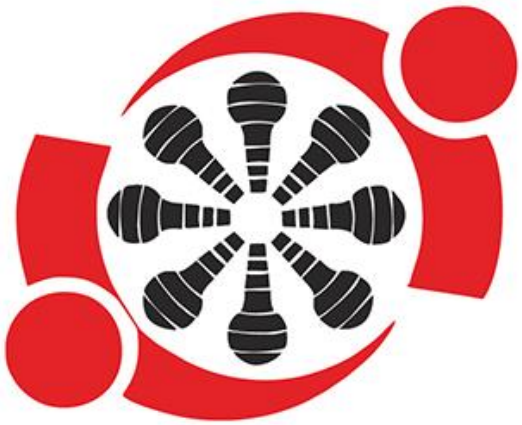

INNOMATİCS
RESEARCH LABS

Machine Learning - Project Report Document

Student Name	Shiva Sai Chakradhar Vurukonda
Batch	AI Elite 18
Project Name	Drugs consumption
Project Domain	Health Care
Type of Machine Learning	Supervised Learning
Type of Problem	Classification
Project Methodology	CRISP-DM

Stages Involved	Data cleaning ,Data preprocessing , Model building ,Evalution .

Stage 1: Business Understanding:

“Given a multilabel dataset containing records for respondents, where each respondent has demographic features (such as Age, Gender, Education, Country, and Ethnicity), personality traits (Nscore, Escore, Oscore, AScore, Cscore), impulsivity (Impulsive and SS), and usage frequency for various drugs (Alcohol, Amphetamines, Amyl, Benzos, Caffeine, Cannabis, Chocolate, Cocaine, Crack, Ecstasy, Heroin, Ketamine, Legal highs, LSD, Meth, Mushrooms, Nicotine, and Semeron), the task is to predict the drug usage based on the available features. The output column represents seven classes: ‘Never Used,’ ‘Used over a Decade Ago,’ ‘Used in Last Decade,’ ‘Used in Last Year,’ ‘Used in Last Month,’ ‘Used in Last Week,’ and ‘Used in Last Day’.”

Stage 2: Data Collection and Understanding

a) **Data Collection:**
The Dataset is collected from the Kaggle

b) **Data Understanding:**
The data contains of 7 float data types and 27 object data types where the input columns are from

S No 1 to 30 and the output columns is VSA (drug consumption period)

S No	Feature Name	Data Type
1	Age	object
2	Gender	object
3	Education	Object
4	Country	Object
5	Ethicity	Object
6	Nscore	Float64
7	Escore	Float64
8	Oscore	Float64
9	Ascore	Float64
10	Cscore	Float64

11	Impulsive	Float64
12	SS	Float64
13	Alcohol	Object
14	Amphet	Object
15	Amyl	Object
16	Benzos	Object
17	Caff	Object
18	Choc	Object
19	Coke	Object
20	Crack	Object
21	Ecstasy	Object
22	Herion	Object
23	Ketamine	Object
24	Legalh	Object
25	LSD	Object
26	Meth	Object
27	Cannabis	Object
28	Mushrooms	Object
29	Nicotine	Object
30	Semer	Object
31	VSA	Object

Stage 3: Data Preparation

a) Exploratory Data Analysis:

S No	Type	Feature Names	Observation
1	Missing Values	0	No
2	Duplicates	0	No

b) Data Cleaning/wrangling:

applied IQR on the data set where there after got 84 null values over all dropped them

S no	Type of Cleaning	Technique	Feature Name	Reason
1	Encoding	Label encoding	'Age', 'Gender', 'Education', 'Country', 'Ethnicity','Alcohol', 'Amphet', 'Amyl', 'Benzos', 'Caff', 'Cannabis','Choc', 'Coke', 'Crack', 'Ecstasy', 'Heroin', 'Ketamine', 'Legalh', 'LSD', 'Meth', 'Mushrooms', 'Nicotine', 'Semer'	Because the onehotencoding ocuppys more space and it give ranking

c) **Feature Selection:**

Selected Features:

This are the columns selected in the data set after applying future selection Age', 'Education', 'Nscore', 'Escore', 'Oscore', 'AScore', 'Cscore', 'Impulsive', 'SS', 'Amphet', 'Benzos', 'Coke', 'Legalh', 'LSD', 'Meth

Stage 4: Model Building:

- Logistic Regression: A linear model that predicts the probability of a binary outcome based on input features by fitting a logistic function.
- Decision Tree Classifier: A tree-structured model that splits the data into subsets based on the value of input features, aiming to maximize information gain or purity in each node.
- k-Nearest Neighbors (k-NN): k-NN classifies a data point based on the majority class among its k-nearest neighbors in the feature space. It is simple and intuitive, requiring no explicit training phase, but can be computationally expensive and sensitive to noisy data and irrelevant features.
- Random Forest Classifier: An ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes as the prediction for classification tasks.
- SVC (Support Vector Classifier): A classifier that separates data points into different classes by finding the hyperplane with the maximum margin between classes in the feature space
- Naive Bayes :Naive Bayes classifiers apply Bayes' theorem with the assumption of independence between features. Despite the "naive" assumption, it works surprisingly well in many real-world applications, particularly for text classification and spam filtering. It's fast and efficient with small data requirements.
- Gradient Descent : Gradient Descent is an optimization algorithm used to minimize the cost function in machine learning models. It iteratively adjusts the model parameters in the direction of the steepest descent of the cost function, calculated using the gradient

S No	Type of Problem	Algorithm Name
1	Classification	Logistic Regression
2	Classification	Decision Tree
3	Classification	K_NN
4	Classification	Naïve Baye's
5	Classification	Gradient Descent

6	Classification	Random Forest
7	Classification	Support Vector Machine

Stage 5: Model Training:

The evaluation metric used for all algorithms during model training was accuracy, precision , Recall , F1_score which measures the proportion of correctly classified instances over the total number of instances.

S No	Algorithm Name	Hyper-parameter tuning	Metric used for Evaluation
1	Logistic Regression	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score
2	Decision Tree	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score
3	K_NN	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score
4	Naïve Baye's	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score
5	Gradient Descent	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score

6	Random Forest	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score
7	Support Vector Machine	Grid Search CV Random Search CV	Accuracy_score precision_score recall_score f1_score

Stage 5: Model Evaluation:

Accuracy: The proportion of correctly predicted instances out of the total instances.

Precision: The proportion of correctly predicted positive instances out of all predicted positive instances.

Recall: The proportion of correctly predicted positive instances out of all actual positive instances.

F1-score: The harmonic mean of precision and recall, providing a balance between the two metrics.

The table of Random Search CV:

S No	Algorithm Name	Metric Score
1	Logistic Regression	Accuracy: 0.758 Precision: 0.649 Recall: 0.758 F1-score: 0.685
2	Decision Tree	Accuracy: 0.774 Precision: 0.599 Recall: 0.774 F1-score: 0.675
3	K_NN	Accuracy: 0.756 Precision: 0.661 Recall: 0.756 F1-score: 0.690
4	Naïve Baye's	Accuracy: 0.676 Precision: 0.709

		Recall: 0.676 F1-score: 0.686
5	Gradient Descent	Accuracy: 0.774 Precision: 0.599 Recall: 0.774 F1-score: 0.675
6	Random Forest	Accuracy: 0.763 Precision: 0.625 Recall: 0.763 F1-score: 0.678
7	Support Vector Machine	Accuracy: 0.774 Precision: 0.599 Recall: 0.774 F1-score: 0.675

Table of the Grid Search CV

S no	Algorithm Name	Metric_Score
1	Logistic Regression	Accuracy: 0.783 Precision: 0.695 Recall: 0.783 F1-score: 0.705
2	Decision Tree	Accuracy: 0.787 Precision: 0.620 Recall: 0.787 F1-score: 0.693
3	K_NN	Accuracy: 0.752 Precision: 0.659 Recall: 0.752 F1-score: 0.697
4	Naïve Baye's	Accuracy: 0.663 Precision: 0.720 Recall: 0.663 F1-score: 0.686
5	Gradient Descent	Accuracy: 0.785 Precision: 0.619 Recall: 0.785 F1-score: 0.692
6	Random Forest	Accuracy: 0.787 Precision: 0.620 Recall: 0.787

		F1-score: 0.693
7	Support Vector Machine	Accuracy: 0.787 Precision: 0.620 Recall: 0.787 F1-score: 0.693

Challenges Faced:

The main challenge was the data understanding where there were 24 object columns and 7 float data type coloumns
when creating the pipeline I was stuck which parameters should be used in the each algorithm by the help of Toward Data Science website I learned from there and kept the parameters

Conclusion:

Based on the project evaluation, Naive Bayes exhibited the shortest training time at 0.01 seconds, with Decision Tree showing the fastest prediction time at 0.004 seconds. Conversely, Support Vector Machine (SVM) required the longest training time at 1.43 seconds, while K-Nearest Neighbors (KNN) had the highest prediction time at 0.095 seconds.

The **Random Forest** classifier emerged as the best-performing model with a best score of 0.783. It demonstrated robust performance across various evaluation metrics, particularly in terms of accuracy and recall, making it suitable for the classification task at hand. Both Random Search CV and Gradient Search CV confirmed its superiority among the evaluated models.