**Customer Classification Using the Tsetlin Machine and Machine Learning Models**

**Introduction**

In today’s competitive modern business environment, understanding customer behavior is important for uderstanding the customers and improving thier satisfaction and targeting the right audience with the use of personalized marketing strategies. One way to achieve such thing is by segmenting customers into various categories based on their way of life and characteristics, such as demo, income, and etc. Effective customer segmentation helps businesses identify rich and important customers and design suitible marketing campaigns leading to better customer retention and increased profits for the company.

**Objective of the Project**

The objective of this project is to use tools like machine learning techniques to classify customers into different categories using a binary classification approach. The focus will be on applying the Tsetlin Machine that is a new machine learning model which is gaining attention for its simplicity especially in classification tasks. To evaluate its usage and effectiveness, we will also compare its performancebelow with more traditional machine learning models like Logistic Regression, Random Forest, and Support Vector Machines .

We will be using a telecomunication dataset in this project. With the setup here our aim is to demonstrate how the Tsetlin Machine performs on a real-world dataset in comparison to established models offering insights into its strengths, capabilities and limitations.

**Dataset Preparation**

In here we will apply some dataset preparation commands that are necessary steps for preparing the dataset before applying machine learning models.

**Uploading the data set**

First we apload the dataset in jupyter.

A screenshot of a computer

Description automatically generated

**Data Cleaning**

We start by checking for missing data that could negatively impact our tselien model performance. Common cleaning steps include steps like removing rows with missing values or even removing outliers from the dataset.

Checking for missing values: with the command below, we have checked for missing values and the result shows their aren’t any missing values.

A screenshot of a computer

Description automatically generated

**Feature Selection**

Feature selection means choosing which variables to include in the model. Not all features may be relevant to the dataset and some may need to be transformed to be used. We’ll review the dataset and choose relevant features that can be important later for tselien model.

A screenshot of a computer

Description automatically generated

**Data Transformation**

The Tsetlin Machine works with binary inputs, so we will convert some of our categorical features into numerical or binary form. For example we will transform the custcat variable into a binary format.

A screen shot of a computer code

Description automatically generated

**Train-Test Split**

then we will split the data into training and testing datasets. This allows us to train the model on one portion of the data and evaluate its performance on other data. We will make the dataset into two parts one for training the models and the other for testing their performance.

A screenshot of a computer program

Description automatically generated

**Tsetlin Machine Implementation**

The main part of this project in hand is implementing the Tsetlin machine for a binary classification task. The Tsetlin machine is a machine learning model that is gaining popularity for its efficiency , ease and importance particularly in classification problems.it works by learning patterns through logical rules (AND, OR, NOT) instead of relying on old numerical optimization techniques.

**Binarization of Features**

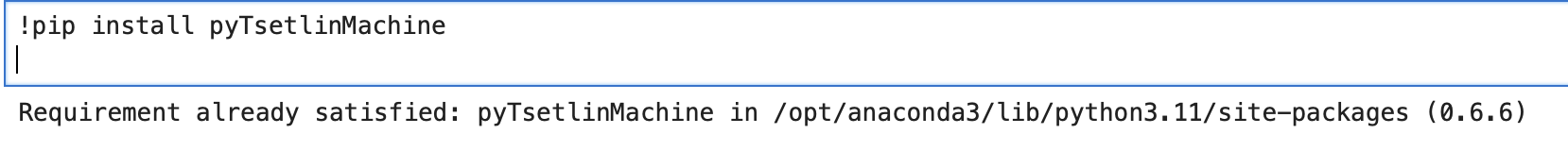
Since the Tsetlin Machine works with binary inputs, we need to change them in binariyation features. This means converting continuous numerical values in the dataset like income and age into binary form using methods such as thresholding.

A screenshot of a computer

Description automatically generated

**Install and Import Tsetlin Machine Library**

To implement it, we will use the pyTsetlinMachine library, a Python wrapper for the Tsetlin Machine algorithm in jupyter.



**Installing and Importing Tsetlin Machine**

In the code below, we install tselin toolls in library required for Tsetlin Machine classification. After installation, from we import the MultiClassTsetlinMachine class from the library, enabling multi-class classification. The help(multiclasstsetlinmachine) function provides detailed documentation on how to use this class including its methods and ways.

A screenshot of a computer program

Description automatically generated

**Loading the Dataset**

In the code below, we import pandas as pd which is used for data manipulation. The dataset is loaded into a pandas dataframe using coomond below where filepath contains the path to the dataset (Telecust1.csv.xls). This step loads the data so that it can be trained and used for model training and evaluation of the dataset.

A screen shot of a computer code

Description automatically generated

**Extracting Features and Target**

the code here features X are extracted by dropping the custcat column from the dataset using the commond df.drop. The target variable y is extracted from by selecting the custcat column from the dataframe. These features below and target variables will be used in model training, where X will represent the input data and y represents the labels the model will predict.

A screenshot of a computer

Description automatically generated

**Binary Classification for Target Variable**

In the code below the custcat column is again converted into a binary classification problem by applying a function that converts values like A to 1 and all other values to 0 using other commond. This creates a binary target variable y binary, which is printed to verify that the transformation has been applied correctly on here.

A computer screen shot of a computer code

Description automatically generated

**Binarizing the Features**

In here the Binarizer class from sklearn.preprocessing is used to convert the contaenuing features into binary values. The method applies the binarization, which assigns binary values 0 or 1 based on a threshold. In this case the threshold is determined automatically by the Binarizer. The transformed feature matrix is printed to verify the results.

A screenshot of a computer program

Description automatically generated

**Predicting and Calculating Accuracy for tsetlin machine**

In the code below after we train a tsetlin machine model predictions are made on the test dataset using predict commond. The accuracy\_score() function from sklearn is then used to calculate the models accuracy by comparing the predictions y\_pred to the actual test labels below in y\_test. The accuracy is printed as a % to show the models performance on the dataset in hand.

A screen shot of a computer code

Description automatically generated

**Training and Evaluating Logistic Regression**

Now a Logistic Regression model is initialized by us using LogisticRegression(max\_iter=200) and trained on a training dataset. Predictions are then made by us on the test dataset using accuracy. the logistic regression model is calculated and the result is printed to evaluate the model's performance.

A screenshot of a computer program

Description automatically generated

**Training and Evaluating Random Forest**

In the code below, a Random Forest classifier is initialized by the essar with 100 trees using randomforestclassifier commond. Then the model is trained on the training data. Then we do the predictions for the test and the models accuracy is calculated and printed using accuracy score to assess its performance. The graph below is avisual repesentation.

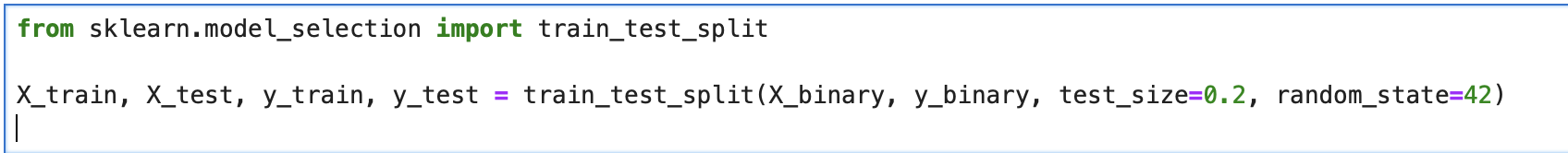
A screen shot of a computer program

Description automatically generatedA screenshot of a graph

Description automatically generated

**baseline models and comparison**

In baseline models and comparison we will want to implement and compare several baseline models to see how they perform against the model tsetlin machine. this usually involves using standard classification algorithms like logistic regression and random forest, which we have already have in our previous steps.it starts with spliting into training and testing sets.



import necessary libraries.

A screen shot of a computer code

Description automatically generated

Loading the data and preprocess: we make sure data is loaded and preprocessed correctly.

A screenshot of a computer program

Description automatically generated

Train Models: below we will train the logistic regression and random forest models.

A screenshot of a computer program

Description automatically generated

**Make predictions and evaluate**

A screenshot of a computer program

Description automatically generatedA graph of a logistic regression

Description automatically generatedA graph with numbers and a blue square

Description automatically generated with medium confidence

**Results and Analysis**

The models trained in this studyby us performed similarly. Both logistic regression and random Forest showed an accuracy of 70%. While this shows they are able tofunction on most data correctly, it doesnt tell the whole story here about how they handled different classes.

The confusion matrices for both models revealed some interesting results. For logistic regression the model was good at accuratly predicting where A was not the case (True Negatives). However it struggled with identifying A with failing to correctly classify any cases of this class (False Negatives). Then resulted in a recall of 0 for class A meaning the model missed all instances of it in the test data.

also the random forest model showed the same behavior. It excelled at predicting not A but it failed to detect any cases of A. with this info it suggests that both models are biased towards the bigger class of the dataset and couldn't capture the smaller class.

**Classification Report Insights**

Looking at the classification reports, the trends became even clearer. Both of the models that we used had a accorate precision of 70% for not A, but a non accurate precision of 0% for A. also recall for class A was also 0% reinforcing that they were unable to identify this class. The F1score for A was very very low which highlights that these models are not effective in scenarios like above situation. the reasons for these results could be:

Class Imbalance

Feature Representation

Model Complexity

To improve the performance of models in hand some strategies can be considered by us:

address class imbalance: Use techniques like making more sampls for the minority class (such as SMOTE) or applying class weights during model training to help the models focus on predicting A for the minority.

Feature engineering: we can explore new features or transformations that might better take in account the differences between the two classes.

Advanced models: we can try using more advance models like gradient boosting or deep learning, which could make us better understand complex patterns in the data.

**Conclusion**

In this project, we focused on customer classification using deferent metods of machine learning, with an emphasis on the tsetlin machine alongside models like logistic regression and random forest models. The tsetlin machine offered possible rules while the other models achieved the same accuracy of around 70% still their were deferences. Binarizing the features helped simplify the data in particularly benefiting the tsetlin machine. Using genetic algorithms for feature selection improved the performance of models achieving a more accurate score of 0.734. in conclusion the setlin machine's interpretability stood out here from the other models, highlighting important features for classification.

**Github link**

[**https://github.com/alijaweddelawari/ai-studio-retake**](https://github.com/alijaweddelawari/ai-studio-retake)

**Dataset Resource**

Pratham Tripathi, license CC0: Public Domain, kaggle.com

being able to download it, somehow Mac terminal did not allow it to run.



Assessment Submission Form

|  |  |
| --- | --- |
| **Student Number**  (If this is group work, please  include the student numbers of all group participants) | GH1024093 |
| **Assessment Title** | Customer Classification Using the Tsetlin Machine and Machine Learning Models |
| **Module Code** | B143 |
| **Module Title** | AI STUDIO |
| **Module Tutor** | [Amirhossein Jamalian](https://study.gisma.com/courses/3937/users/2053) |
| **Date Submitted** | 04/10/2024 |

**Declaration of Authorship**

I declare that all material in this assessment is my own work except where there is clear acknowledgement and appropriate reference to the work of others.

I fully understand that the unacknowledged inclusion of another person’s writings or ideas or works in this work may be considered plagiarism and that, should a formal investigation process confirms the allegation, I would be subject to the penalties associated with plagiarism, as per GISMA Business School, University of Applied Sciences’ regulations for academic misconduct.

Signed………………Ali Jawed Delawari……………. Date ………04/10/2024………………