MANSI KAMBLE

Contact

@ mansikamble1110@gmail.com

☎ 9579068536

📍 14/4, Laxmi Nagar, Opp Prince Park, Virar Road, Nalasopara (East) 401209

in www.linkedin.com/in/mansi-kamble-31442a244

Personal Details

Date of Birth : 11/10/2001

Technical Skills

HTML	100%
CSS	100%
JS	100%
Python	80%
UI/UX Design	80%
SQL	60%
C	60%
C++	40%

Achievements

GoogleCloudReady Facilitator Program (Jun 2022 - Jul 2022)

OBJECTIVE

To enhance my professional skills, capabilities and knowledge in an organization which recognizes the value of hard work and trusts me with responsibilities and challenges. I also seek challenging opportunities where I can fully use my skills for the success of the organization.

EXPERIENCE

Fresher -

EDUCATION

Viva Institute of Technology 2019-Present
Bachelor of Engineering
8.87

Bhavan's College 2017-2019
H.S.C
54.00%

Infant Jesus High School 2017
S.S.C
77.60%

SOFT SKILLS

Focused

Punctual

Patient

Hard working

Problem solving

Quick learner

Leadership

MANAGEMENT SKILLS

Team management

Project management

PROJECTS

Face Mask and Social Distancing Detector (Feb 2022 - May 2022)

Recently created an application for detecting the face mask and social distancing using ML.

Travel Advisor (Jul 2021 - Dec 2021)

Created a website for recommending different tourist places, hotels with restricted and COVID infected areas using HTML, CSS, JS and jQuery .

Interests

- Technical paper writing
- Reading books
- Exploring new things
- Researching
- Painting

Random Password Generator (Feb 2021 - Jun 2021)

Created a desktop application which generates random passwords according to length and strength using Python's tkinter framework.

File Converter (Jul 2020 - Dec 2020)

Created a desktop application which converts any word, pdf, excel file to any desired file format using HTML, CSS, JS and APIs.

CERTIFICATES

Participated in Imperia Intercollegiate Project Competition (03-2022)

Participated in TechSpark 2.0 Technical Paper Presentation (04-2021)

Machine Learning for Prediction of Imbalanced Data: Credit Fraud Detection

Thanh Cong Tran
Hong Bang International University (HIU)
215 Dien Bien Phu, Ward 15, Binh Thanh District,
Ho Chi Minh City, Vietnam
congtt@hiu.vn

Tran Khanh Dang (✉)
Ho Chi Minh City University of Technology (HCMUT)
268 Ly Thuong Kiet street, District 10, Ho Chi Minh City, Vietnam
Vietnam National University Ho Chi Minh City (VNU-HCM)
Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam
khanh@hcmut.edu.vn

2021 15th International Conference on Ubiquitous Information Management and Communication (IMCOM) | 978-1-6654-2318-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/IMCOM51814.2021.9377352

Abstract—**Online transactions have increased drastically over the past decades. Credit card transactions account for a large percentage of these transactions. This leads to rise activities of credit card fraud transactions, causing losses in the finance industry. Therefore, it is vital to create reliable fraud detection systems, including two labels of fraud and no-fraud. However, there are highly unbalanced data between these two labels. In this paper, we use two resampling approaches of synthetic minority oversampling technique (SMOTE) and adaptive synthetic (ADASYN) to handle an imbalanced dataset to obtain the balanced dataset. The machine learning (ML) algorithms, named random forest, k nearest neighbors, decision tree, and logistic regression are applied to this balanced dataset. The comprehensive classification measurements, including fundamental, combined, and graphical measurements are used to evaluate the performances of these models. We observe that after resampling the dataset, the ML algorithms mentioned show the positive results of classification for fraudulent activities.**

Keywords— Machine learning, Classification, Imbalanced Data, SMOTE, ADASYN, Fraud Detection, classification measurements.

I. Introduction

Fraudulent activities have been increasing significantly in various industries worldwide, particularly in the financial industry. In financial companies, credit card fraud is considered the most problematic and is needed to prevent it as soon as possible. In order to reduce dramatically consequences of credit card fraud, fraud detection approaches need to investigate to strictly handle. Systems of fraud detection are trained through previous transactions so as to determine upcoming ones [1].

In fraud identification, the number of fraudulent cases is significantly less than normal circumstances. This leads to the status of imbalanced data. In the skewed dataset, one class of dataset has an extremely high great deal of instances while the other class accounts for a very small number of ones. However, machine learning algorithms work effectively on the balanced distribution of classes. In order to tackle the issue of the skewed datasets, various remedies have been researched in the past decades. In these researches, three groups, known as data-level, algorithm-level, and ensemble solutions, are commonly proposed [2].

In this paper, two techniques of resampling approach, named synthetic minority oversampling technique (SMOTE) and adaptive synthetic (ADASYN) are used to obtain a balanced dataset. The machine learning (ML) algorithms, named random forest (RF), k nearest neighbors (KNN), decision tree (DT), and logistic regression (LR), then, are utilized to train the balanced dataset obtained. Comparisons of performances of machine learning algorithms based on two resampling techniques are pointed out to indicate the best case to detect credit fraud.

The remainder of this study is described as follows: Section 2 depicts related work. Section 3 is about the description of the dataset. Section 4 presents sampling techniques. Section 5 describes machine learning algorithms and classification measurements. Section 6 illustrates the classification outcomes. Lastly, section 7 concludes this paper.

II. Relataed Works

Credit card fraud has resulted in a huge loss in both customers and financial companies worldwide. Therefore, researchers have an effort to search for optimized methods to detect and prevent this fraud. Recently, machine learning approaches have applied to detect fraudulent activities. In [3], the approach named extreme outlier elimination and hybrid sampling technique is proposed to generate reliable anticipations to not only fraud but also non-fraud circumstances. In [4], several ensemble classifiers are analyzed comprehensively through regression, voting. These classifiers, then, are compared with effective single classifiers such as KNN, naïve bayes, support vector machine (SVM), DT, radial basis function (RBF), and multilayer perceptron (MLP). These algorithms are evaluated based on three different datasets that are treated using SMOTE. In [1], the SMOTE-edited nearest neighbor (ENN) method is reviewed as the best one to detect the fraud compared with other different classifiers among a set of oversampling approaches, and the SMOTE-Tomek's Links (TL) showed good outcomes according to the set of under-sampling techniques. In [5], the combined probabilistic and neuro-adaptive method is proposed to an identified database of credit card transactions. This combination illustrated a high classification of fraud. In [6], a hidden Markov model (HMM) is employed to create a model of the operations sequences in credit card transaction processing and showed how it is used for the identification of frauds.

In summary, there are a variety of ways to create models for identifying fraudulent activities in order to reduce risks in the financial sector. In [7], several modern techniques based on machine learning, data mining, sequence alignment, fuzzy logic, genetic programming are analyzed to detect credit card

978-0-7381-0508-6/21/$31.00 ©2021 IEEE

fraud. However, depending on each specific case, these techniques mentioned in this paper have different strengths and weaknesses. In [8], a study of imbalanced classification methods is reviewed for the credit fraudulent activities in the experiment. This study indicated that using unbalanced classification approaches were not effective and sufficient to cope with the skewed data problem. Moreover, the ML algorithms are applied to the skewness dataset, but the classifiers did not detect true positive and true negative values in several circumstances [9]. Additionally, the classification accuracy criterion does not effectively and comprehensively evaluate the ML models, resulting in an accuracy paradox in several circumstances [10], [11].

In this paper, the comprehensive performance measurements for classification tasks [12], [13] which are vital prerequire to establish the fraud detection systems are utilized to model evaluation of RF, KNN, DT, and LR algorithms. The classification measurements include three major groups. The first group is fundamental measurements, including accuracy, precision, specificity, and sensitivity. The second group is the combined measurement, such as F1 score, geometric mean (G-mean), likelihood ratios, balanced accuracy, Mathews correlation coefficient (MCC). The last one is the graphical plot, named receiver operating characteristic curve (ROC), and area under the ROC curve (AUC). Moreover, in order to increase the performance of classification, processing data is implemented first. Then, we employ resampling techniques i.e. SMOTE and ADASYN to cope with imbalanced dataset first.

III. Dataset

In this study, we select the dataset of credit card fraud prediction as a case study of the imbalanced dataset obtained in [14]. In September 2013, this dataset included transactions that are collected within 2 days and created by credit cards through European cardholders. The dataset has 284807 transactions and 31 columns. These columns consist of 28 numerical features, named V1 to V28, a feature of “Time”, a feature of “Amount” and a label of “Class”. The label of “Class” is divided into two classes, such as positive class (frauds) denoted 1, and negative class (no-frauds) denoted 0. This dataset is highly skewed so we can define the positive and negative classes as the minority and majority classes. The imbalanced ratio (IR) is considered as a ratio of the sample size of the majority class over the sample size of the minority one, as shown in (1) [15].

$$IR = \frac{N_{maj}}{N_{\min}} \qquad (1)$$

In the dataset of this paper, the minority class (frauds) occupied only 0.172% of all transactions revealed in Fig. 1.

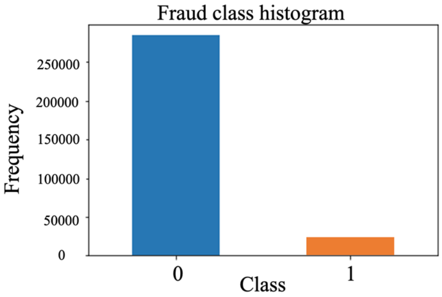


Fig. 1. Fraud class histogam

After analyzing the dataset, we recognized that the feature “Amount” has values from 0 to 25691.16. However, if features are of relatively the same scale and/or close to the normal distribution, the ML algorithms will have good performances or converge faster. Therefore, standardization is used to eliminate the mean and scale to unit variance to alleviate the wide range of the feature “Amount”, so approximately 68% of the values lie in between (-1, 1) [16]. A new feature, named “normAmount” is added to the dataset. Additionally, the “Time” and “Amount” features do not need to build models so we will drop these features. We obtained the dataset of 30 features and the “Class” label.

Next, so as to cope with the issue of imbalanced data, two resampling techniques: SMOTE and ADASYN are used in this study. After implementing these techniques for credit fraud detection dataset, we accomplished the balanced dataset.

IV. Sampling Techniques

There are various techniques to deal with the imbalanced dataset [17]. These techniques are summarized and categorized as four major groups, known as algorithm level, data level, cost-sensitive learning, and ensemble-based. First, the algorithm level technique aims at making an attempt to adapt current classifier learning algorithms to bias the learning for the minority class. Second, the purpose of the data level technique is to rebalance the classification distribution through resampling data space. Next, the cost-sensitive learning technique combines algorithm and data level techniques to optimize the total cost errors to both classes. Finally, the ensemble-based approaches include a combination between one of the aforementioned techniques and an ensemble learning algorithm, such as data level and cost-sensitive techniques.

Moreover, the data level methods are separated into oversampling methods, under-sampling methods, and hybrids methods. The under-sampling approaches drop instances of the majority class to make a subset of the primitive dataset. This leads to the loss of much valuable information on the original dataset. Conversely, the oversampling methods randomly duplicate instances in the minority class, so the dataset size may increase, cause rising training time of ML models. The hybrids approaches mix both sampling methods, but these methods are quite complex.

In order to address these disadvantages, we use the SMOTE and ADASYN techniques to cope with the imbalanced dataset. As both techniques are popular and have demonstrated their benefits when solving skewed datasets in many different applications [18], [19], [20].

The SMOTE is one of the popular and effective oversampling methods. The SMOTE is used to duplicate existing instances of minority class and permitted to make new artificial instances class based on knowledge about neighbors that surrounds each sample in the minority class. The SMOTE is separated into three parts, known as randomizing the minority class instances, calculating k nearest neighbors for each minority class instance, and producing synthetic instances [21].

The ADASYN is a technique of improvement from the SMOTE. This technique is relied on the idea of utilizing the weighted distribution to adaptively generate minority class samples: more synthetic data is produced for the minority class samples that are tougher to learn compared to those minority class samples that are more straightforward to learn. The ADASYN technique proved its effectiveness through improving learning relevant to the distributions of data in two ways. This technique minimizes bias caused by a classification imbalance and adapts to the classification decision boundary for difficult samples. [22].

V. Machine Learning Algorithms

A. *Random forest*

The RF is one of the supervised learning algorithms for both classification and regression. The RF is an ensemble approach that includes many decision trees. This algorithm illustrates better outcomes once the number of trees in the forest is increasing and also prevents the model from overfitting. Each decision tree in the forest gives several outcomes. These outcomes are combined together so as to achieve more accurate and stable predictions [23], [24].

B. *K nearest neighbors*

The KNN algorithm, which is a non-parametric method is utilized to deal with both classification and regression problems. Although this algorithm is simple, it can outperform dramatically more complex approaches. The KNN requests only k selected, a number of neighbors to be taken into account once conducting the classification. If k value is small, the estimate of classification is susceptible to large statistical error. On the other hand, if k value is huge, this allows the distant points to contribute to the classification, causing removing several details of the class distributions. As a result, the value of k is considered to select in order to mitigate the classification error on several independent validation data or by cross-validation procedures [25], [26].

C. *Decision tree*

The DT algorithm is one of the most commonly used classifiers for classifying problems. This is mainly due to the fact that this algorithm is capable of handling complicated problems by giving a more understandable, easier to explain representation and also their adaptability to the inference task by generating out logic classification rules. The DT encompasses nodes for checking properties, edges for branching according to values of the chosen attribute and leaves labeling classes in which for each leaf a unique class is attached. There are two major procedures in the DT. The first procedure is to build the tree, while the second one is used for the knowledge inference, i.e. classification [27], [28].

D. *Logistic regression*

The LR algorithm is one of the most prevailing approaches for binary classification. The relationship between predictors that might be continuous, binary, and categorical is indicated in the LR model. The dependent variable can be binary. According to several predictors, we anticipate whether something will occur or not. We identify the probability of belonging to each category for a given set of predictors [29].

E. *Classification measurement performance*

In order to evaluate the performance of the aforementioned ML algorithms on the resampled dataset, the comprehensive evaluation measures, including fundamental, combined, and graphical evaluation performance are used in this paper.

a) Fundamental evaluation measures: In binary classification, a confusion matrix shown in Table I is one of the best ways to assess the ML models. The confusion matrix shows the relations between the classifier outputs and the true ones. Table I indicates four different combinations of predicted and actual values explained as true positive (TP), false positive (FP), true negative (TN), false negative (FN). In this study, samples with the presence of fraud are defined as the positive class, and samples with the absence of no-fraud are considered as the negative class. Based on the confusion matrix, accuracy, precision, sensitivity, and specificity are calculated.

TABLE I. Confusion matrix

		Predicted Values	
Actual Values	Class	Negative (0)	Positive (1)
	Negative (0)	True Negative (TN)	False Positive (FP)
	Positive (1)	False Negative (FN)	True Positive (TP)

Classification accuracy is considered as one of the most popularly employed measures to the performance of classification. It is a proportion of the accurately classified samples over the total number of samples as shown in (2).

$$Accuracy = \left(TP + TN\right) / \left(TP + TN + FP + FN\right) \tag{2}$$

Positive predictive value or precision illustrates the ratio of positive samples accurately classified over the total number of positive predicted samples as shown in (3).

$$Precision = TP / \left(TP + FP\right) \tag{3}$$

Sensitivity, true positive rate, or recall illustrates the rate of positive samples accurately classified over the total of positive predicted samples and negative predicted samples as indicated in (4).

$$Sensitivity = TP / \left(TP + FN\right) \tag{4}$$

Specificity is expressed as the ratio between negative samples correctly classified and the total of positive predicted samples and negative predicted samples as indicated in (5).

$$Specificity = TN / (TN + FP) \tag{5}$$

b) Combined evaluation measures: Apart from basic measurements, combined assessments for binary classification are also considered as the alternative approaches to assess the ML models. These assessments are straightforward but effective. This study considers several measures such as F1 measure, G-mean, likelihood ratios, balanced accuracy, and MCC.

F1 measure is expressed as the harmonic mean among recall and precision as illustrated in (6). F1 measure has a range from 0 to 1, which means its high values reveal high classification performance.

$$F1 = \frac{2*(Precision*Recall)}{(Precision+Recall)} \qquad (6)$$

The G-mean is considered as the prediction accuracy outcome for both sensitivity and specificity as shown in (7).

$$G = \sqrt{Sensitivity*Specificity} \qquad (7)$$

The likelihood ratios are separated as a positive likelihood ratio and a negative likelihood ratio. As shown in (8), the positive likelihood ratio is defined as the ratio sensitivity over one minus specificity.

$$L = Sensitivity \;/\; \left(1 - Specificity\right) \qquad (8)$$

Conversely, the negative likelihood ratio is expressed in (9) as the ratio between one minus sensitivity and specificity.

$$\lambda = \left(1 - Sensitivity\right) \;/\; Specificity \qquad (9)$$

Based on the definition of the negative likelihood ratio and the positive likelihood ratio, that means that a lower negative likelihood and a higher positive likelihood ratio will have correspondingly better performance on negative and positive classes. The threshold of positive likelihood is shown in Table II.

TABLE II. INTERPRETATION OF THRESHOLDS OF POSITIVE LIKELIHOOD

L value	Contribution of Model
>10	Good
5 -10	Fair
1-5	Poor
1	Negligible

The balanced accuracy represents the average of sensitivity and specificity as shown in (9).

$$Balanced\ accuracy = \left(Sensitivity + Specificity\right) \;/\; 2 \qquad (9)$$

MCC is not affected by the imbalanced dataset problem. MCC illustrated in (10) is defined as a contingency matrix approach of computing the Pearson product-moment correlation coefficient between predicted and actual values.

$$MCC = \frac{TP*TN - FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (10)$$

c) Graphical evaluation performance: ROC curve which is a graphical evaluation describes the diagnostic capability of a binary classifier system since its taxonomy threshold is different. The ROC curve is established by plotting the ratio of true positive versus false positive rates. The AUC measures the whole two-dimensional area under the whole ROC curve. The AUC can assess models by overall performance, so it is more considered in model evaluation. The suggestion of the following scale for AUC value interpretation is shown in Table III.

TABLE III. AUC PERFORMANCE

AUC value	Performance of Model
0.9 – 1.0	Excellent
0.8 – 0.9	Very good
0.7 – 0.8	Good
0.6 – 0.7	Fair
0.5 – 0.6	Poor

VI. RESULTS

A. Results based on skewed dataset

The performance of four ML algorithms on the skewed dataset is represented in this section. The confusion matrices of RF, KNN, DT, and LR obtained on the skewed dataset are shown in Table IV, V, VI, VII, respectively.

TABLE IV. CONFUSION MATRIX OF RF ON SKEWED DATASET

	Predicted Values		
True Values	Class	0	1
	0	85290	6
	1	34	113

TABLE V. CONFUSION MATRIX OF KNN ON SKEWED DATASET

	Predicted Values		
True Values	Class	0	1
	0	85289	7
	1	41	106

TABLE VI. CONFUSION MATRIX OF DT ON SKEWED DATASET

	Predicted Values		
True Values	Class	0	1
	0	85271	25
	1	36	111

TABLE VII. CONFUSION MATRIX OF LR ON SKEWED DATASET

	Predicted Values		
True Values	Class	0	1
	0	85284	12
	1	56	91

The accuracies of the mentioned ML models on the skewed dataset are indicated in Table VIII. Based on Table VIII, the accuracies values of the mentioned ML models are extremely high. The aforementioned confusion matrices show that TP samples are much less than TN samples. However, in order to establish a reliable fraud detection system, detecting positive samples should more than negative ones. We, therefore, need to improve these models despite their high accuracies.

TABLE VIII. ACCURACIES OF ML MODELS ON SKEWED SATASET

Measurements	RF	KNN	DT	LR
Accuracy	99.95%	99.94%	99.91%	99.92**%**

B. Results based on resampling dataset

The ML models on the skewed dataset do not show good performances to detect the fraudulent activities, so we need to resample this skewed dataset to the balanced dataset, then applying the ML methods to accomplish better models. In this section, the performances of four ML approaches based on SMOTE and ADASYN techniques are evaluated for the prediction of fraud using 30 features discussed in the dataset section. The confusion matrices of RF, KNN, DT, and LR algorithms based on the resampled dataset are accomplished to identify the TN, FN, FP, and TP. The confusion matrices of RF, KNN, DT, and LR algorithms with SMOTE are shown in Table IX, XI, XIII, and XV, respectively. The confusion matrices of RF, KNN, DT, and LR algorithms with ADASYN are shown in Table X, XII, XIV, and XVI, respectively. It is clearly shown that the TP samples are greater than the TN sample, except the case of the LR algorithm with SMOTE and ADASYN. That is the first step to demonstrate the effectiveness once applying two resampling techniques to the fraudulent dataset.

TABLE IX. CONFUSION MATRIX OF RF WITH SMOTE

	Predicted Values		
True Values	Class	0	1
	0	85143	19
	1	0	85427

TABLE X. CONFUSION MATRIX OF RF WITH ADASYN

	Predicted Values		
True Values	Class	0	1
	0	85128	30
	1	1	85428

TABLE XI. CONFUSION MATRIX OF KNN WITH SMOTE

	Predicted Values		
True Values	Class	0	1
	0	84953	209
	1	0	85427

TABLE XII. CONFUSION MATRIX OF KNN WITH ADASYN

	Predicted Values		
True Values	Class	0	1
	0	84979	179
	1	0	85429

TABLE XIII. CONFUSION MATRIX OF DT WITH SMOTE

	Predicted Values		
True Values	Class	0	1
	0	84892	270
	1	113	85314

TABLE XIV. CONFUSION MATRIX OF DT WITH ADASYN

	Predicted Values		
True Values	Class	0	1
	0	84940	218
	1	63	85366

TABLE XV. CONFUSION MATRIX OF LR WITH SMOTE

	Predicted Values		
True Values	Class	0	1
	0	83045	2117
	1	7256	78171

TABLE XVI. CONFUSION MATRIX OF LR WITH ADASYN

	Predicted Values		
True Values	Class	0	1
	0	77244	79194
	1	11565	73864

The fundamental classification performance measurements of four ML algorithms with SMOTE and ADASYN are revealed in Fig. 2. The accuracies of the aforementioned algorithms with resampled datasets are almost similar (above 99%), except the LR algorithm, approximately 94.51% and 88.96% based on SMOTE and ADASYN, respectively. The four ML algorithms precision also indicates positive outcomes with the highest value of 99.98% for RF with SMOTE. Similarly, the specificity of all the mentioned algorithms shows very high results (above 99%), except the LR algorithms, approximately 97.51% and 91.48% based on SMOTE and ADASYN respectively. Furthermore, the sensitivity reveals extremely good outcomes, approximately 100% in terms of RF and KNN with both SMOTE and ADASYN techniques. In contrast, the sensitivity result of LR based on ADASYN is only about 86.46%. We also can observe that LR with ADASYN has the lowest results regarding all fundamental indexes among cases. The results of four ML methods based on SMOTE and ADASYN are quite similar in each ML approach. Generally, the fundamental measurements illustrate excellent outcomes of the four ML approaches with the resampling data. The fundamental classification measurements show the results of not only accuracy value, but also other values as mentioned to evaluate ML models more comprehensively.

Fundamental Measurements	RF		KNN		DT		LR	
	SMOTE	ADASYN	SMOTE	ADASYN	SMOTE	ADASYN	SMOTE	ADASYN
Accuracy	99.99%	99.98%	99.88%	99.90%	99.78%	99.84%	94.51%	88.96%
Precision	99.98%	99.96%	99.76%	99.79%	99.68%	99.75%	97.36%	91.12%
Sensitivity	100.00%	100.00%	100.00%	100.00%	99.87%	99.93%	91.51%	86.46%
Specificity	99.98%	99.96%	99.75%	99.79%	99.68%	99.74%	97.51%	91.48%

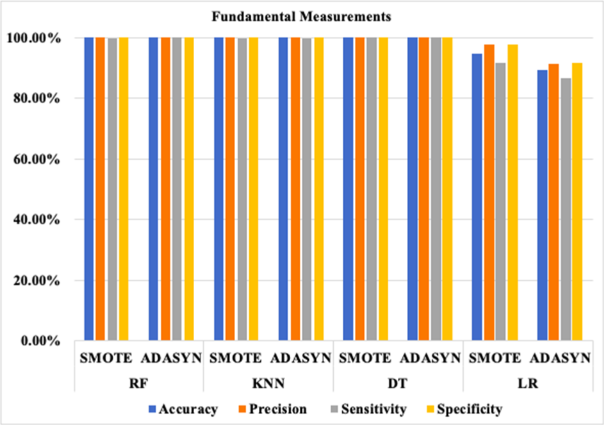


Fig. 2. Fundamental classification performance measurements

Next, in terms of combined measurements, the indexes such as F1 score, G-mean, balanced accuracy, and MCC are performed in Fig. 3. Overall, the results of the combined measures of ML models based on resampling data are also very high. The results of RF, KNN, and DT with SMOTE and ADASYN are also similar, above 99%. However, LR with SMOTE and ADASYN shows relatively low, particularly only 77.23% of the MCC index. We see that LR with ADASYN has the lowest results of all indexes the combined measurements. However, RF with SMOTE and ADASYN results are almost similar and show the highest values for all these indexes.

Combined Measurements	RF		KNN		DT		LR	
	SMOTE	ADASYN	SMOTE	ADASYN	SMOTE	ADASYN	SMOTE	ADASYN
F1 score	99.99%	99.98%	99.88%	99.90%	99.78%	99.84%	94.34%	88.73%
G-mean	99.99%	99.98%	99.88%	99.89%	99.78%	99.84%	94.46%	88.94%
Balanced accuracy	99.99%	99.98%	99.88%	99.89%	99.78%	99.84%	94.51%	88.97%
MCC	99.98%	99.96%	99.76%	99.79%	99.55%	99.67%	89.17%	77.23%

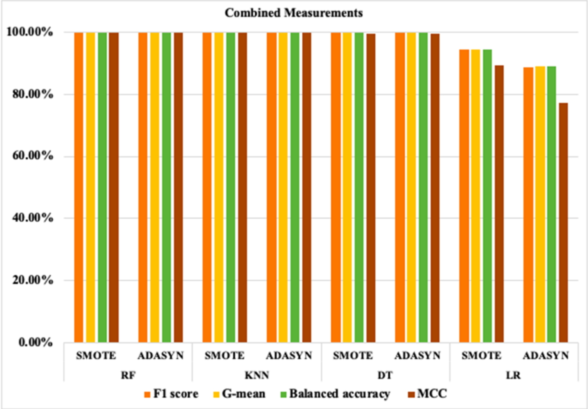


Fig. 3. Combined classification performance measurements

The positive likelihood ratio computed is shown in Table XVII. Based on Table II, it is clearly shown that all model contribution is good. But, there is a dramatic difference between these outcomes of models. The results of RF is very high, about 4482.21, while LR shows very low results, only 10.15.

TABLE XVII. POSITIVE LIKELIHOOD RATIO

ML algorithms		Positive likelihood ratio	Model contribution
RF	SMOTE	4482.21	Good
	ADASYN	2838.57	Good
KNN	SMOTE	407.47	Good
	ADASYN	475.74	Good
DT	SMOTE	315.00	Good
	ADASYN	390.34	Good
LR	SMOTE	36.81	Good
	ADASYN	10.15	Good

Last but not least, we also debate graphical assessment performance in order to make a sufficient evaluation of the fraudulent activities in this study. The ROC curve analysis with SMOTE and ADASYN of all the mentioned ML algorithms are shown in Fig. 4 and Fig. 5, respectively. These ROC curves obtained indicate high results. A comparison of AUC is also described in Fig. 6. The AUC of RF and KNN based on the SMOTE technique shows the highest results, about 100%, whereas these metrics of DT and LR have results of 99.8% and 98.9%, respectively. Additionally, the AUC results of four ML algorithms with ADASYN also show the same trend to SMOTE, with approximately 100% of RF and KNN. It is followed by DT and LR, about 99.8% and 95.9%, respectively.

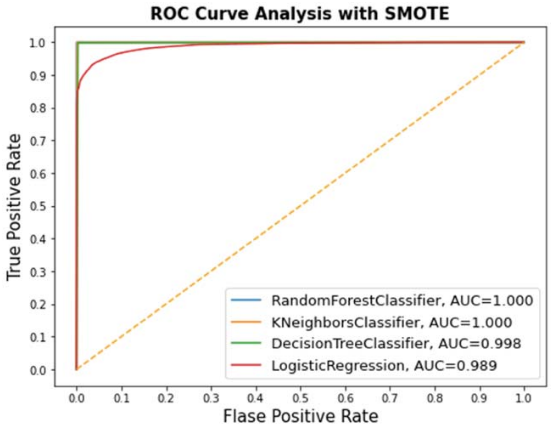


Fig. 4. ROC Curve Analysis with SMOTE

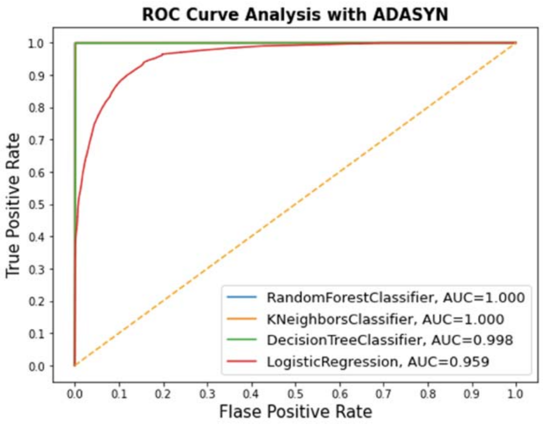


Fig. 5. ROC Curve Analysis with ADASYN

Measurement	RF		KNN		DT		LR	
	SMOTE	ADASYN	SMOTE	ADASYN	SMOTE	ADASYN	SMOTE	ADASYN
AUC	100.00%	100.00%	100.00%	100.00%	99.80%	99.80%	98.90%	95.90%

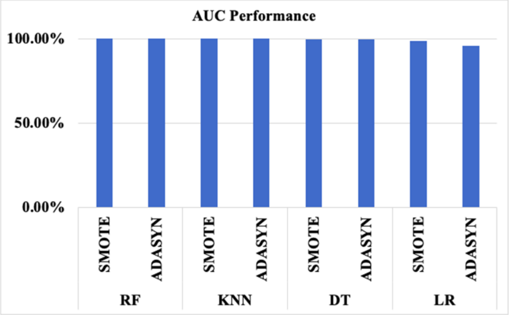


Fig. 6. AUC measurement performance

In general, the measurements show positive outcomes of all mentioned approaches. The results of the fraud classification of each ML algorithm based on SMOTE and ADASYN are almost similar. Additionally, this study also shows different evaluation outcomes of these ML algorithms which may important factors to consider for creating the fraud detection system. Based on the results of three-main group classification measurements, RF and KNN algorithms based on SMOTE and ADASYN have better results than DT and LR. Moreover, LR with ADASYN shows quite low results regarding all indexes compared with other algorithms.

VII. CONCLUSION AND FUTUREWORK

In this study, we comprehensively analyze the fraud dataset through ML algorithms. The processing data stage is applied to this dataset so as to increase the effectiveness of ML models. Since this dataset is highly skewed, we employ two simple, but effective resampling techniques such as SMOTE and ADASYN to achieve balanced data. The distinguishing classification evaluation indexes are utilized in the balanced dataset to prove the effectiveness of ML models of fraud detection. The classification measurements consist of three primary types, known as fundamental, combined, and graphical assessment. The comparisons between each algorithm based on SMOTE and ADASYN, and between classification measures-based ML algorithms are indicated in this study. These comparisons are considered as one of the crucial factors to build the speed and effective system detection of fraud.

In the future, we want to collect precisely and extended the dataset through financial companies so as to accomplish the exhaustive model. We continue to improve the fraud detection system through other different ML approaches. Besides, extending and applying ML techniques over highly-skewed datasets to other application domains like big data sampling and clustering [31, 32, 33], recommendation systems [34], and security and privacy issues with deep learning [35] are also of our great interest in the future.

References

[1] D. S. Sisodia, N. K. Reddy and S. Bhandari, "Performance evaluation of class balancing techniques for credit card fraud detection," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, Chennai, 2017.

[2] B. Zhua, B. Baesens and S. K. Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction," *Information Sciences,* vol. 408, pp. 84-99 , 2017.

[3] T. M. Padmaja, N. Dhulipalla, R. S. Bapi and P. Krishna, "Unbalanced Data Classification Using extreme outlier Elimination and Sampling Techniques for Fraud Detection," *15th International Conference on Advanced Computing and Communications (ADCOM),* pp. 511-516, 2007.

[4] P. Kumari and S. P. Mishra, "Analysis of Credit Card Fraud Detection Using Fusion Classifiers," *Advances in Intelligent Systems and Computing,* vol. 711, pp. 111-122, 2018.

[5] R. Brause, T. Langsdorf and M. Hepp, "Neural data mining for credit card fraud detection," in *Proceedings 11th International Conference on Tools with Artificial Intelligence*, Chicago, IL, USA, 1999.

[6] A. Srivastava, A. Kundu, S. Sural and A. K. Majumdar, "Credit Card Fraud Detection Using Hidden Markov Model," *IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,* vol. 5, pp. 37 - 48, 2008.

[7] S. B. E. Raj and A. A. Portia, "Analysis on credit card fraud detection methods," in *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, Tamilnadu, India, 2011.

[8] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. Hacid and H. Zeineddine, "An Experimental Study With Imbalanced Classification Approaches for Credit Card Fraud Detection," *EEE Access,* vol. 7, pp. 93010-93022, 2019.

[9] S. Mittal and S. Tyagi, "Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, India, 2019.

[10] M. F. Uddin, "Addressing Accuracy Paradox Using Enhanched Weighted Performance Metric in Machine Learning," in *2019 Sixth HCT Information Technology Trends (ITT)*, United Arab Emirates, 2019.

[11] F. J. Valverde-Albacete and C. Pela´ez-Moreno, "100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox," *PLOS ONE,* vol. 9, no. 1, pp. 1-10, 2014.

[12] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing and Management,* vol. 45, no. 4, p. 427–437, 2009.

[13] M. Bekkar, H. K. Djemaa and T. A. Alitouche, "Evaluation Measures for Models Assessment over Imbalanced Data Sets," *Journal of Information Engineering and Applications,* vol. 3, no. 10, pp. 27-38, 2013 .

[14] "Kaggle," [Online]. Available: https://www.kaggle.com/mlg-ulb/creditcardfraud. [Accessed 2 9 2020].

[15] RuiZhua, YiwenGuob and Jing-HaoXuec, "Adjusting the imbalance ratio by the dimensionality of imbalanced data," *Pattern Recognition Letters,* vol. 133, pp. 217-223, 2020.

[16] "Towards data science," [Online]. Available: https://towardsdatascience.com/scale-standardize-or-normalize-with-scikit-learn-6ccc7d176a02. [Accessed 5 9 2020].

[17] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk and F. Herrera, Learning from Imbalanced Data Sets, Switzerland: Springer, 2018.

[18] K. Li, W. Zhang, Q. Lu and X. Fang, "An Improved SMOTE Imbalanced Data Classification Method Based on Support Degree," in *2014 International Conference on Identification, Information and Knowledge in the Internet of Things*, Beijing, China, 34-38.

[19] L. Demidova and I. Klyueva, "SVM classification: Optimization with the SMOTE algorithm for the class imbalance problem," in *2017 6th Mediterranean Conference on Embedded Computing (MECO)*, Bar, Montenegro, 2017.

[20] C. Lu, X. L. S. Lin and H. Shi, "Telecom Fraud Identification Based on ADASYN and Random Forest," in *020 5th International Conference on Computer and Communication Systems (ICCCS)*, Shanghai, China, 2020.

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Artificial Intelligence Research ,* vol. 16, p. 321–357, 2002.

[22] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Hong Kong, China, 1322-1328.

[23] T. K. Ho, "Random decision forests," in *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition*, IEEE Computer Society, Washington, DC, USA, 1995.

[24] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, p. 5–32, 2001.

[25] O. Beckonert, M. E. Bollard, T. M. Ebbels, H. C. Keun, H. Antti, E. Holmes, J. C. Lindon and J. K. Nicholson, "NMR-based metabonomic toxicity classification: hierarchical cluster analysis and k-nearest-neighbour approaches," *Analytica Chimica Acta,* vol. 490, no. 1-2, p. 3–15, 2003.

[26] "Classification of pyrolysis mass spectra by fuzzy multivariate rule induction-comparison with regression, K-nearest neighbour, neural and decision-tree methods," *B.K. Alsbergav; R. Goodacrea; J.J. Rowlandb; D.B. Kella ,* vol. 348, no. 1-3, pp. 389-407 , 1997.

[27] J. R. Quinlan, "Induction of decision trees," *Machine Learning ,* vol. 1, p. 81–106, 1986.

[28] J. R. Quinlan, C4.5 : programs for machine learning, San Mateo, Calif. : Morgan Kaufmann Publishers, 1993.

[29] Bishop and C. M, Pattern recognition and machine learning, New York: Springer, 2006.

[30] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal Of Artificial Intelligence Research,* vol. 16, pp. 321-357, 2002.

[31] N. L. Hoang, L. H. Trang, T. K. Dang: A Comparative Study of the Some Methods Used in Constructing Coresets for Clustering Large Datasets. *SN Comput. Sci.*, Springer Nature, 1(4): 215 (2020)

[32] L. H. Trang, N. L. Hoang, T. K. Dang*:* A Farthest First Traversal based Sampling Algorithm for k-clustering. IMCOM 2020*: 1-6*

[33] N. L. Hoang, T. K. Dang, L. H. Trang: A Comparative Study of the Use of Coresets for Clustering Large Datasets. FDSE 2019*: 45-55*

[34] T. K. Dang , Q. P. Nguyen, V. S. Nguyen: Evaluating Session-Based Recommendation Approaches on Datasets from Different Domains. FDSE 2019: 577-592

[35] T. Ha, T. K. Dang, H. Le, T. A. Truong: Security and Privacy Issues in Deep Learning: A Brief Review. SN Comput. Sci., Springer Nature, 1(5): 253 (2020)

Inverse Document Frequency-Weighted Word2Vec model to recommend apparels

Priyanka Meel
Information Technology
Delhi Technological University
New Delhi, India
priyankameel@dtu.ac.in

Agniva Goswami
Information Technology
Delhi Technological University
New Delhi, India
agniva03@gmail.com

***Abstract*— with the rapid growth of e-commerce markets the need for recommendation engine and efficient algorithms are becoming the need of the hour for business models of the companies to generate a huge amount of profit. This paper proposes a hybrid algorithm to benefit apparel retailing market which gives the benefits of both, semantics based search and frequency based search. Later this paper compares results of the proposed hybrid algorithm with the other known algorithms used to recommend products.**

Keywords- e-commerce; recommendation engine; hybrid algorithm; semantics; frequency;

I. INTRODUCTION

Nowadays, with the popularity of e-commerce websites, online shopping has become a trend among people. With billions of users accessing the shopping websites every day, there arises a necessity to develop state of the art machine learning algorithms which can make the user buying experience hassle free. Companies such as Amazon invest a lot on research and development to improve the current existing algorithms. One such field of research is product recommendation, either content based or collaborative based.

According to statistics, Product recommendation generates about 35% of revenue of Amazon.com, which ranges to approximately more than 40billion USD in 2016. Internally Amazon uses both content based and collaborative filtering based recommendation; we have used content based (text and image) recommendation here because the collaborative data is very closely guarded by Amazon.

This paper suggests recommendations methods to women apparels. Basically apparel is an item that is worn on the body. Product recommendation makes the buying easier for the customer and also benefits the company as it lures the customer in buying more products. Researchers have developed many content based recommendation algorithms such as Term frequency, Bag of Words, Inverse Document Frequency, Word2Vec which was originally developed by Google in the year 2013, it is on their shoulder that we are standing and applying these algorithms and sometimes modifying to improve our query results accordingly. Some examples where product recommendation is used are, Amazon.com in “people who bought this also bought...”, youtube.com suggestions, etc.

RELATED WORK:

Basically in BagofWords ,let’s say we have N data and thus N titles, if for each title we can represent it as a D-dimensional vector where D is the total number of words in all the titles, therefore we can apply Euclidean distance method to it to predict similar products. Where each value of the vector denotes the frequency of the word in the particular title.

In tf-idf(term frequency-inverse document frequency), we replace the values with the tf-idf value, tf (Wj,Ti) is Number of times word Wj occurred in Ti divided by Number of words in T, and idf(Wj,D) is equal to log(number of titles in Corpus D divided by number of titles in D that contains word Wj),and finally we multiply the tf value with the idf value.

A recommendation system or engine is a kind of information system that suggests only important results to the user by eliminating redundant information by using different algorithms which are automated or computerized. Recommender systems are used in search engines, social tags, research article recommendation, movies etc.

A recommender system suggests the recommendations by one the two ways:

1. Collaborative filtering
2. Content based filtering.

Content-based Filtering: This approach is common while a recommender system is being designed. Content-based filtering methods are based on a description of the item and a profile of the user’s preferences [10]. In content-based recommender system, items and user’s preferences are used to recommend items. The algorithm recommends items based on the items the user has seen in the past. The user provides the data to the system either implicitly or explicitly, implicitly for example by clicking a link and explicitly by rating an item, the content based recommender system uses these data to further recommend products. In particular, various possible items are compared with the query item or item that was rated by the user. The recommendation suggestion is updated with every activity of the user, for example when the user gives good rating to a particular product, preference of that product on future recommendations increases.

978-1-7281-1380-7/19/$31.00 ©2019 IEEE

Collaborative Filtering: in this type of filtering the algorithm that is used, takes into consideration the behaviour of other user and items, in terms of what rating the other user has given to the same item, what closest items other user has preferred to the current item etc. For example if item A is liked by three users and item B is liked by two users, then the recommender system will predict that item A will also be liked by the current user and will be shown in the result

II. METHODOLOGY

A. DATA ACQUISITION, PRE-PROCESSING & CLEANING

1. Data Acquisition: The data has been acquired from Amazon product Advertisement API available at:

https://docs.aws.amazon.com/AWSECommerceService/latest/DG/Welcome.html [14]. I have acquired over 183000 data about women's tops in apparel section.

2. Data Cleaning: The initial number of data points after acquisitions are 183138, and the total number of features are 19. The 19 features contains features such as 'asin', 'author', 'titile', etc. But among these only a few are useful in uniquely identification of items, so we excluded all others and kept features such as 'asin', 'brand', 'color', 'product_type_name', 'medium_image_url', 'title', 'formatted_price'.

- **Handling missing data of various features:** The data was then checked to see if any null values are present or not. There were null values in features 'formatted_price' and 'color'.

After removing the data rows which has null 'formatted_price', the number of data points were reduced from 183138 to 28395.

Then the null values of 'color' features were handled, after the removal of rows which had null 'color', the data points reduced from 28395 to 28385.

- **Removing Near Duplicate Items:** Some titles had all duplicate words except the last few words.
  The algorithm:
  1. Split every title into words and for each title, check the number of similar words with the previous title and count it.
  2. Subtract the counter from the total number of words.
  3. If it is greater than 2, or simply the number of dissimilar words in the two title is great than 2, then consider it as a new title, otherwise ignore the title.

After de-duping, the number of data points reduced to around 16 thousand.

3. Data Pre-Processing:

Initially stop words have been downloaded from NLTK, such as an', 'down', 'during', 'about', 'was', 'couldn', 'of', 'few', 'my', 'weren', 'off', 'all', 'because', 'had', 'both', 'after', 'having', 'again', 'y' etc have been removed by simply iterating of the words present in the whole document corpus because words like these are not at all useful in recommendation

B. THE PROPOSED ALGORITHM:

The Flowchart:

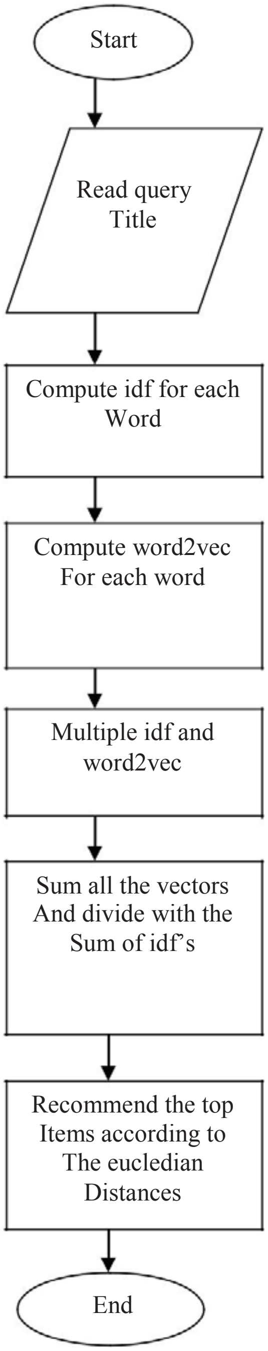

Start
Read query Title
Compute idf for each Word
Compute word2vec For each word
Multiple idf and word2vec
Sum all the vectors And divide with the Sum of idf's
Recommend the top Items according to The eucledian Distances
End

Fig. 1 flow chart of idf-weighted word2vec algorithm

IDF-WEIGHTED WORD2VEC MODEL

The similarity of average Word2Vec with IDF weighted word2vec is that, in average word2vec we take IDF supposedly to be 1, but here for each word W(j) in a title T(i), we run Word2Vec and get a 300 dimensional vector and then multiply that vector with that word W(j)'s IDF(inverse document frequency) and then sum up all the corresponding cell of the each vector to get a new vector just like average Word2Vec, and then her we divide each cell of the resultant vector by the total sum of IDF 's of each word W(j) of the title, to get our final vector of each title T(i). The Algorithm also shown in flow chart (Fig 1)**:**

1. **Computing IDF:**
   - IDF is always measured for a given word Wj and a given document corpus D, IDF(Wj,D) is equal to log(number of titles in Corpus D divided by number of titles in D that contains word Wj), thus IDF measure how less frequent a word is**.**
2. **Computing Word2Vec:**
   - Let T(i) be a title ,which consists of 'k' words, w1,w2,…..,wk, for each word W(j), we get a 300 dimensional vector by simply running Word2Vec model on word W(j).
   - For every title T(i), take up every word W(j)'s 300 dimensional vector which we got from previous step, and multiply it with the IDF of that particular word W(j) which we got from step 1.
   - Now create vector of size 300, where the value of each cell is equal to the sum of the values of all the corresponding cell of all the vectors of each word W(j). Finally we divide each cell of the resultant vector by the total sum of IDF 's of each word W(j) of the title(calculated from step 1), to get our final Vector of each title T(i).

We can apply similar approach to make TF-IDF weighted Word2Vec, where we just multiply each word W(j)'s vector with its TF-IDF and the rest of the procedure is same.

III.ANALYSIS AND RESULTS

The results of Bag of Words and TF-IDF show that these algorithms are purely based on the frequency of words in the title and in the whole data corpus, whereas algorithms like average word2vec lays stress on semantic based recommendations. Thus when these two types of algorithms are merged, we get results which show products based on semantics as well as based on the texts present in the title.

In the results of IDF-Weighted Word2Vec, the two unique results we got are :

1. Anna kaci woman's asymmetrical sheer brown leopard cheetah print long sleeve top multicoloured medium.
2. Merona woman's printed blouse brown leopard print xxl

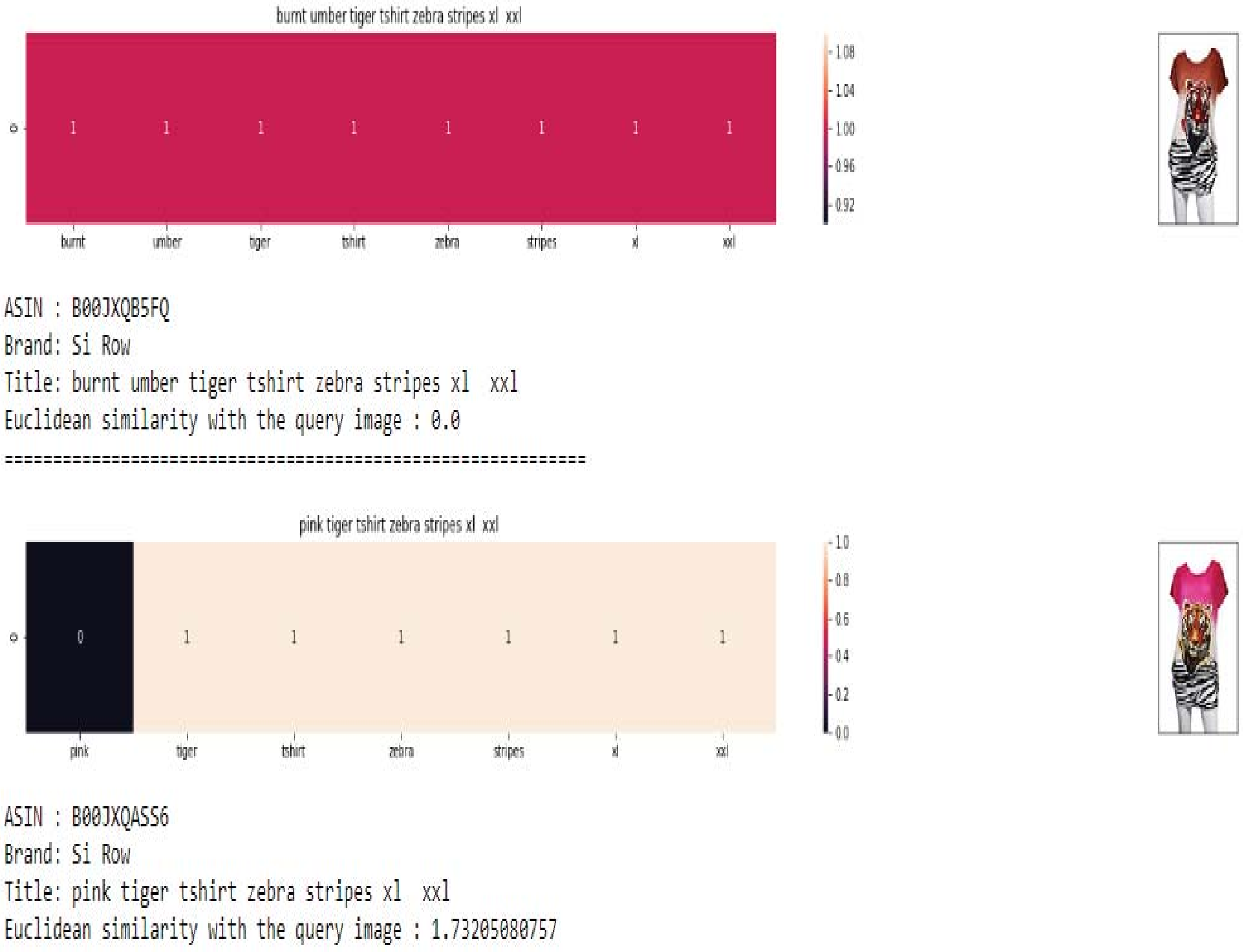
In both these results we see the word 'leopard', we get such type of recommendation because of Word2Vec being semantic based; leopard is semantically similar to tiger, which is contained in our query image. Also here, we get results based on the similarity of texts of our query results like 'women's' etc.

Thus this algorithm has computational complexity same as Average Word2Vec but has better results than Average Word2Vec because of the inclusion of both semantics and frequency of words.

Bag of Words results: As this is a frequency based technique so the results shown below (Fig. 2) shows that images which has more common words to that of the query title has lesser eucledian distance, hence given more preference in search results, for example in Fig.2.1 "pink tiger shirt zebra stripes xl xxl" has most number of common words hence lesser eucledian distance.

Tf-Idf results: It is also a frequency based algorithm, so the results are similar to BagofWords algorithm, but gives better results than BagofWords, because it also takes into account the presence of the word in the whole data corpus, thus it gives results shown in (Fig. 3)

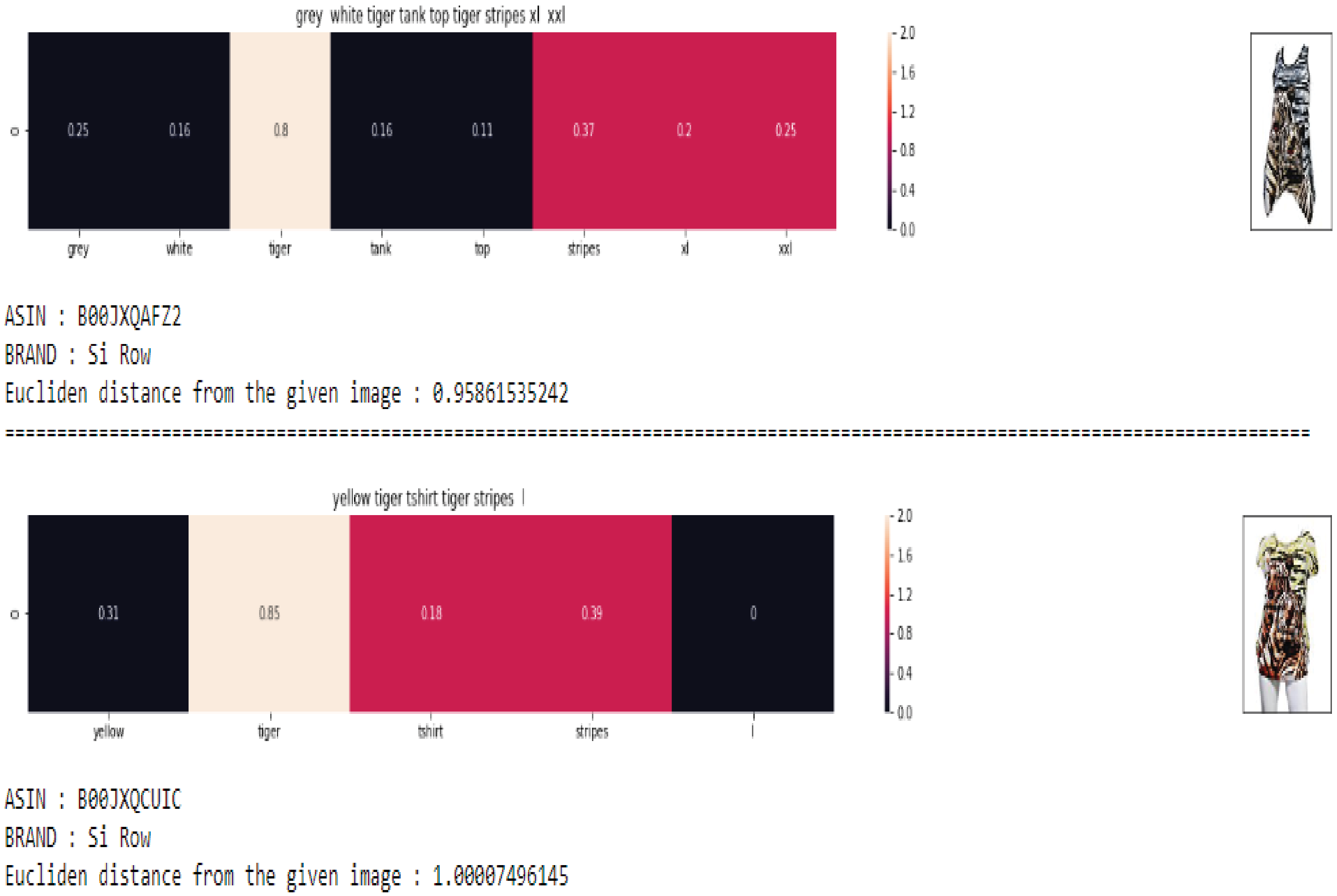
Idf Weighted Word2Vec Results: In addition to the results of BagofWords and tf-idf, in idf weighted Word2Vec we also got the results shown below in (Fig. 4),which were not given by the above two algorithms at all because it combines both semantic based results as well as frequency based results. It gives results based on the fact that a word occurs many times in a title and very fewer titles in the data corpus should contain that particular word.



```
ASIN : B00JXQB5FQ
Brand: Si Row
Title: burnt umber tiger tshirt zebra stripes xl  xxl
Euclidean similarity with the query image : 0.0
============================================================

ASIN : B00JXQASS6
Brand: Si Row
Title: pink tiger tshirt zebra stripes xl  xxl
Euclidean similarity with the query image : 1.73205080757
```

Fig.2 BagofWords results



```
ASIN : B00JXQAFZ2
BRAND : Si Row
Eucliden distance from the given image : 0.95861535242
=============================================================================================================================
```

```
ASIN : B00JXQCUIC
BRAND : Si Row
Eucliden distance from the given image : 1.00007496145
```

Fig.3 tf-idf results

IDF-Weighted Word2Vec Results:

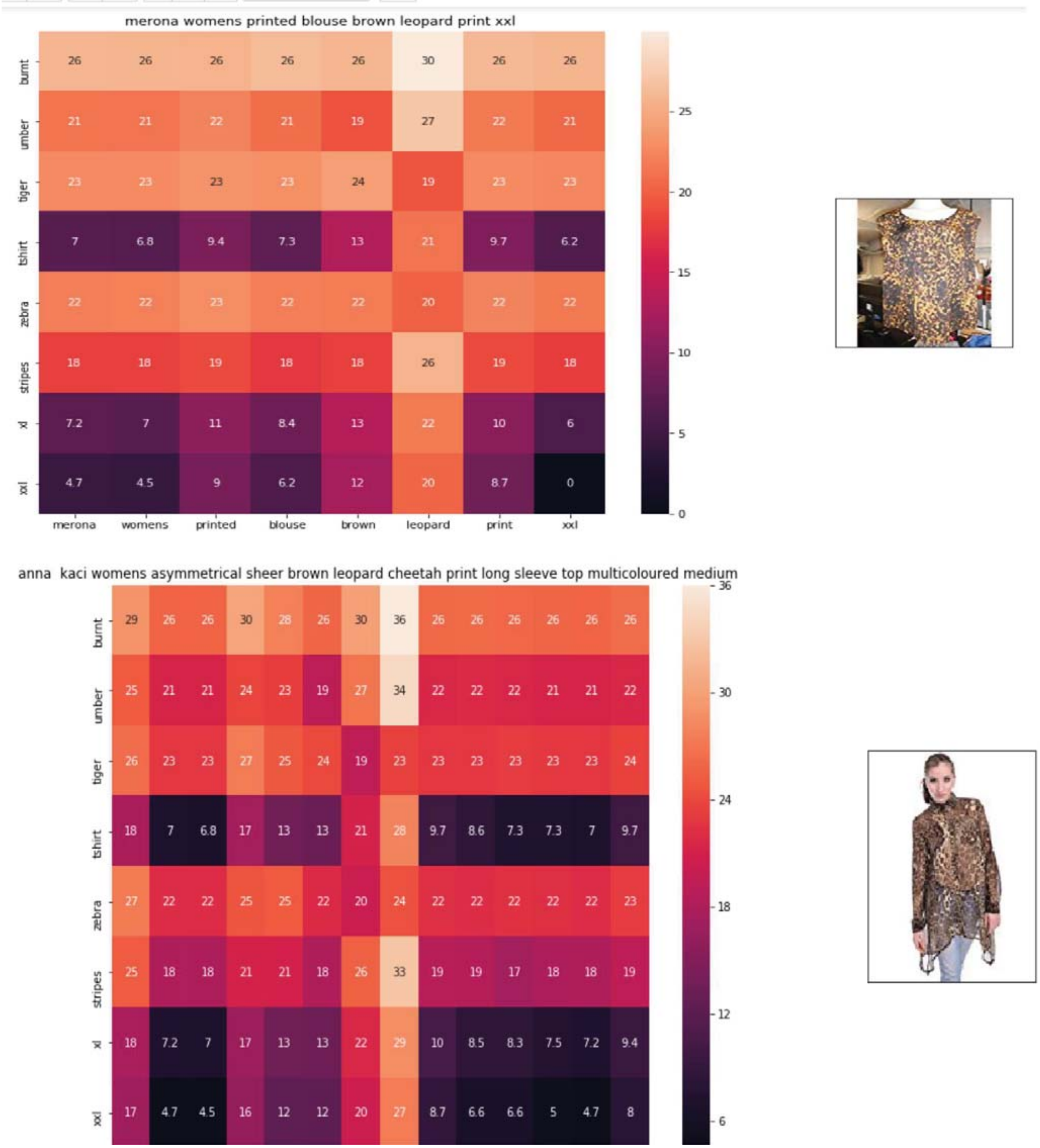


Fig.4 idf-weighted word2vec results

IV. CONCLUSION AND FUTURE WORK

The results above shows the benefits of using idf-weighted word2vec algorithm over BagofWords, tf-idf. Because of the semantic based results it gives better predictions of the user's future needs and frequency based results gives predictions based on the users query product. More work can be done in this area, on addition to idf weighted word2vec we can increase the efficiency by reading the image and classifying the image property and then give results which are similar to the query image.

REFERENCES

[1] R. C. Bagher, H. Hassanpour, and H. Mashayekhi, "User trends modeling for a content-based recommender system," *Expert Syst. Appl.*, vol. 87, pp. 209–219, 2017.

[2] M. S. Tajbakhsh and J. Bagherzadeh, "Microblogging hash tag recommendation system based on semantic TF-IDF: Twitter use case," *Proc. - 2016 4th Int. Conf. Futur. Internet Things Cloud Work. W-FiCloud 2016*, pp. 252–257, 2016.

[3] G. Carullo, A. Castiglione, and A. De Santis, "Friendship recommendations in online social networks," *Proc. - 2014 Int. Conf. Intell. Netw. Collab. Syst. IEEE INCoS 2014*, pp. 42–48, 2014.

[4] J. Hannon, M. Bennett, and B. Smyth, "Recommending twitter users to follow using content and collaborative filtering approaches," *Proc. fourth ACM Conf. Recomm. Syst. - RecSys '10*, p. 199, 2010.

[5] M. Razghandi and S. A. H. Golpaygani, "A Context-Aware and User Behavior-Based Recommender System with Regarding Social Network Analysis," *2017 IEEE 14th Int. Conf. E-bus. Eng.*, pp. 208–213, 2017

[6] S. Aral and D. Walker, "Identifying Influential and Susceptible Members of Social Networks," *Science,* vol. 337, no. 6092, pp. 337-341, June 2012.

[7] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[8] P. Alencar and D. Cowan, "The use of machine learning algorithms in recommender systems : A systematic review," *Expert Syst. Appl.*, vol. 97, pp. 205–227, 2018.

[9] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a Social Network or a News Media?", *Int. World Wide Web Conf. Comm.*, pp. 1–10, 2010.

[10] Aggarwal, Charu C. (2016). *Recommender Systems: The Textbook*. Springer-ISBN *9783319296579.*

[11] Yoav Goldberg and Omer Levy, "word2vec Explained: Deriving Mikolov et al.'s Negative-Sampling Word-Embedding Method", arXiv:1402.3722v1, Feb 2014.

[12] Appliedaicourse.com

[13] https://deeplearning4j.org

[14] https://docs.aws.amazon.com/AWSECommerceService/latest/DG/Welcome.html