## Report

## **Objective:**

## This study aims to carry out an extensive analysis and compare the efficacy of different machine learning techniques along three unique datasets. Evaluating whether distinct implementations of machine learning algorithms achieve the desired effectiveness across application-specific data as well as revealing the most optimal implementation for a particular design in empirical designs and evaluation measures is the goal of the assessment. We will use two machine learning techniques for each of the datasets undergoing analysis. The study plans to identify relevant features of these datasets and assess their impact on the performance of these machine learning techniques which may help to enhance our knowledge of their some standard areas of applicability and limitations. This incorporate critical examination as well as data mining methods, extraction and processing of data and building as well as evaluating machine learning model in different problem domains.

**Datasets and their sources:**

1. Employee data Classification: <https://www.kaggle.com/datasets/tawfikelmetwally/employee-dataset>
2. Shop customer dataset: <https://www.kaggle.com/datasets/datascientistanna/customers-dataset>
3. Steel Industry energy consumption: https://www.kaggle.com/datasets/csafrit2/steel-industry-energy-consumption/data

**Identification of Dataset Characteristics which May Impact the Performance of ML Techniques:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Employee dataset** | **Shop Customer dataset** | **Steel industry energy consumption dataset** |
| No. Of dependent variable | 1 | 1 | 1 |
| No. Of independent variable | 8 | 7 | 9 |
| Total number of records | 4653 | 2000 | 35040 |
| Datatypes of combination | Binary, Categorical, Numerical, Textual | Binary, Categorical, Numerical | Binary, Nominal, Categorical, Numerical |
| No. Of missing values | 0 | 35 | 0 |
| Technique to deal with missing value | No missing values found | Removed null values (very minimal %) | No missing values found |
| No. Of duplicates rows | 1889 | 0 | 0 |
| Technique to deal with duplicate rows | Removed by drop\_duplicates() | None | None |
| Total No. Of outliers | 0 | 0 | 765 |

1. **Implementation of machine learning techniques on Employee data :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ML Technique** | **Accuracy** | **MSE** | **RMSE** | **MAE** | **R2** |
| **SVM using Sklearn** | 0.78072 | 0.21927 | 0.46827 | 0.2192 | 0.08542 |
| **SVM using LibSVM** | 0.80470 | 0.19529 | 0.44192 | 0.19529 | 0.13064 |
| **Decision tree using Sklearn** | 0.68173 | 0.31826 | 0.56414 | 0.31826 | -0.27136 |
| **Decision tree using Xgboost** | 0.75768 | 0.24231 | 0.49225 | 0.24231 | -0.00503 |

**Number of records in class 0-1676**

**Number of records in class 1-1088**

**Q1: The two-implementation of identically named technique perform differently. The reason is explained below.**

**Q2: Now we will discuss about SVM and Decision tree, the two Algorithms that we have implemented :**

**a )SVM**

Accuracy gap between the two SVM implementations majorly depends on the gamma parameter.

SVM (Gamma='auto'): Accuracy = 0. 7884

LibSVM (Gamma='scale'): Accuracy = 0. 8047

The implementation of LibSVM run better partly due to gamma='scale' parameter that performs the best in keeping the decision boundary optimal.

**b )Decision tree**

The difference in accuracy differences arise because the DecisionTreeClassifier using Sklearn whose Accuracy is 0.6817 uses only one decision tree for the model, hence, it has a risk of overfitting or underfitting. The XGBClassifier whose Accuracy is 0.7915 is an ensemble method that builds detached trees each implying and subsequently puts them all together in an iterative process. It is done to make sure that overfitting problem and underfitting problem are dealt with properly to improve the prediction accuracy.

1. **Implementation of machine learning techniques on Shop customer dataset :**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ML Technique** | **Accuracy** | **MSE** | **RMSE** | **MAE** | **R2** |
| **SVM using Sklearn** | 0.48644 | 0.48644 | 0.69745 | 0.48644 | -0.94596 |
| **SVM using LibSVM** | 0.51186 | 0.51186 | 0.71544 | 0.511864 | -1.04766 |
| **Decision tree using Sklearn** | 0.50338 | 0.49661 | 0.70470 | 0.49661 | -0.98664 |
| **Decision tree using Xgboost** | 0.48644 | 0.48644 | 0.69745 | 0.48644 | -0.94596 |

**Number of records in class 0-979-high**

**Number of records in class 1-986-low**

**Q1: The two-implementation of identically named technique perform differently. The reason is explained below.**

**Q2: Now we will discuss about SVM and Decision tree, the two Algorithms that we have implemented :**

**a )SVM**

Accuracies are different for SVC because of the fact that sklearn internally scales features and adjust parameters while LibSVM require explicit feature scaling and parameter set. Using the same solver (linear kernel), the non-identical elements cause different outcomes.

SVC (scikit-learn): Accuracy = 48.64%

LibSVM SVM: Accuracy = 51.19%

**b )Decision tree**

The DecisionTreeClassifier has an accuracy of 50% with. 34% of the build one decision tree, the XGBClassifier achieves an accuracy of 48%. 64% of the model is based on a tree ensemble that uses gradient boosting. However, XGBoost usually does better but if the hyperparameters are not optimized or the dataset is peculiar, the higher accuracy may not be the case. The main similarity between them is that decision trees are at their core, but the implementation and optimization strategies determine the differences.

1. **Implementation of machine learning techniques on Steel Industry energy consumption dataset :**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ML Technique** | **MSE** | **RMSE** | **MAE** | **R2** |
| **SVM using Sklearn** | 0.01039 | 0.10191 | 0.06822 | 0.98966 |
| **SVM using LibSVM** | 0.00968 | 0.09837 | 0.06717 | 0.99036 |
| **Decision tree using Sklearn** | 0.00857 | 0.09255 | 0.04595 | 0.99147 |
| **Decision tree using Xgboost** | 0.00716 | 0.08462 | 0.04827 | 0.99287 |

**Q1: The two-implementation of identically named technique perform differently. The reason is explained below.**

**Q2: Now we will discuss about SVM and Decision tree, the two Algorithms that we have implemented :**

**a )SVM**

Although both of the techniques use the same SVR algorithm, the disparities in the method of feature scaling, parameter tuning, as well as optimization strategies result in the different performance.

scikit-learn SVR R² Accuracy = 0.9897

LibSVM SVR R² Accuracy = 0.9904

scikit-learn SVR Automatically scales features. High-level wrapper simplifies usage but limits parameter tuning flexibility. LibSVM SVR Requires manual feature scaling. Direct access to the LibSVM API allows for finer parameter tuning.

**b )Decision tree**

A drawback of sklearn's DecisionTreeRegressor is that it employs a single decision tree, which may lead to overfitting of data. Instead of that, XGBRegressor that uses a gradient boosting technique to create multiple trees sequentially to minimize errors and results in good generalization of the model and high accuracy.