Predicting customer churn at a bank

1 INTRODUCTION

1.1 Overview

Using a source of 1000 bank records, we created predictive model to demonstrate the ability to apply machine learning through Logistic Regression model to predict the likelihood of customer churn at a bank.

1.2 Purpose

Churn is a scourge on subscription businesses. When your revenue is based on recurring monthly or annual contracts, every customer who leaves puts a dent in your cash flow. Building a predictive churn model helps you make proactive changes to your retention efforts that drive down churn rates. Understanding how churn impacts your current revenue goals and making predictions about how to manage those issues in the future also helps you stem the flow of churned customers.

2 LITERATURE SURVEY

2.1 Existing problem

Existing approaches or method to solve this problem

There are seven different machine learning models to predict customer churn, including Logistic Regression, Decision Tree, Random Forest, Deep Learning (TensorFlow), K-Nearest Neighbor, Support Vector Machine and XGBoost.

2.2 Proposed solution

What is the method or solution suggested by you?

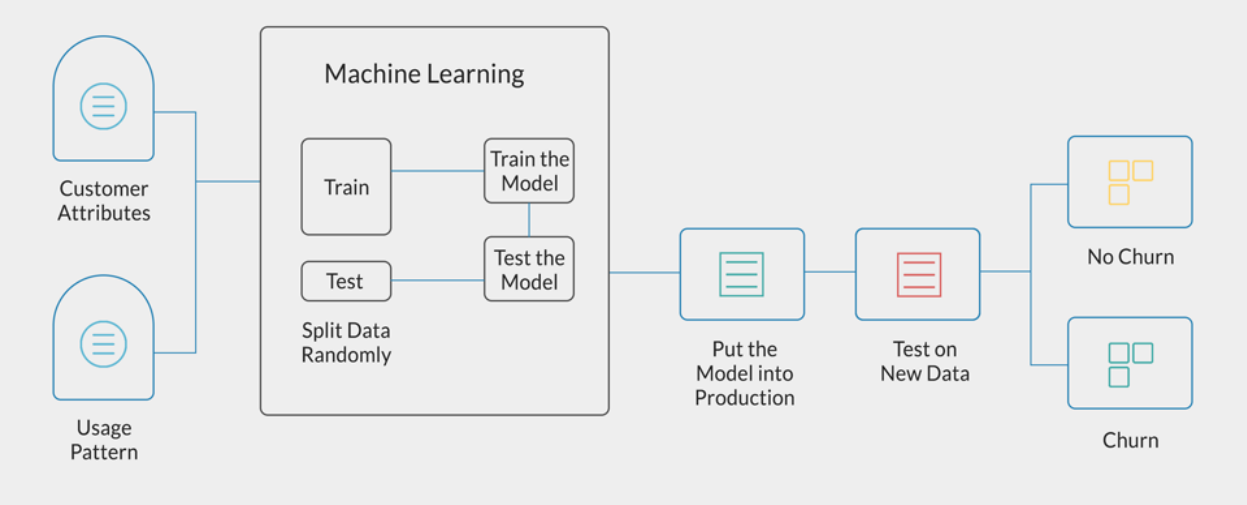
Mathematical modeling for churn is built on a statistical process called logistic regression. This process determines the relationships between points in your data set based on a formula and limits the outcome to between 0 and 1. You'll take all the customer information, purchase history, SaaS metrics, and prior churn data and turn it

into a statistical prediction of when certain types of customers might churn in the future.

3 THEORITICAL ANALYSIS

3.1 Block diagram

Diagrammatic overview of the project.



3.2 Hardware / Software designing

Hardware and software requirements of the project

1. Eclipse IDE for Enterprise Java and Web Developers for building and testing the logistic regression model coded in java.
2. Weka GUI for data processing and visualization.
3. Jupyter notebook for data processing and visualization.
4. kaggel for dataset.

4 EXPERIMENTAL INVESTIGATIONS

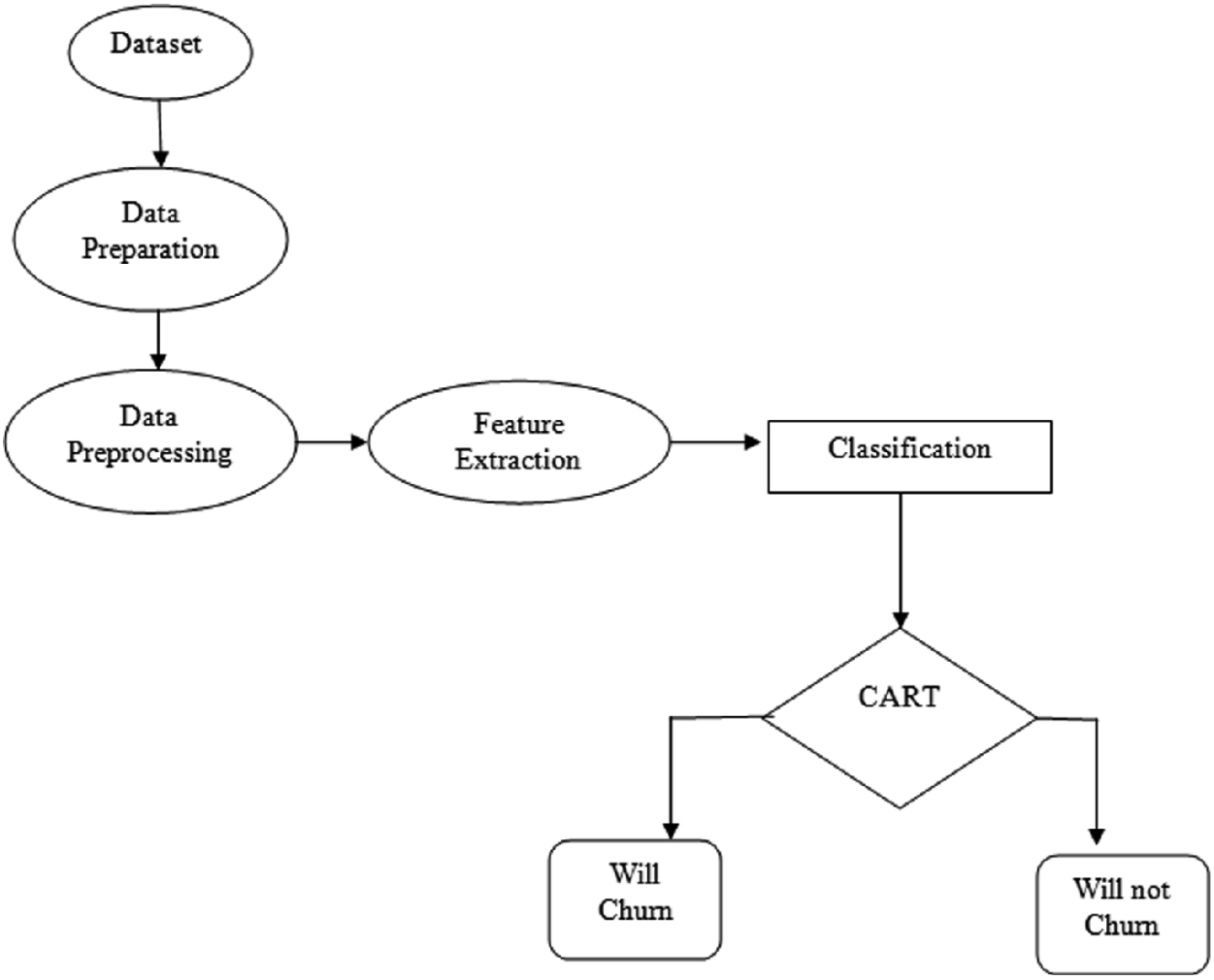
Analysis or the investigation made while working on the solution.

The accuracy and realiability of the logistic regression is compared with the other machine learning models which gives the better accuracy.They were many data

processing techniques to make the data more useful and also many libraries to visualize the data to make the modeling more accurate.

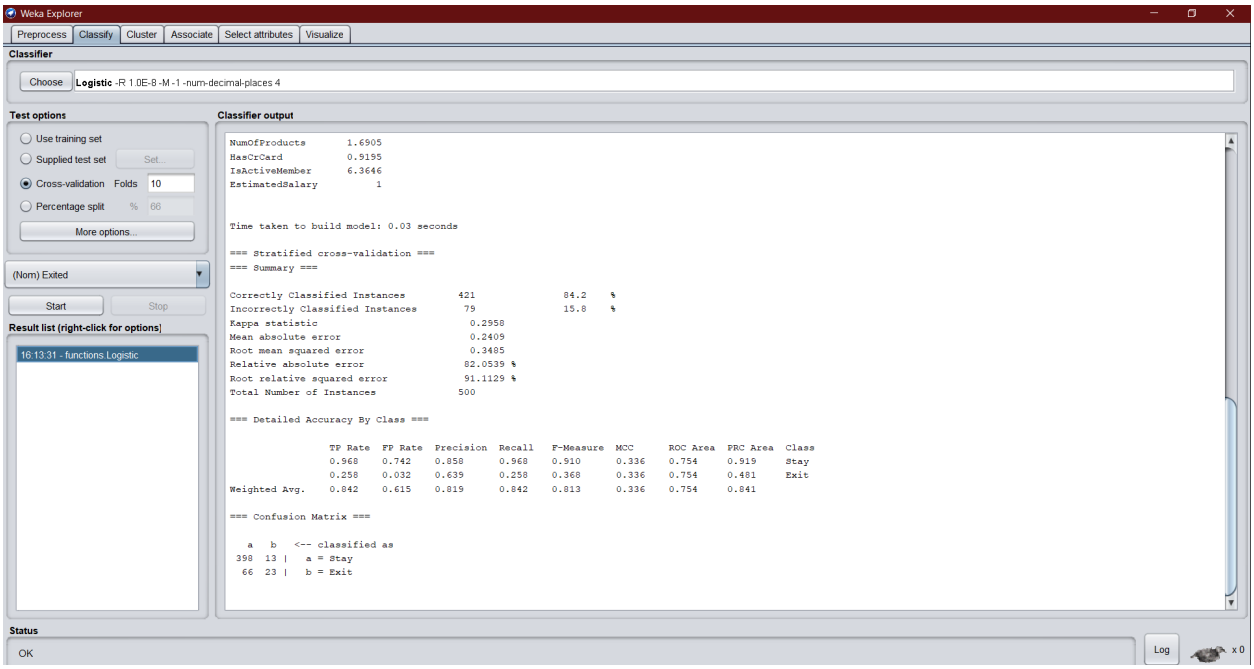
5 FLOWCHART

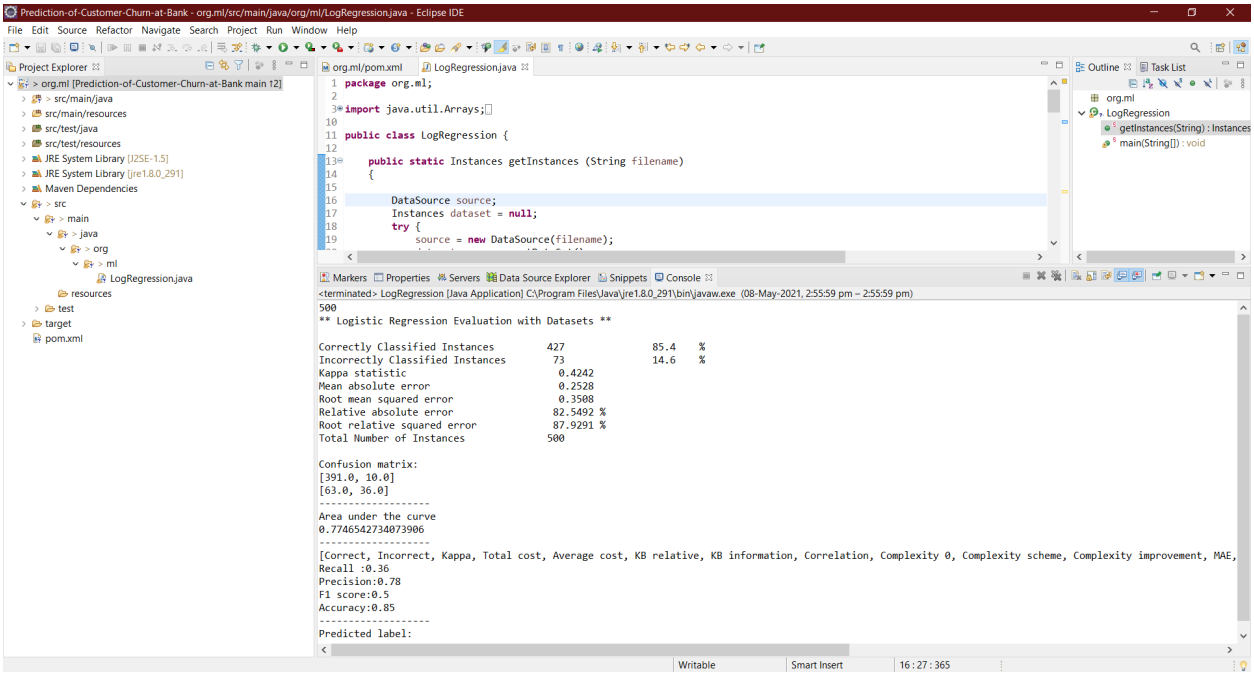
Diagram showing the control flow of the solution



6 RESULT

Final findings (Output) of the project along with screenshots.





Output:

Correctly Classified Instances	427	85.4	%
Incorrectly Classified Instances	73	14.6	%
Kappa statistic	0.4242		

Mean absolute error 0.2528
Root mean squared error 0.3508
Relative absolute error 82.5492 %
Root relative squared error 87.9291 %
Total Number of Instances 500
Recall :0.36
Precision:0.78
F1 score:0.5
Accuracy:0.85

Area under the curve
0.7746542734073906

Confusion matrix:
[391.0, 10.0]
[63.0, 36.0]

7 ADVANTAGES & DISADVANTAGES

List of advantages and disadvantages of the proposed solution

Advantage:

Churn prediction allows companies to develop loyalty programs and retention campaigns to keep as many customers as possible. Building a predictive churn model helps you make proactive changes to your retention efforts that drive down churn rates. Understanding how churn impacts your current revenue goals and making predictions about how to manage those issues in the future also helps you stem the flow of churned customers.

Disadvantage:

Traditional customer churn prediction models is that they do not fully align with their business objective, as they only predict the gross outcome, i.e., whether a customer will churn. Models estimating the net effect, however, focus on whether a customer is intent on churning AND will be retained when targeted with the campaign.Traditional customer churn prediction models are subject to feedback loops . When an organization operates a customer churn prediction model to select customers for retention campaigns, it

factually alters customer behavior. The data collected *during operation* of a customer churn prediction model is therefore *biased*.

8 APPLICATIONS

The areas where this solution can be applied

Its applicable in all Churn or Subscription based services in financial sector ,entertainment sector or in any other industry.

9 CONCLUSION

Conclusion summarizing the entire work and findings.

1. Downloaded the data set from Kaggle and loaded it into my jupyter notebook. Additionally, I also imported the requisite libraries which were essential for completing this project. The only task required now was to analyze the data, clean it, and train an ML model using the cleaned data set.
2. Missing values have the capability to hinder the training process of an ML algorithm as well as affect the accuracy of the trained model. Filled the missing values with the means of the data.
3. Data Exploration is a crucial step since it allows me to familiarise myself with the different features present in a data-set as well as the type of values that each feature column holds.
4. Data visualization is a key aspect of any Machine Learning or Data Science project. Visualizations often provide a birds-eye view of the data that allows a Data Scientist or an ML Engineer to discern trends and patterns from the data on hand. I used seaborn library's countplot function to plot categorical features and then, tried to discover trends, prevalent amongst customers that churn.
5. Lastly, the data-set was split into training and testing data to facilitate the training of an ML model.
6. Customer churn prediction is a classification problem therefore, I have used Logistic Regression algorithm for training my Machine Learning model. In my opinion, Logistic Regression is a fairly easy algorithm to implement, interpret, and very efficient to train.
7. This is the final step of my Machine Learning project, which is to test the performance of my ML model. This step is crucial since I can gauge the accuracy

of my ML model on unseen customer data. To determine the performance of my ML model, I used the test data and calculated the accuracy score as well as the confusion matrix for the predicted labels.

10 FUTURE SCOPE

Enhancements that can be made in the future.

As mentioned in the above there are many other algorithms other than the logistic regression model for machine learning problems.The machine learning is key aspect and evolving faster these days there may be more precise models which gives accurate results in the future.

11 BIBILOGRAPHY

References of previous works or websites visited/books referred for analysis about the project, solution previous findings etc.

https://www.retentionengine.com/?campaignid=12868874054&adgroupid=121638534397&adid=518249716925&utm_term=churn%20model&utm_campaign=RE+2.0&utm_source=adwords&utm_medium=ppc&hsa_acc=4282566529&hsa_cam=12868874054&hsa_grp=121638534397&hsa_ad=518249716925&hsa_src=g&hsa_tgt=kwd-502212063512&hsa_kw=churn%20model&hsa_mt=p&hsa_net=adwords&hsa_ver=3

APPENDIX

A. Source Code

```
package org.ml;

import java.util.Arrays;

import weka.classifiers.Classifier;
import weka.classifiers.evaluation.Evaluation;
import weka.core.Instance;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;
```

```
public class LogRegression {

	public static Instances getInstances (String filename)
	{

		DataSource source;
		Instances dataset = null;
		try {
			source = new DataSource(filename);
			dataset = source.getDataSet();
			dataset.setClassIndex(dataset.numAttributes()-1);



		} catch (Exception e) {
			// TODO Auto-generated catch block
			e.printStackTrace();

		}

		return dataset;
	}

	public static void main(String[] args) throws Exception{

		Instances train_data =
getInstances("C:\\Users\\Desktop\\Prediction-of-Customer-Churn-at-Bank\\org.ml\\src
\\main\\java\\org\\ml\\thop.arff");
		Instances test_data =
getInstances("C:\\Users\\Desktop\\Prediction-of-Customer-Churn-at-Bank\\org.ml\\src
\\main\\java\\org\\ml\\thop2.arff");
		System.out.println(train_data.size());

		/** Classifier here is Linear Regression */
		Classifier classifier = new weka.classifiers.functions.Logistic();
		/** */
		classifier.buildClassifier(train_data);
```

```
		/**
		 * train the alogorithm with the training data and evaluate the
		 * algorithm with testing data
		 */
		Evaluation eval = new Evaluation(train_data);
		eval.evaluateModel(classifier, test_data);
		/** Print the algorithm summary */
		System.out.println("** Logistic Regression Evaluation with Datasets **");
		System.out.println(eval.toSummaryString());
//		System.out.print(" the expression for the input data as per alogorithm is ");
//		System.out.println(classifier);

		double confusion[][] = eval.confusionMatrix();
		System.out.println("Confusion matrix:");
		for (double[] row : confusion)
			System.out.println(   Arrays.toString(row));
		System.out.println("-------------------");

		System.out.println("Area under the curve");
		System.out.println( eval.areaUnderROC(0));
		System.out.println("-------------------");

		System.out.println(eval.getAllEvaluationMetricNames());

		System.out.print("Recall :");
		System.out.println(Math.round(eval.recall(1)*100.0)/100.0);

		System.out.print("Precision:");
		System.out.println(Math.round(eval.precision(1)*100.0)/100.0);
		System.out.print("F1 score:");
		System.out.println(Math.round(eval.fMeasure(1)*100.0)/100.0);

		System.out.print("Accuracy:");
		double acc = eval.correct()/(eval.correct()+ eval.incorrect());
		System.out.println(Math.round(acc*100.0)/100.0);
```

```
                System.out.println("------------------");
                Instance predicationDataSet = test_data.get(73);
                double value = classifier.classifyInstance(predicationDataSet);
                /** Prediction Output */
                System.out.println("Predicted label:");
                System.out.print(value);


        }

}
```

Attach the code for the solution built.