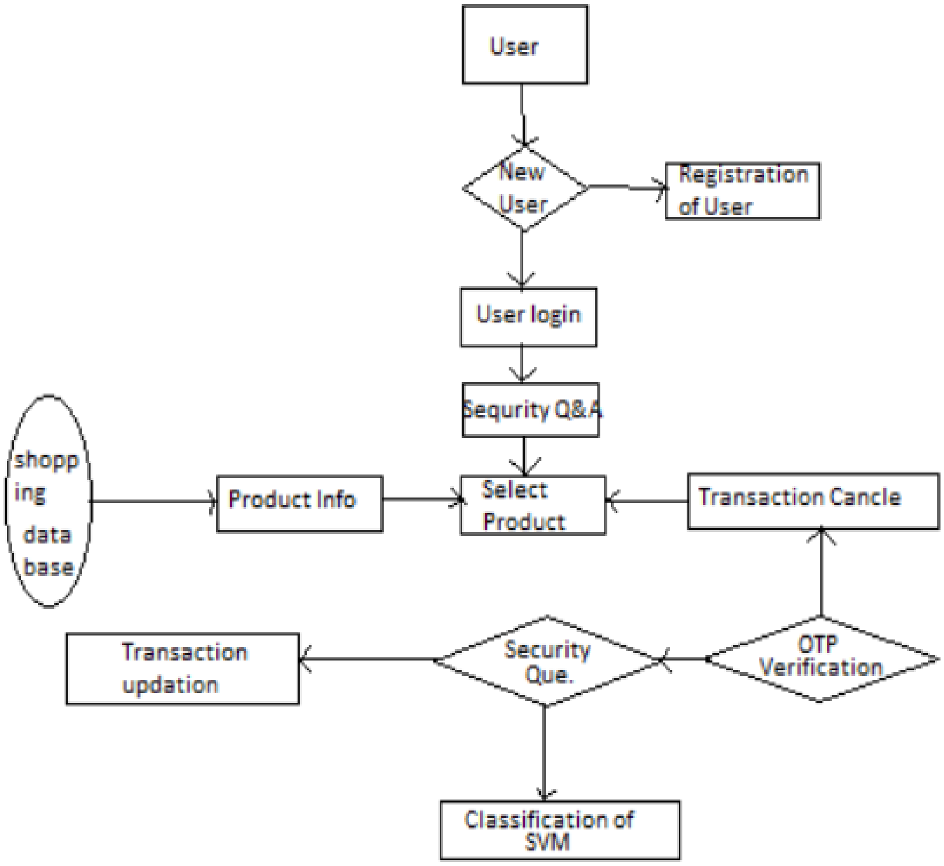
Experiment No-13

Title: Design and analysis of Fraud detection using SVM algorithm

Aim: Design and analysis of Fraud detection using SVM algorithm

SVM is a Supervised machine learning algorithm that can be used for both classifications and regression problems. It's often used in classification issues, however. In this algorithm, we plot each data item in an "N" dimensional space to some degree, with the price of each function being the price of a chosen coordinate showing the hyper-plane differentiating the 2 groups from the hyper-plane. During this algorithm, we take the shopping data item of each consumer to some degree in an 'n-dimensional space (where n is the number of features you have) with the price of each function being the price of a chosen coordinate.

In methodology, This contains total of five steps, firstly, If the user is a new user then the system will ask for registration. The user will register herself or himself with the help of a name. city, mobile number, and email Then the system will ask for login details such as username and password. To enroll for the security questions verification, the login user selects several questions and supplies confidential answers that only the user knows. One Time Passwords (OTP) also play an important role in this whole system. OTP can be used once and then expires. The user needs to prove his or her identity with the help of OTP. To Predict or classify patterns SVM Classifies into fraudulent or non-fraudulent.



Performance Evaluation It Concludes that each system faces its own problems while dealing with dataset description. The proposed system come with the SVM model of real databases system come with the SVM model of real database which helps in acquiring a maximum of 99.9% accuracy. The Artificial Neural Network (ANN) comes with 97.32% accuracy while Hidden Markov Model(HMM) has 94.7% accuracy. ANN has a high processing time and excessive training for large neural networks, difficult to set up and run. Also, Bayesian Networks need excessive training and have 96.52% accuracy.

Conclusion This project presents a classification of online credit / open-ended credit the challenges faced by cardholders also because of the card issuer, the variety of fraud implemented by the persons who commit the fraud. The behavior-based classification method using SVM is used in this project. Effective performance in fraud detection is often given using SVM. SVM typically provides a creative solution. Within the type of threshold for separating the data, the SVM gains versatility. These qualities make the SVM carry out the problem of classification in the complex domain and also generate an honest result. The suggested approach offers greater identification precision and is also scalable for managing larger quantities of transactions.

Experiment No-14

Title: Design and analysis of spam filtering using naïve Bayes algorithm

Aim: Design and analysis of spam filtering using naïve Bayes algorithm

Probability is one of the major branches in mathematics. It is often used to solve many real-life situations. Mathematicians spent years building practical models to fit in the growing issues. Within this area, conditional probability represents the probability of an event happening in relation to the occurrence of another event. This characteristic of the Bayes Theorem makes it the ideal choice for mathematicians and scientists to obtain the conditional probability that would otherwise remain unknown to them. Consequently, this theorem is utilized in many applications to categorize objects into different groups or to predict whether or not certain event is going to happen given what has already occurred, such as machine translation – language translation carried out by computers ,text categorization – computers assigning texts into different categories [3], or risk management – prediction of financial risk of tasks using computer.

A spam is often characterized by its advertising nature or the fraud message it contains in order to deceive the users to acquire their confidential personal information. Users often identify such emails by the frequent appearance of specific phrases such as "Please enter your bank account number here" or "Congratulations, you've won 100,000$!". These patterns are recognizable by computer algorithms and are used as decisive factors to judge the nature of an email. Through using such methodologies, even though the computers haven't developed equal intelligence as humans yet at this point, the email servers can be almost as clever as human and save email users countless amount of time from being annoyed by filtering spams ourselves. Behaving as a type of artificial intelligence, the spam filters are often computer programs designed by computer scientists, using a Bayesian analytical algorithm to generalize patterns in spams, and compute the probability that an email is actually a spam given these patterns.

This algorithm performs Bayesian analysis to help email users to maintain a "healthy" inbox without being constantly annoyed by the spams or wasting their time on identifying which emails are spam and which are not. Although no prediction can be made on whether an email is spam or no under the assumptions of classical conditional probability, the application of Naive Bayes Classifier (NBC) here is capable of accomplishing this task. The final decision to classify this email into the "Spam" section or the "Inbox" section is made by comparing the computed probability with the threshold value. A value between 0 and 1 reflects how serious the email server believes the consequence of mistakenly classifying a non-spam email as spam is.

NAIVE BAYES CLASSIFIER IN SPAM FILTERING

Naive Bayes Mode is one of the two most-used classification models. The basic concept of Naive Bayes Classifier (NBC) is applying Bayes Theorem where the objects or attributes have independence. The detailed process is show below. a) There are two possible classes or categories, denoted by symbol A and A′, to classify each email into in this application: spam and non-spam; b) Vector denoted by $\vec{}$ = < , , , , … > is used to represent a series of common attributes of spam emails. In this application, the attributes of emails are simply individual words or phrases; c) Every email is represented by a vector denoted by $\vec{}$ = < , , , , … >, where each represents the value of the attribute .

The email spam filter will first scan through each email searching for those specific words or phrases, and form the vector of each email using the presence or absence of the attributes it detected; d) All attributes take the binary form in this application, meaning that each is "1" if this word or phrase is present in an email, and "0" if this word or phrase is absent in an email; e) Therefore, the vector representation of each email will look like an “ID number” consisting of a string of “1”s and “0”s, indicating which attributes are present in this particular email.

CONCLUSION

To conclude, this paper has investigated the mathematical process of Naive Bayes Classifier in email servers, and how this algorithm applies the extended version of Bayes Theorem to compute the probability of each email being spam and further classify the emails into “Spam” section and “Inbox” section. Furthermore, this paper has explored the critical role threshold value play in this algorithm, and how does the variance of this value affect the final classification results. Even though this algorithm is more sophisticated and effective in real life, it still has certain limitations, which accounts for the mistakes made by these anti-spam filters, as users can find a spam email in their “Inbox” sometimes while finding certain legitimate emails in the “Spam” section once in a while. One significant weakness of this algorithm is the conditional independence assumption, which is the premise to use Bayes Theorem. Often in reality, events, or attributes, are interrelated to and dependent on each other. Thus, the presence or absence of each word or phrase have an impact on the presence or absence of other words or phrases as words of relevant topics appear together, vice versa.

Experiment No-15

Title: Design and analysis of customer segmentation using Decision tree classifier

Aim: Design and analysis of customer segmentation using Decision tree classifier

Customer segmentation is the process of dividing a customer base into groups of individuals that are similar in certain ways relevant to marketing, such as age, gender, interests, and spending habits. It enables companies to target specific groups with tailored promotions, products, or services that are most likely to resonate with them. Machine learning has become a popular tool for automating the process of customer segmentation, providing a more efficient and effective way to identify patterns and relationships within customer data.

There are several different methods for using machine learning to perform customer segmentation, including:-

- **Clustering algorithms:** These algorithms divide customers into groups based on their characteristics and behaviour. For example, **k-means Clustering** can be used to find the k number of clusters in a dataset.
- **Decision trees:** These algorithms use a tree-like model to identify the most important variables that influence customer behaviour. By using decision trees, companies can determine which customers are most likely to respond to certain marketing campaigns or products.
- **Neural networks:** These algorithms can be used to model complex relationships between customers and their behaviour. Neural networks can identify patterns in customer data that are not easily recognizable through traditional methods.
- **Association rule learning:** This method finds the relationships between customer attributes and behaviours, such as buying habits and product preferences. Association rule learning can help companies understand which products are frequently purchased together and target customers accordingly.

```
# Importing the Libraries
import numpy as np
import pandas as pd
import datetime
import matplotlib
import matplotlib.pyplot as plt
from matplotlib import colors
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt, numpy as np
from mpl_toolkits.mplot3d import Axes3D
from sklearn.cluster import AgglomerativeClustering
from matplotlib.colors import ListedColormap
from sklearn import metrics
import warnings
import sys
if not sys.warnoptions:
    warnings.simplefilter("ignore")
```

```
np.random.seed(42)
# Loading the dataset
dataset = pd.read_csv("marketing_campaign.csv" ,sep="\t")
print("Number of datapoints in the dataset:", len(dataset))
dataset.head()
# We need to remove the NA values from our dataset, so we will use .dropna()
datasetdataset = dataset.dropna()
no=len(dataset)
print(f" After eliminating the rows with missing values, there are ultimately {no} number of datapoints in the dataset ")
dataset["Dt_Customer"] = pd.to_datetime(dataset["Dt_Customer"])
dates = []
for i in dataset["Dt_Customer"]:
    ii = i.date()
    dates.append(i)
# Dates of the most recent and oldest client enrollments on record
newest_date = max(dates)
print(f"Date of the most recent customer's enrollment in the records: {newest_date}")
oldest_date = min(dates)
print(f" Date of records' oldest customer's enrollment: {oldest_date}")
```

Experiment No-16

Title: Design and analysis of predict patient readmissions based on their medical history and treatment

Aim: Design and analysis of predict patient readmissions based on their medical history and treatment

Patients and providers face a great amount of uncertainty before, during, and after hospital encounters. Predictive modeling holds promise for identifying patients at the highest risk for adverse events, such as extended length of stay (LOS), 30-day readmission, and death within the hospital encounter. Despite the success of predictive models in achieving discriminatory power in these and other areas, simplistic models cannot account for complicated intersections of medical, institutional, and demographic factors. Conversely, complex models that account for these interactions are difficult or impossible to interpret or audit, and therefore may be inactionable or harmful if put into use, and can also be difficult for healthcare providers to understand or accept[1–3]. Recent studies suggest that a focus on metrics such as 30-day readmission without addressing underlying causes may lead to increased patient mortality and increased cost without improving patient outcomes[4].

Thirty-day readmissions were predicted with an area under the receiver operator characteristic curve (ROC AUC, here abbreviated as simply “AUC”) of 0.76. The Brier score loss (BSL) was 0.11, Average precision was 0.38 Other off-the-shelf ML models, including a deep neural network, were trained on the same task, with performance generally inferior to the Gradient Boosting Machine (GBM), or in the case of the deep neural network, similar When trained and evaluated on a smaller cohort of 300,000 hospitalizations, performance metrics were similar: AUC 0.75, BSL 0.11. The most impactful features included (ranked from the most to the least important): primary diagnosis, days between the current admission and the previous discharge, number of past admissions, LOS, total emergency department visits in the past 6 months, number of reported comorbidities, admission source, discharge disposition, and Body Mass Index (BMI) on admission and discharge, as well as others. Including more than the top ten variables in the model did not improve predictive power for the cohort overall but does allow for more specific rationale for prediction for certain patients, as well as examination of feature interactions for further exploration.

In order to examine possible changes in causes of readmission risk as a function of time from discharge, we predicted readmission risk for several readmission thresholds and calculated SHAP (SHapley Additive exPlanation) for each. SHAP values for 3- and 7-day readmission are shown in Supplementary Fig. 5a, b, respectively. For example, 7-day readmission risk prediction achieved AUC of 0.70 with a BSL of 0.05 (Table (Table2).2). The most impactful feature remained primary diagnosis, but other features played more important roles—e.g., BlockGroup rose to second most important variable (from ninth), number of emergency department visits in the past 6 months rose to third importance from fourth, admission blood counts increased in importance, and insurance provider rose to eighth from twelfth. BMI on admission fell several places, and BMI on discharge no longer features in the top variables. The BMI variables are unique in that missing values tend to be important, in addition to extreme values, perhaps correlating with disease burden and/or hospital practices that could be further investigated.

LOS was predicted in terms of the number of days and was binarized at various thresholds. LOS in days was predicted poorly, within 3.97 days measured by root mean square error (RMSE; average LOS 2.94–3.71 days). LOS over 5 days was predicted with an AUC of 0.84 and a BSL of 0.15 (calibration curve shown in Average precision was 0.70. When trained and evaluated on a cohort of 300,000 patients, performance was similar: AUC 0.81 and BSL 0.17. Other ML models, including a deep neural network, were trained on the same task, with performance generally inferior to the GBM (see Supplementary Fig. 2 and Supplementary Table 1). The most impactful features included the type of admission, primary diagnosis code, patient age, admission source, LOS of the most recent prior admission, medications administered in the hospital in the first 24 h, insurance, and early admission to the intensive care unit, Impactful features for LOS at thresholds of 3 and 7 days. The AUC did not differ in these time points compared to 5 days. Given that primary diagnosis is often assigned late in the hospital encounter or even after discharge, we trained the LOS models with and without this feature for comparison. Results are shown. Overall, predictive performance was decreased, as expected. AUC for LOS > 5 days was 0.781, BSL was 0.173, and average precision was 0.640.