            features|crew|  Ship_name|
         +--------------------+----+-----------+
         |[6.0,30.276999999...|3.55|    Journey|
         |[6.0,30.276999999...|3.55|      Quest|
         |[26.0,47.262,14.8...| 6.7|Celebration|
         |[11.0,110.0,29.74...|19.1|   Conquest|
         |[17.0,101.353,26....|10.0|    Destiny|
```

Importing vector and vector assembler .Grouping the necessary features.Transforming the dataset using assembler .

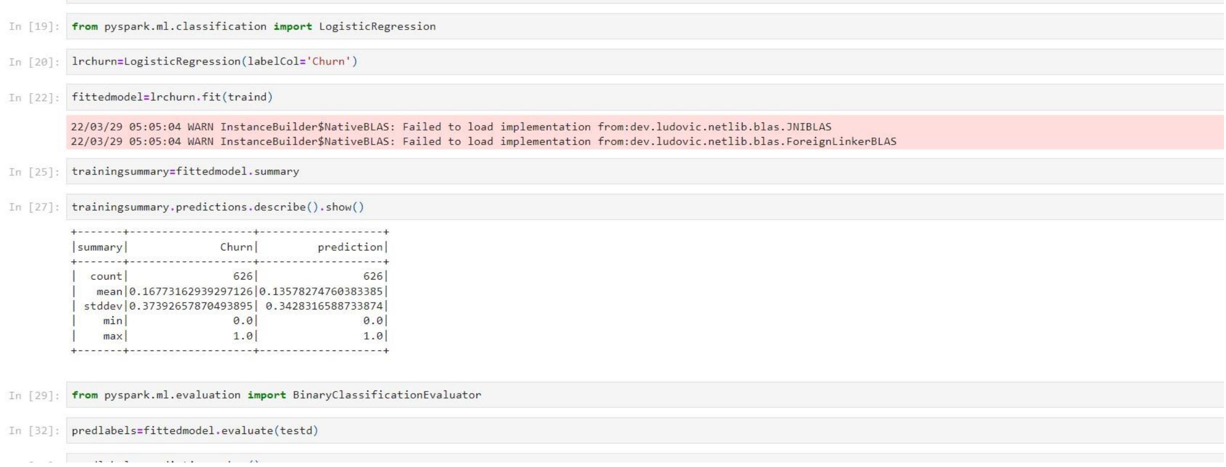
2.Churn prediction using logistic regression



```
22/03/29 04:44:26 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

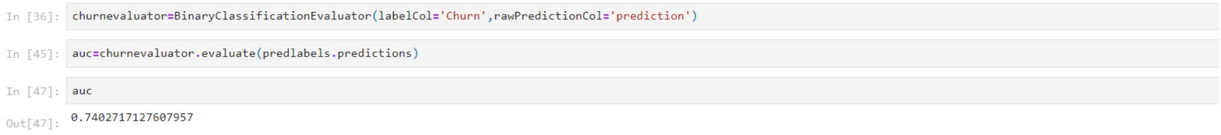
```
In [3]:  df=spark.read.csv('customer_churn.csv',inferSchema=True,header=True)

In [4]:  df.printSchema()

         root
          |-- Names: string (nullable = true)
          |-- Age: double (nullable = true)
          |-- Total_Purchase: double (nullable = true)
          |-- Account_Manager: integer (nullable = true)
          |-- Years: double (nullable = true)
          |-- Num_Sites: double (nullable = true)
          |-- Onboard_date: string (nullable = true)
          |-- Location: string (nullable = true)
          |-- Company: string (nullable = true)
          |-- Churn: integer (nullable = true)

In [5]:  df.columns

Out[5]:  ['Names',
          'Age',
          'Total_Purchase',
          'Account_Manager',
          'Years',
          'Num_Sites',
          'Onboard_date',
          'Location',
          'Company',
          'Churn']

In [6]:  from pyspark.ml.feature import VectorAssembler

In [7]:  assembler=VectorAssembler(inputCols=['Age','Total_Purchase','Account_Manager','Years','Num_Sites'],outputCol='features')

In [13]: output=assembler.transform(df)
```

Reading the entire dataset .Printing the schema and importing vector assembler to group the features useful for analysis.



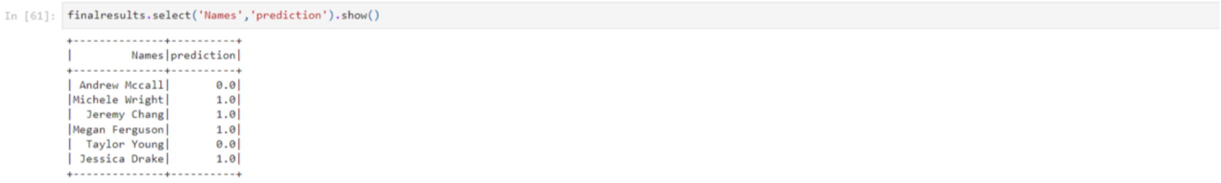
```
In [19]: from pyspark.ml.classification import LogisticRegression

In [20]: lrchurn=LogisticRegression(labelCol='Churn')

In [22]: fittedmodel=lrchurn.fit(traind)

         22/03/29 05:05:04 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:dev.ludovic.netlib.blas.JNIBLAS
         22/03/29 05:05:04 WARN InstanceBuilder$NativeBLAS: Failed to load implementation from:dev.ludovic.netlib.blas.ForeignLinkerBLAS

In [25]: trainingsummary=fittedmodel.summary

In [27]: trainingsummary.predictions.describe().show()

         +-------+-------------------+-------------------+
         |summary|              Churn|         prediction|
         +-------+-------------------+-------------------+
         |  count|                626|                626|
         |   mean|0.16773162939297126|0.13578274760383385|
         | stddev|0.37392657870493895| 0.3428316588733874|
         |    min|                0.0|                0.0|
         |    max|                1.0|                1.0|
         +-------+-------------------+-------------------+

In [29]: from pyspark.ml.evaluation import BinaryClassificationEvaluator

In [32]: predlabels=fittedmodel.evaluate(testd)
```

Importing logistic regression and binary classification evaluator .Performing fitting on the dataset.



```
In [36]: churnevaluator=BinaryClassificationEvaluator(labelCol='Churn',rawPredictionCol='prediction')

In [45]: auc=churnevaluator.evaluate(predlabels.predictions)

In [47]: auc

Out[47]: 0.7402717127607957
```

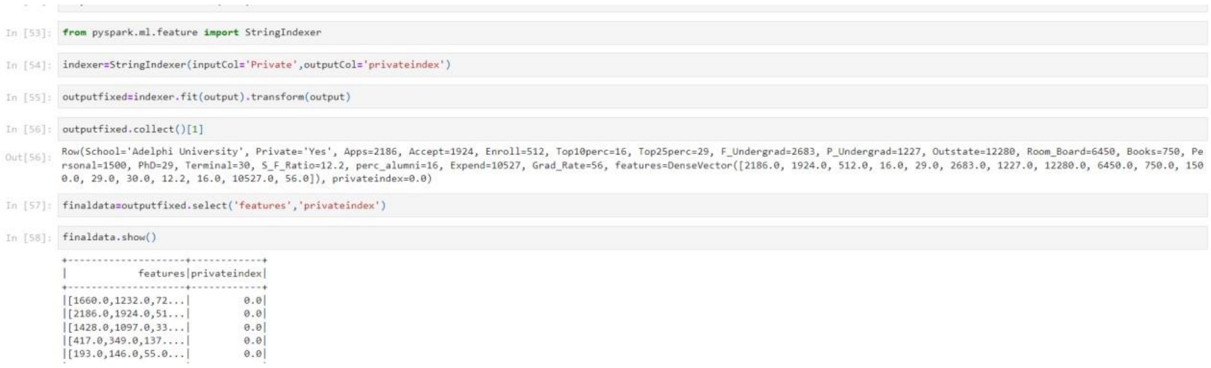
Evaluating area under the curve



```
In [61]: finalresults.select('Names','prediction').show()

+--------------+----------+
|         Names|prediction|
+--------------+----------+
| Andrew Mccall|       0.0|
|Michele Wright|       1.0|
|  Jeremy Chang|       1.0|
|Megan Ferguson|       1.0|
|  Taylor Young|       0.0|
| Jessica Drake|       1.0|
+--------------+----------+
```

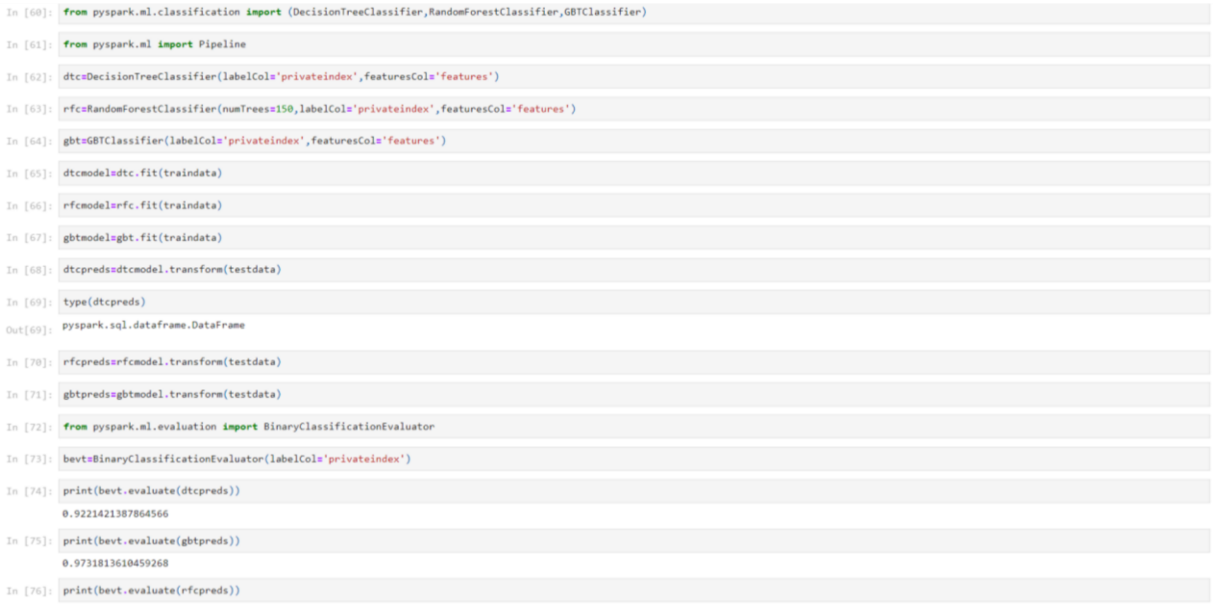
predictions on new dataset.

3.Private / Public university prediction and comparison of various models



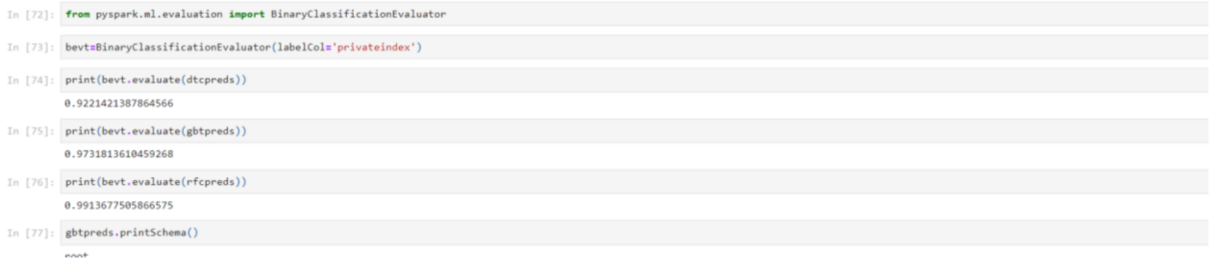
```
In [53]: from pyspark.ml.feature import StringIndexer

In [54]: indexer=StringIndexer(inputCol='Private',outputCol='privateindex')

In [55]: outputfixed=indexer.fit(output).transform(output)

In [56]: outputfixed.collect()[1]

Out[56]: Row(School='Adelphi University', Private='Yes', Apps=2186, Accept=1924, Enroll=512, Top10perc=16, Top25perc=29, F_Undergrad=2683, P_Undergrad=1227, Outstate=12280, Room_Board=6450, Books=750, Pe
         rsonal=1500, PhD=29, Terminal=30, S_F_Ratio=12.2, perc_alumni=16, Expend=10527, Grad_Rate=56, features=DenseVector([2186.0, 1924.0, 512.0, 16.0, 29.0, 2683.0, 1227.0, 12280.0, 6450.0, 750.0, 150
         0.0, 29.0, 30.0, 12.2, 16.0, 10527.0, 56.0]), privateindex=0.0)

In [57]: finaldata=outputfixed.select('features','privateindex')

In [58]: finaldata.show()

+--------------------+------------+
|            features|privateindex|
+--------------------+------------+
|[1660.0,1232.0,72...|         0.0|
|[2186.0,1924.0,51...|         0.0|
|[1428.0,1097.0,33...|         0.0|
|[417.0,349.0,137....|         0.0|
|[193.0,146.0,55.0...|         0.0|
```

Importing string indexer to convert the categorical column to numerical



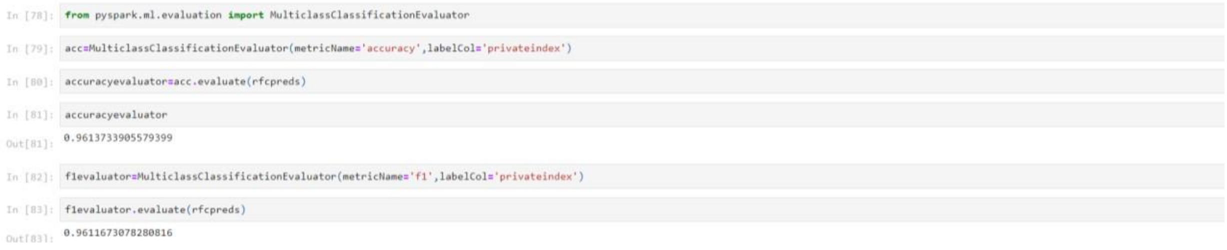
```
In [60]: from pyspark.ml.classification import (DecisionTreeClassifier,RandomForestClassifier,GBTClassifier)

In [61]: from pyspark.ml import Pipeline

In [62]: dtc=DecisionTreeClassifier(labelCol='privateindex',featuresCol='features')

In [63]: rfc=RandomForestClassifier(numTrees=150,labelCol='privateindex',featuresCol='features')

In [64]: gbt=GBTClassifier(labelCol='privateindex',featuresCol='features')

In [65]: dtcmodel=dtc.fit(traindata)

In [66]: rfcmodel=rfc.fit(traindata)

In [67]: gbtmodel=gbt.fit(traindata)

In [68]: dtcpreds=dtcmodel.transform(testdata)

In [69]: type(dtcpreds)

Out[69]: pyspark.sql.dataframe.DataFrame

In [70]: rfcpreds=rfcmodel.transform(testdata)

In [71]: gbtpreds=gbtmodel.transform(testdata)

In [72]: from pyspark.ml.evaluation import BinaryClassificationEvaluator

In [73]: bevt=BinaryClassificationEvaluator(labelCol='privateindex')

In [74]: print(bevt.evaluate(dtcpreds))
         0.9221421387864566

In [75]: print(bevt.evaluate(gbtpreds))
         0.9731813610459268

In [76]: print(bevt.evaluate(rfcpreds))
```

Importing decision tree classifier ,random forest classifier and gbt classifier



```
In [72]: from pyspark.ml.evaluation import BinaryClassificationEvaluator

In [73]: bevt=BinaryClassificationEvaluator(labelCol='privateindex')

In [74]: print(bevt.evaluate(dtcpreds))
         0.9221421387864566

In [75]: print(bevt.evaluate(gbtpreds))
         0.9731813610459268

In [76]: print(bevt.evaluate(rfcpreds))
         0.9913677505866575

In [77]: gbtpreds.printSchema()
         root
```

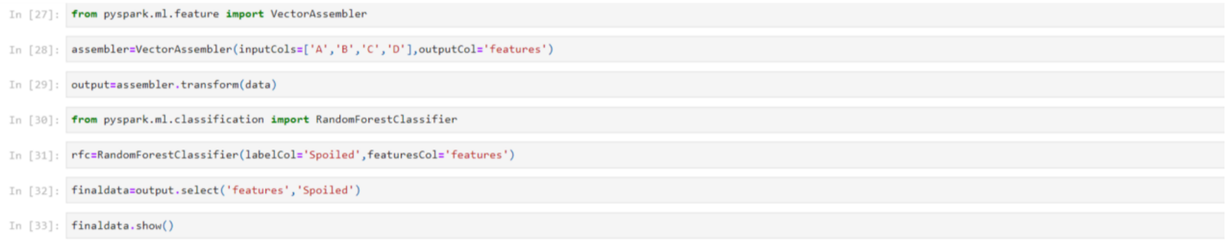
Importing binary classification evaluator measuring area under the curve for random forest classifier,decision tree classifier and gradient boost trees.



```
In [78]: from pyspark.ml.evaluation import MulticlassClassificationEvaluator

In [79]: acc=MulticlassClassificationEvaluator(metricName='accuracy',labelCol='privateindex')

In [80]: accuracyevaluator=acc.evaluate(rfcpreds)

In [81]: accuracyevaluator
Out[81]: 0.9613733905579399

In [82]: f1evaluator=MulticlassClassificationEvaluator(metricName='f1',labelCol='privateindex')

In [83]: f1evaluator.evaluate(rfcpreds)
Out[83]: 0.9611673078280816
```

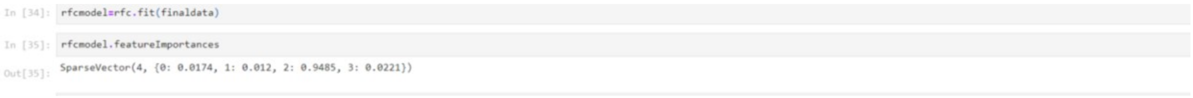
Importing multiclass classification evaluator and finding the accuracy,fi score for random forest classifier.

4. Analyzing dog food spoilage using Random Forest Model



```
In [27]: from pyspark.ml.feature import VectorAssembler

In [28]: assembler=VectorAssembler(inputCols=['A','B','C','D'],outputCol='features')

In [29]: output=assembler.transform(data)

In [30]: from pyspark.ml.classification import RandomForestClassifier

In [31]: rfc=RandomForestClassifier(labelCol='Spoiled',featuresCol='features')

In [32]: finaldata=output.select('features','Spoiled')

In [33]: finaldata.show()
```

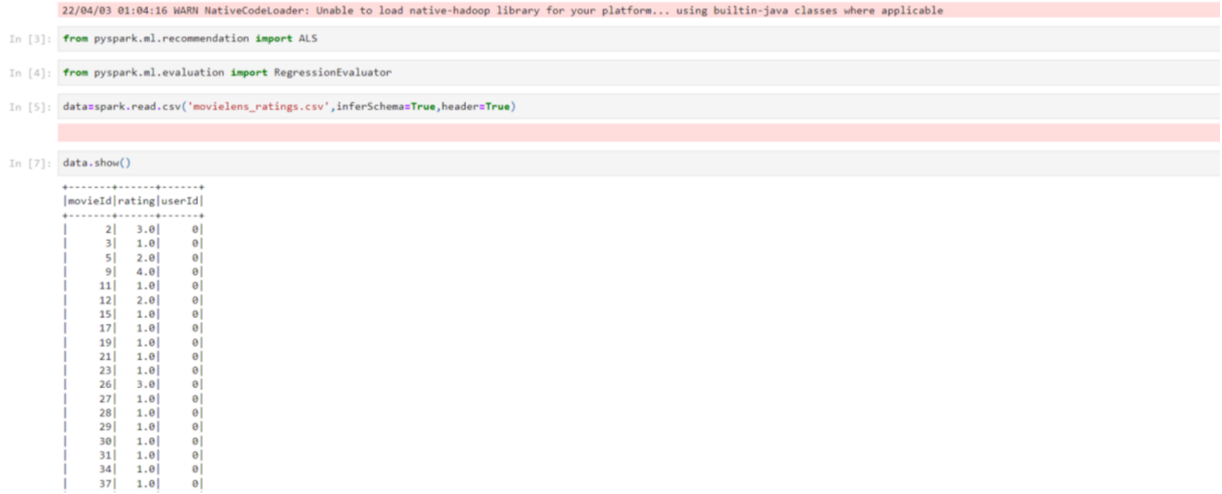
Importing vector assembler to convert the features into a vector form.We have used random forest classifier since the label is discrete .



```
In [34]: rfcmodel=rfc.fit(finaldata)

In [35]: rfcmodel.featureImportances
Out[35]: SparseVector(4, {0: 0.0174, 1: 0.012, 2: 0.9485, 3: 0.0221})
```

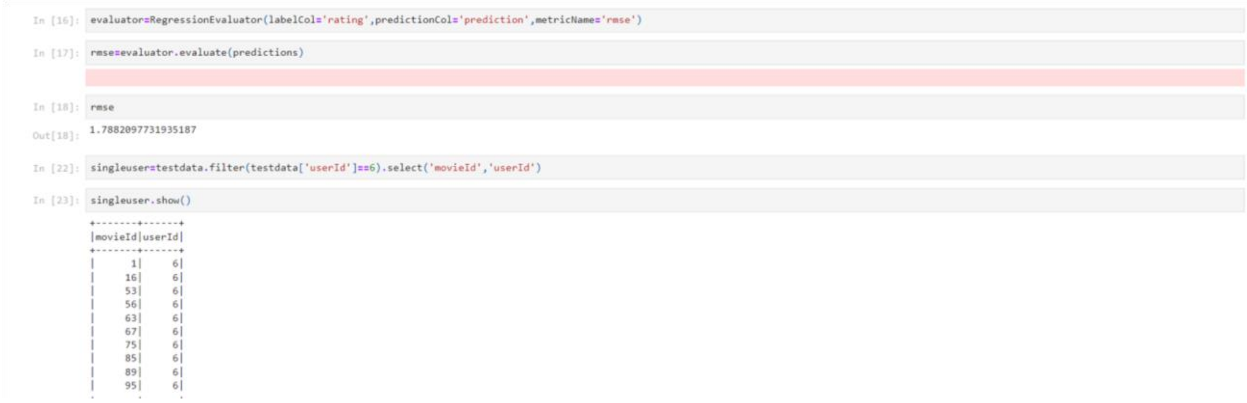
Evaluating the importance of every preservative. The final conclusion is that chemical C is of primary importance.It is the chemical which contributes to a great extent in food spoilage

5 .Implementation of movie recommendation system using collaborative filtering



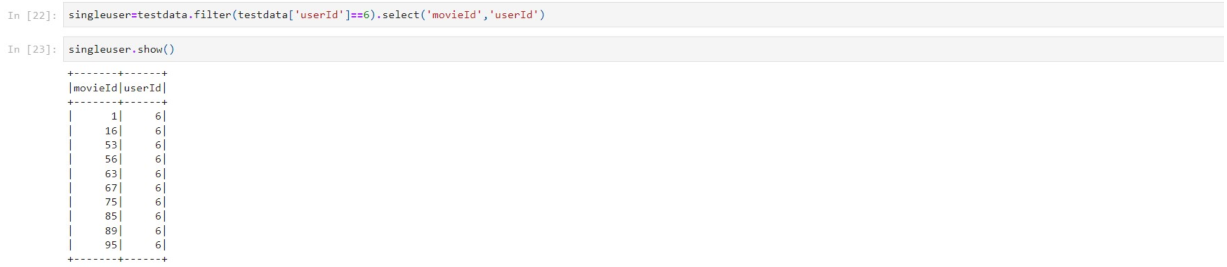
```
22/04/03 01:04:16 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable

In [3]: from pyspark.ml.recommendation import ALS

In [4]: from pyspark.ml.evaluation import RegressionEvaluator

In [5]: data=spark.read.csv('movielens_ratings.csv',inferSchema=True,header=True)

In [7]: data.show()

+-------+------+------+
|movieId|rating|userId|
+-------+------+------+
|      2|   3.0|     0|
|      3|   1.0|     0|
|      5|   2.0|     0|
|      9|   4.0|     0|
|     11|   1.0|     0|
|     12|   2.0|     0|
|     15|   1.0|     0|
|     17|   1.0|     0|
|     19|   1.0|     0|
|     21|   1.0|     0|
|     23|   1.0|     0|
|     26|   3.0|     0|
|     27|   1.0|     0|
|     28|   1.0|     0|
|     29|   1.0|     0|
|     30|   1.0|     0|
|     31|   1.0|     0|
|     34|   1.0|     0|
|     37|   1.0|     0|
```

Importing ALS algorithm and regression evaluator.Reading the csv file



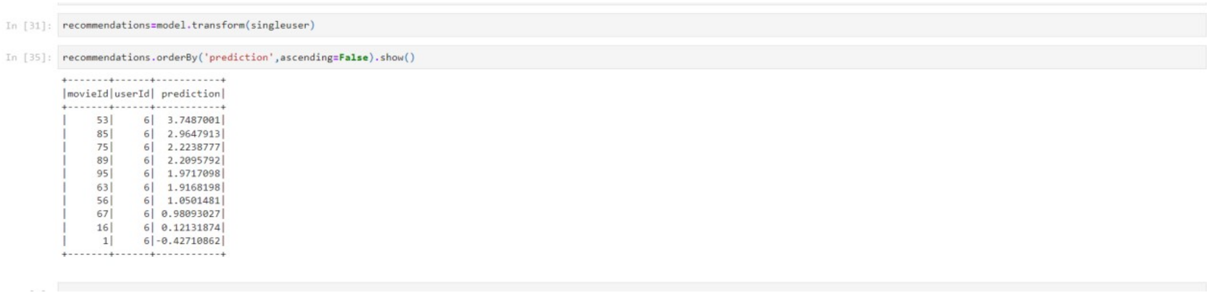
```
In [16]: evaluator=RegressionEvaluator(labelCol='rating',predictionCol='prediction',metricName='rmse')

In [17]: rmse=evaluator.evaluate(predictions)

In [18]: rmse

Out[18]: 1.7882097731935187

In [22]: singleuser=testdata.filter(testdata['userId']==6).select('movieId','userId')

In [23]: singleuser.show()

+-------+------+
|movieId|userId|
+-------+------+
|      1|     6|
|     16|     6|
|     53|     6|
|     56|     6|
|     63|     6|
|     67|     6|
|     75|     6|
|     85|     6|
|     89|     6|
|     95|     6|
```

Finding root mean squared error



```
In [22]: singleuser=testdata.filter(testdata['userId']==6).select('movieId','userId')

In [23]: singleuser.show()

+-------+------+
|movieId|userId|
+-------+------+
|      1|     6|
|     16|     6|
|     53|     6|
|     56|     6|
|     63|     6|
|     67|     6|
|     75|     6|
|     85|     6|
|     89|     6|
|     95|     6|
+-------+------+
```

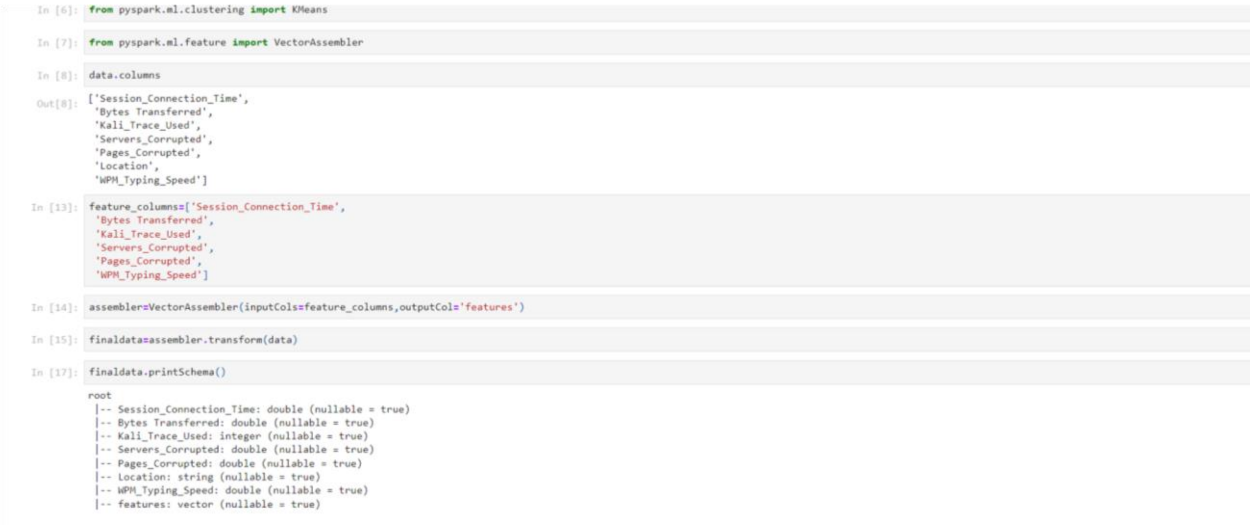
Selecting the features of a particular user



```
In [31]: recommendations=model.transform(singleuser)

In [35]: recommendations.orderBy('prediction',ascending=False).show()

+-------+------+-----------+
|movieId|userId| prediction|
+-------+------+-----------+
|     53|     6|  3.7487001|
|     85|     6|  2.9647913|
|     75|     6|  2.2238777|
|     89|     6|  2.2095792|
|     95|     6|  1.9717098|
|     63|     6|  1.9168198|
|     56|     6|  1.0501481|
|     67|     6| 0.98093027|
|     16|     6| 0.12131874|
|      1|     6|-0.42710862|
+-------+------+-----------+
```

Predictions to show by how much value the particular user may like the movie

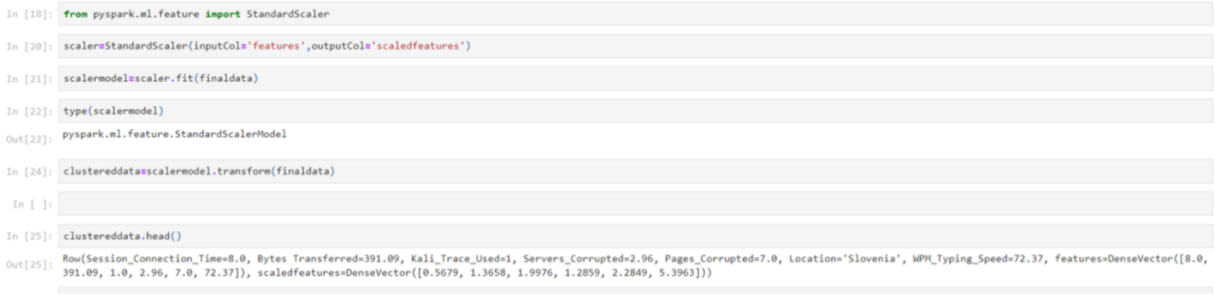
6.Hacking analysis using k means clustering



```
In [6]:  from pyspark.ml.clustering import KMeans

In [7]:  from pyspark.ml.feature import VectorAssembler

In [8]:  data.columns

Out[8]:  ['Session_Connection_Time',
          'Bytes Transferred',
          'Kali_Trace_Used',
          'Servers_Corrupted',
          'Pages_Corrupted',
          'Location',
          'WPM_Typing_Speed']

In [13]: feature_columns=['Session_Connection_Time',
          'Bytes Transferred',
          'Kali_Trace_Used',
          'Servers_Corrupted',
          'Pages_Corrupted',
          'WPM_Typing_Speed']

In [14]: assembler=VectorAssembler(inputCols=feature_columns,outputCol='features')

In [15]: finaldata=assembler.transform(data)

In [17]: finaldata.printSchema()

         root
          |-- Session_Connection_Time: double (nullable = true)
          |-- Bytes Transferred: double (nullable = true)
          |-- Kali_Trace_Used: integer (nullable = true)
          |-- Servers_Corrupted: double (nullable = true)
          |-- Pages_Corrupted: double (nullable = true)
          |-- Location: string (nullable = true)
          |-- WPM_Typing_Speed: double (nullable = true)
          |-- features: vector (nullable = true)
```

Importing vector assembler and k means .Converting useful feature columns to a vector form.

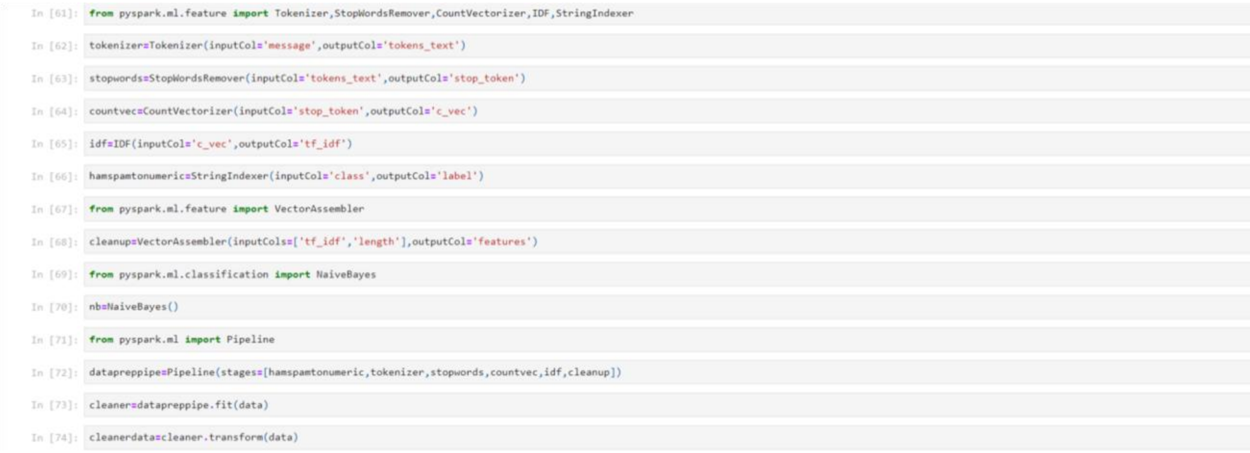
Printing the data schema.



```
In [18]: from pyspark.ml.feature import StandardScaler

In [20]: scaler=StandardScaler(inputCol='features',outputCol='scaledfeatures')

In [21]: scalermodel=scaler.fit(finaldata)

In [22]: type(scalermodel)

Out[22]: pyspark.ml.feature.StandardScalerModel

In [24]: clustereddata=scalermodel.transform(finaldata)

In [ ]:

In [25]: clustereddata.head()

Out[25]: Row(Session_Connection_Time=8.0, Bytes Transferred=391.09, Kali_Trace_Used=1, Servers_Corrupted=2.96, Pages_Corrupted=7.0, Location='Slovenia', WPM_Typing_Speed=72.37, features=DenseVector([8.0,
         391.09, 1.0, 2.96, 7.0, 72.37]), scaledfeatures=DenseVector([0.5679, 1.3658, 1.9976, 1.2859, 2.2849, 5.3963]))
```

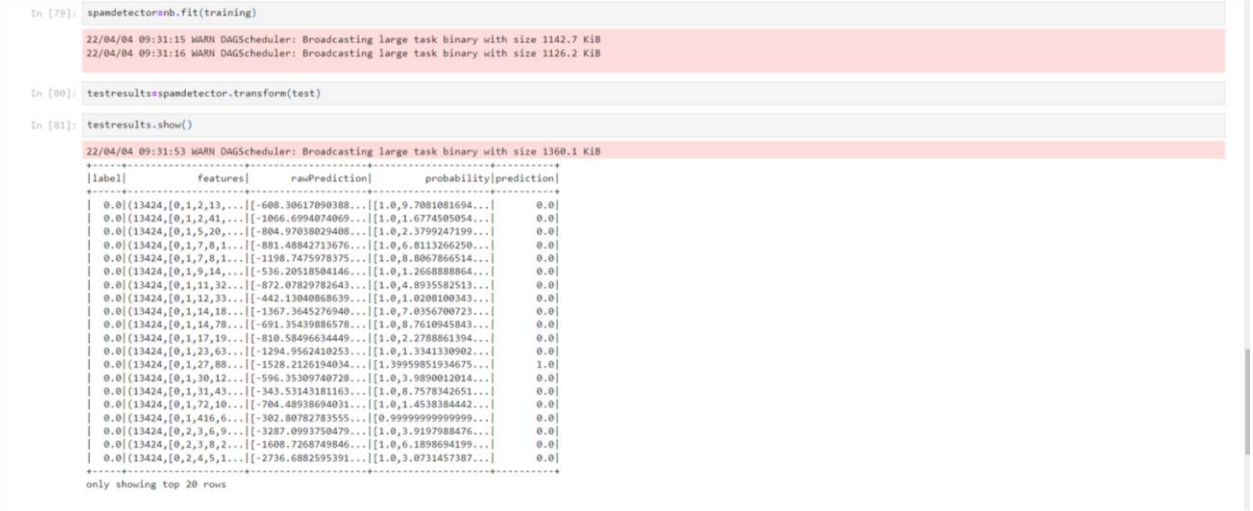
Scaling the features using standard Scaler

7.Spam filter using Natural Language Processing



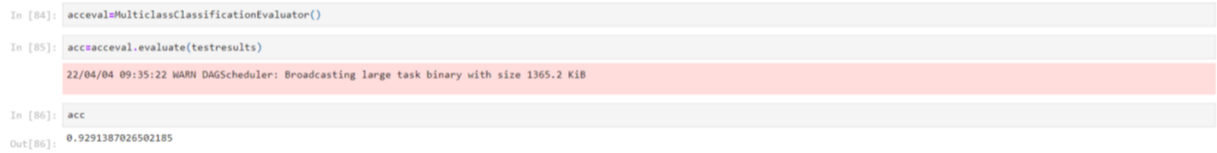
```
In [61]: from pyspark.ml.feature import Tokenizer,StopWordsRemover,CountVectorizer,IDF,StringIndexer

In [62]: tokenizer=Tokenizer(inputCol='message',outputCol='tokens_text')

In [63]: stopwords=StopWordsRemover(inputCol='tokens_text',outputCol='stop_token')

In [64]: countvec=CountVectorizer(inputCol='stop_token',outputCol='c_vec')

In [65]: idf=IDF(inputCol='c_vec',outputCol='tf_idf')

In [66]: hamspamtonumeric=StringIndexer(inputCol='class',outputCol='label')

In [67]: from pyspark.ml.feature import VectorAssembler

In [68]: cleanup=VectorAssembler(inputCols=['tf_idf','length'],outputCol='features')

In [69]: from pyspark.ml.classification import NaiveBayes

In [70]: nb=NaiveBayes()

In [71]: from pyspark.ml import Pipeline

In [72]: datapreppipe=Pipeline(stages=[hamspamtonumeric,tokenizer,stopwords,countvec,idf,cleanup])

In [73]: cleaner=datapreppipe.fit(data)

In [74]: cleanerdata=cleaner.transform(data)
```

Importing Tokenizer , Stop Words Remover ,Count Vectorizer , IDF, String Indexer and creating a pipeline so that various stages are involved.



```
In [79]: spamdetector=nb.fit(training)

22/04/04 09:31:15 WARN DAGScheduler: Broadcasting large task binary with size 1142.7 KiB
22/04/04 09:31:16 WARN DAGScheduler: Broadcasting large task binary with size 1126.2 KiB

In [80]: testresults=spamdetector.transform(test)

In [81]: testresults.show()

22/04/04 09:31:53 WARN DAGScheduler: Broadcasting large task binary with size 1360.1 KiB
+-----+--------------------+--------------------+--------------------+----------+
|label|            features|       rawPrediction|         probability|prediction|
+-----+--------------------+--------------------+--------------------+----------+
|  0.0|(13424,[0,1,2,13,...|[-608.30617090388...|[1.0,9.7081081694...|       0.0|
|  0.0|(13424,[0,1,2,41,...|[-1066.6994074069...|[1.0,1.6774505054...|       0.0|
|  0.0|(13424,[0,1,5,20,...|[-804.97038029408...|[1.0,2.3799247199...|       0.0|
|  0.0|(13424,[0,1,7,8,1...|[-881.48842713676...|[1.0,6.8113266250...|       0.0|
|  0.0|(13424,[0,1,7,8,1...|[-1198.7475978375...|[1.0,8.8067866514...|       0.0|
|  0.0|(13424,[0,1,9,14,...|[-536.20518504146...|[1.0,1.2668888864...|       0.0|
|  0.0|(13424,[0,1,11,32...|[-872.07829782643...|[1.0,4.8935582513...|       0.0|
|  0.0|(13424,[0,1,12,33...|[-442.13040868639...|[1.0,1.0208100343...|       0.0|
|  0.0|(13424,[0,1,14,18...|[-1367.3645276940...|[1.0,7.0356700723...|       0.0|
|  0.0|(13424,[0,1,14,78...|[-691.35439886578...|[1.0,8.7610945843...|       0.0|
|  0.0|(13424,[0,1,17,19...|[-810.58496634449...|[1.0,2.2788861394...|       0.0|
|  0.0|(13424,[0,1,23,63...|[-1294.9562410253...|[1.0,1.3341330902...|       0.0|
|  0.0|(13424,[0,1,27,88...|[-1528.2126194034...|[1.39959851934675...|       1.0|
|  0.0|(13424,[0,1,30,12...|[-596.35309740728...|[1.0,3.9890012014...|       0.0|
|  0.0|(13424,[0,1,31,43...|[-343.53143181163...|[1.0,8.7578342651...|       0.0|
|  0.0|(13424,[0,1,72,10...|[-704.48938694031...|[1.0,1.4538384442...|       0.0|
|  0.0|(13424,[0,1,416,6...|[-302.80782783555...|[0.99999999999999...|       0.0|
|  0.0|(13424,[0,2,3,6,9...|[-3287.0993750479...|[1.0,3.9197988476...|       0.0|
|  0.0|(13424,[0,2,3,8,2...|[-1608.7268749846...|[1.0,6.1898694199...|       0.0|
|  0.0|(13424,[0,2,4,5,1...|[-2736.6882595391...|[1.0,3.0731457387...|       0.0|
+-----+--------------------+--------------------+--------------------+----------+
only showing top 20 rows
```

Predictions based on naïve Bayes classifier.



```
In [84]: acceval=MulticlassClassificationEvaluator()

In [85]: acc=acceval.evaluate(testresults)

22/04/04 09:35:22 WARN DAGScheduler: Broadcasting large task binary with size 1365.2 KiB

In [86]: acc

Out[86]: 0.9291387026502185
```

Accuracy of the model

REFERENCES

[1].From 0 to 1:Spark for data science with python : Udemy

[2]. Machine Learning with PySpark – Review : Indonesian journal of Electrical Engineering and computer science

[3].A Project-driven Approach to Learning PySpark :Towards data science

[4].Processing Large Raster and Vector Data in Apache Spark : T. Grust et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2019), Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 551.

[5].Introduction to PySpark | Distributed Computing with Apache Spark:Geeks for Geeks

[6].SQL for beginners : Learn SQL using MY SQL and data base design : Udemy

[7].Data Warehouse Fundamentals for beginners : Udemy

[8].Relational Data base design : Udemy