KNN CLASSIFICATION ALGORITHM

PROJECT REPORT

[SUBMITTED IN PARTIAL FULFILLMENT]

AS A PART OF CURRICULUM OF B.SC. (HONS.) COMPUTER SCIENCE

2019-2022

SHYAMA PRASAD MUKHERJI COLLEGE FOR WOMEN
UNIVERSITY OF DELHI

SUBMITTED BY

HIMANI BARGALI (19075570013)

B.Sc. (H) Computer Science (III YEAR)

UNDER THE SUPERVISION OF,

DR. SHWETA TYAGI

ASSISTANT PROFESSOR

DEPT. OF COMPUTER SCIENCE

SHYAMA PRASAD MUKHERJI COLLEGE FOR WOMEN

(UNIVERSITY OF DELHI)

PUNJABI BAGH (WEST), NEW DELHI-110026

CERTIFICATE

This is to certify that the project work entitled “KNN Classification Algorithm” is a bonafide work carried out by Himani Bargali in partial fulfillment of the requirement for the award of the degree of B.Sc. (H) in Computer Science, Shyama Prasad Mukherji College for Women, University of Delhi, New Delhi during the academic year 2019-2022. The project has been approved as it satisfies the requirements in respect of project work prescribed for the said degree.

Dr. Jaya Gera — (Head of Department, Computer Science)

Dr. Shweta Tyagi — (Supervisor)

DECLARATION

I Himani Bargali, student of III year, B.Sc. (H) in Computer Science, Shyama Prasad Mukherji College for Women, University of Delhi, hereby declare that the project report entitled "KNN Classification Algorithm" submitted by me to the University of Delhi, during the academic year 2019-2022, is a record of original work carried out by us under the guidance of Dr. Shweta Tyagi, professor of department of computer science, Shyama Prasad Mukherji College for Women, University of Delhi, New Delhi. This project work is submitted in partial fulfillment of the requirement for the award of the degree of B.Sc. (H) in Computer Science.

We further declare that the work reported in this project is original and has not submitted, in part or full, to any other university or institution for the award of any other degree.

DATE: 28/03/2022 **Himani Bargali (19075570013)**

ACKNOWLEDGEMENT

It is our privilege to express our most sincere regards to our project supervisor Dr. Shweta Tyagi for their valuable inputs, able guidance, encouragement, whole-hearted co-operation and constructive criticism throughout the duration of our project.

We express sincere thanks to our teacher, Dr. Prof. Jaya Gera (Head of Department, Computer Science) for encouraging and allowing us to present the project on the topic "KNN Classification Algorithm" at our department premises for the partial fulfillment of the requirements leading to the award of B.Sc. degree.

We take this opportunity to thank all our lecturers who have directly or indirectly helped our project. We pay our respects and love to our parents and all other family members and friends for their love and encouragement throughout our career.

I Thank You All.

Himani Bargali (19075570013)

TABLE OF CONTENTS

Abstract..... 6

1. Introduction..... 7
2. KNN Classification Algorithm..... 8
3. Advantages..... 10
4. Disadvantages..... 10
5. Application & uses..... 10
6. Dataset..... 11
7. Experiment & Results..... 13

Conclusion..... 21

References..... 22

Appendix..... 27

ABSTRACT

Data mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analyzing data patterns in large batches of data using one or more software. Classification is a data mining function that assigns items in a collection to target categories or classes. One of the most used classification techniques is K Nearest Neighbor (KNN) algorithm. It is an effortless but productive data mining algorithm. It is effective for classification as well as regression. However, it is more widely used for classification prediction. KNN groups the data into coherent clusters or subsets and classifies the newly inputted data based on its similarity with previously trained data. The input is assigned to the class with which it shares the nearest neighbors. In this project, we propose KNN algorithm on Adult Income Dataset. The experimental results show that the proposed KNN classification works well in terms of accuracy and efficiency. Our task is to analyze the dataset and predict whether the income of an adult will exceed 50k per year or not by developing a supervised model.

1. INTRODUCTION

According to Han and Kamber, "Data Mining is known to be a part of knowledge discovery (KDD) process in which data is analyzed and summarized from different perspectives and converted into useful information. It helps in extracting the hidden and valid data which has the potential of being transformed into useful information." It is similar to machine learning process and can also be termed as supervised learning process. Supervised learning is the process of data mining for deducing rules from training datasets. A broad array of supervised learning algorithms exists, every one of them with its own advantages and drawbacks. In classification the first step is to divide the data in two portions known as training set and testing set. In these datasets, one attribute must be necessarily defined as class [5].

KNN (k-Nearest Neighbors) is one of many (supervised learning) algorithms used in data mining and machine learning, it's a classifier algorithm where the learning is based "how similar" is a data (a vector) from other. In the case of KNN, indeed it doesn't compare the new (unclassified) data with all other, actually it performs a mathematical calculation to measure the distance between the data to makes the classification. This calculation can be any calculation that measures the distance between two points, for example: Euclidian, Manhattan, Distance-Weighted, etc.

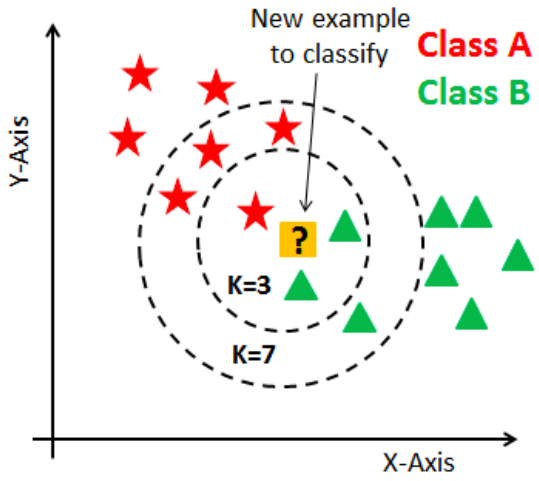

Fig. 1: Pictorial Representation of KNN [6]

KNN is employed to *Adult Income Dataset*. Origin of the dataset: Extraction was done by Barry Becker from the 1994 Census database. The US Adult Census dataset is a repository of 48,842 entries extracted from the 1994 US Census database [4].

The main objective of the dataset is to classify people earning <=50k or >50k based on several explanatory factors affecting the income of a person like Age, Occupation, Education, etc. We have used data cleaning method in this KNN imputation using VIM library (Visualization and Imputation of Missing Values). The descriptive features include 4 numeric and 7 nominal categorical features. The target feature has two classes defined as "<=50K" and ">50K" respectively. The full dataset contains about 45K observations.

Classification accuracy is normally improved through ensemble models like bagging (which averages the prediction of a number of classification models), boosting (it uses the voting scheme over a number of classification models), or a combination of classifiers from different or same families. Therefore, we propose optimization for classification methods and prove through experimental results that our model improves the classification accuracy [5].

2. *KNN* Classification Algorithm

The *KNN* working can be explained on the basis of Adult Income Dataset:

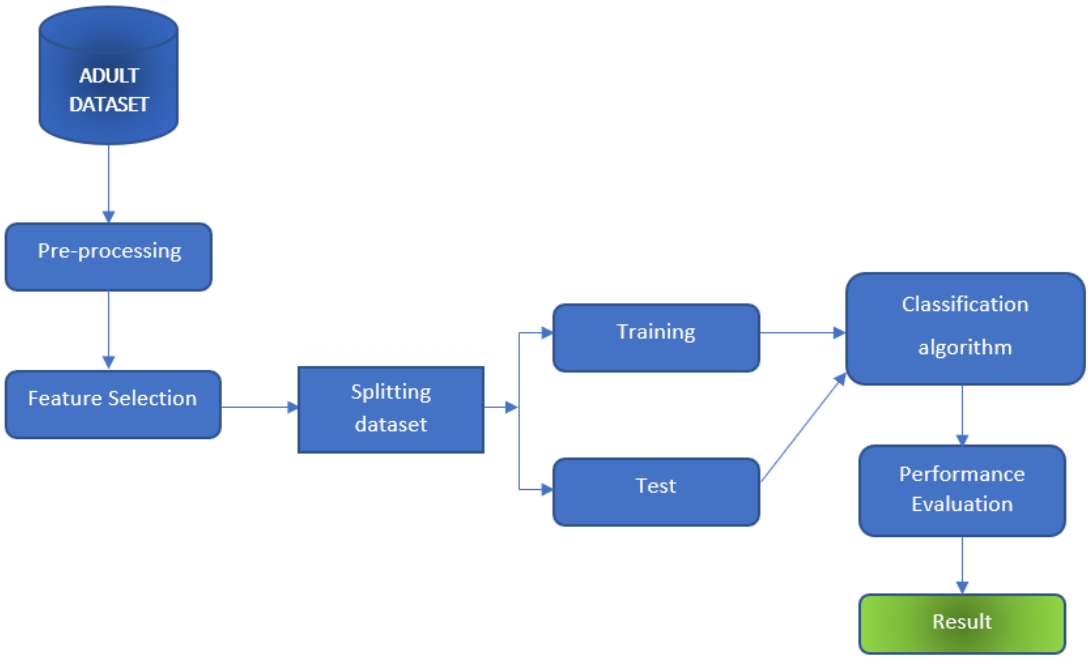

Fig. 2: Proposed Methodology

In the training phase, the model will store the data points. In the testing phase, the distance from the query point to the points from the training phase is calculated to classify each point in the test dataset. Various distances can be calculated, but the most popular one is the Euclidean distance (for smaller dimension data).

Euclidean distance between a query point (q) and a training data point (p) is defined as:

$$d(\mathbf{p},\mathbf{q}) = d(\mathbf{q},\mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

Fig. 3: Euclidean Distance formula

Let’s learn it with an example:

We have 500 N-dimensional points, with 300 being class 0 and 200 being class 1.

The procedure for calculating the class of query point is:

The distance of all the 500 points is calculated from the query point.

Based on the value of K, K nearest neighbors is used for the comparison purpose.

Let’s say K=7, 4 out of 7 points are of class 0, and 3 are of class 1. Then based on the majority, the query point p is assigned as class 0.

In real-world problems, the dataset is separated into three parts, namely, training, validation, and test data. In KNN, the training data points get stored, and no learning is performed. Validation data is to check the model performance, and the test data is used for prediction.

To select optimal K, plot the error of model (error = 1- accuracy) on training as well as on the validation dataset. The best K is where the validation error is lowest, and both training and validation errors are close to each other.

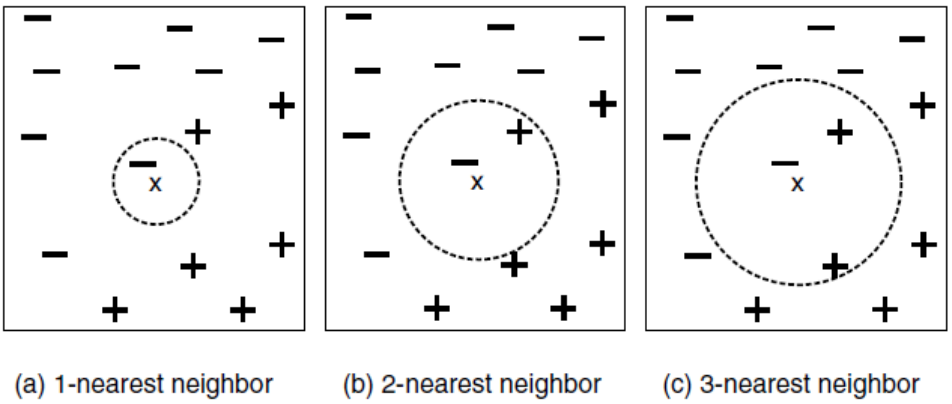

Fig. 4: The 1-, 2-, and 3-nearest neighbours of an instance [7].

The algorithm computes the distance (or similarity) between each test example $z = (\mathbf{x}\_, y\_)$ and all the training examples $(\mathbf{x}, y) \in D$ to determine its nearest-neighbour list, Dz. Such computation can be costly if the number of training examples is large. However, efficient indexing techniques are available to reduce the number of computations needed to find the nearest neighbours of a test example [7].

Algorithm 5.2 The k-nearest neighbor classification algorithm.

1: Let k be the number of nearest neighbors and D be the set of training examples.
2: **for** each test example $z = (\mathrm{x}', y')$ **do**
3: Compute $d(\mathrm{x}', \mathrm{x})$, the distance between z and every example, $(\mathrm{x}, y) \in D$.
4: Select $D_z \subseteq D$, the set of k closest training examples to z.
5: $y' = \underset{v}{\operatorname{argmax}} \sum_{(\mathbf{x}_i, y_i) \in D_z} I(v = y_i)$
6: **end for**

Fig. 5: The k-nearest neighbour classification algorithm [7].

3. ADVANTAGES

i. No Training Period: KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g., SVM, Linear Regression etc.

ii. Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm [8].
iii. KNN is very easy to implement. There are only two parameters required to implement KNN i.e., the value of K and the distance function (e.g., Euclidean or Manhattan etc.) [8].

4. DISADVANTAGES

i. Does not work well with large dataset: In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm [8].
ii. Does not work well with high dimensions: The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.
iii. Need feature scaling: We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.
iv. Sensitive to noisy data, missing values and outliers: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers [8].

5. APPLICATIONS & USES

i. KNN can be used for Recommendation Systems. Although in the real world, more sophisticated algorithms are used for the recommendation system. KNN is not suitable for high dimensional data, but KNN is an excellent baseline approach for the systems. Many companies make a personalized recommendation for its consumers, such as Netflix, Amazon, YouTube, and many more.
ii. KNN can search for semantically similar documents. Each document is considered as a vector. If documents are close to each other, that means the documents contain identical topics.
iii. KNN can be effectively used in detecting outliers. One such example is Credit Card fraud detection.

6. DATASET

The employed *Adult Income Dataset* is an individual's annual income results from various factors. Intuitively, it is influenced by the individual's education level, age, gender, occupation, and etc. This is a widely cited KNN dataset. The dataset contains 15 columns. The possibility in predicting income level based on the individual's personal information is discussed in this dataset [10].

Shown below are a few characteristics of Adult Income Dataset:

i. Class: Income
ii. The income is divided into two classes: <=50K and >50K

iii. Number of attributes: 14
iv. 48842 instances (no. of rows), mix of continuous and discrete values.

KNN needs int/ float values as it calculates the distance between the query point and other points. The dataframe has many columns which are not likely to affect one's income like marital status, race, etc. So, we drop those columns. The remaining columns will still have string values. And string values cannot be converted to float. So, we assign different integer values for different string values accordingly.

Each entry contains the following information about an individual:

Table 1. Description of Attributes of Adult Income Dataset

Attribute	Description	Type	Range
Age	The age of an individual.	Quantitative (Ratio)	17 to 99
Workclass	A general term to represent the employment status of an individual.	Qualitative (ordinal)	Private, Self.emp.not.inc, Self.emp.inc, Federal.gov, Local.gov, State.gov, Without.pay, Never.worked
Fnlwgt	Final weight. In other words, this is the number of people the census believes the entry represents.	Quantitative (Ratio)	Greater than 12285
Education	The highest level of education achieved by an individual.	Qualitative (Ordinal)	Bachelors, Some.college, 11^{th}, HS.grad, Prof.school, Assoc.acdm, Assoc.voc, 9^{th}, $7th.8^{th}$, 12^{th}, Masters, $1st.4^{th}$, 10^{th}, Doctorate, $5th.6^{th}$, Preschool
Educational-num	The highest level of education achieved in numerical form.	Qualitative (Ordinal)	1 to 16
Marital-status	Marital status of an individual. Married.civ.spouse corresponds to a civilian spouse while Married.AF.spouse is a	Qualitative (Ordinal)	Married.civ.spouse, Divorced, never.married, Separated, Widowed, Married.spouse.absent, Married.AF.spouse

	spouse in the Armed Forces.		
Occupation	The general type of occupation of an individual.	Qualitative (Ordinal)	Tech.support, Craft.repair, Other.service, Sales, Exec.managerial, Prof.specialty, Handlers.cleaners, Machine.op.inspct, Adm.clerical, Farming.fishing, Transport.moving, Priv.house.serv, Protective.serv, Armed.Forces
Relationship	Represents what this individual is relative to others. For example, an individual could be a Husband. Each entry only has one relationship attribute and is somewhat redundant with marital status. We might not make use of this attribute at all	Qualitative (Ordinal)	Wife, Own.child, Husband, Not.in.family, Other.relative, Unmarried.
Race	Descriptions of an individual's race.	Qualitative (Ordinal)	White, Asian.Pac.Islander Amer.Indian.Eskimo, Other, Black
Gender	The biological sex of the individual.	Qualitative (Ordinal)	Male, Female
Capital-gain	Capital gains for an individual.	Quantitative (Ratio)	Greater than or equal to 0
Capital-loss	Capital loss for an individual.	Quantitative (Ratio)	Greater than or equal to 0
Hours-per-week	The hours an individual has reported to work per week.	Quantitative (Ratio)	Less than or equal to 112

Native-country	Country of origin for an individual.	Qualitative (Ordinal)	United.States, Cambodia, England, Puerto.Rico, Canada, Germany, Outlying. US, (Guam.USVI.etc) India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican.Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia El.Salvador, Trinadad&Tobago, Peru, Hong, Holand.Netherlands
Income (label)	Whether or not an individual makes more than $50,000 annually.	Quantitative (Ratio)	<=50k, >50k

7. EXPERIMENT AND RESULTS

The main objective of the dataset is to classify people earning <=50k or >50k based on several explanatory factors affecting the income of a person like Age, Occupation, Education, etc.

Let's start by importing some of the required libraries.

I. install.packages("class")

It is used for various functions for classification, including k-nearest neighbor, Learning Vector Quantization and Self-Organizing Maps.

II. install.packages("caret")

The caret package (short for Classification and Regression Training) contains functions to streamline the model training process for complex regression and classification problems.

III. install.packages("editrules")

Facilitates reading and manipulating data restrictions on numerical and categorical data.

IV. install.packages("VIM")

It is used for the visualization of missing and/or imputed values, which can be used for exploring the data and the structure of the missing and/or imputed values.

V. install.packages("ROCR")

ROCR is a package for evaluating and visualizing the performance of scoring classifiers in the statistical language R.

Load the packages:

We use the keyword *library* to load the packages.

I. library(class)
II. library(caret)
III. library(editrules)
IV. library(VIM)
V. library(ROCR)

Loading the data and performing Exploratory Data Analysis **(**EDA**):**

EDA is a statistical approach or technique for analyzing data sets in order to summarize their important and main characteristics generally by using some visual aids.

Firstly, read the CSV file that is Adult dataset and then print the first 6 observations in dataset to check the dataset.

Generating Rule set and checking if any rules are violated:

Create a rule set named (rules) in the file (KNN_rules.txt) to check whether the ruleset is violated by the data or not as shown in Fig 5.

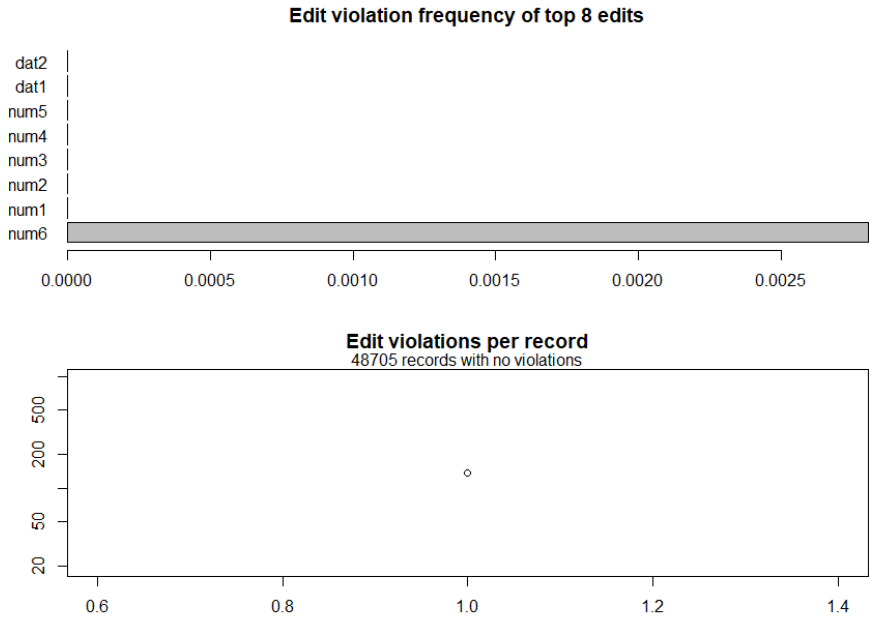

Fig. 6: Violation Graph

EDA of the dependent variable:

The original dataset contains a distribution of 24.08% entries labeled with >50k and 75.91% entries labeled with <=50k. The following graph and statistics pertain to the original dataset.

- library(ggplot2)

ggplot2 is a plotting package that provides helpful commands to create complex plots from data in a data frame.

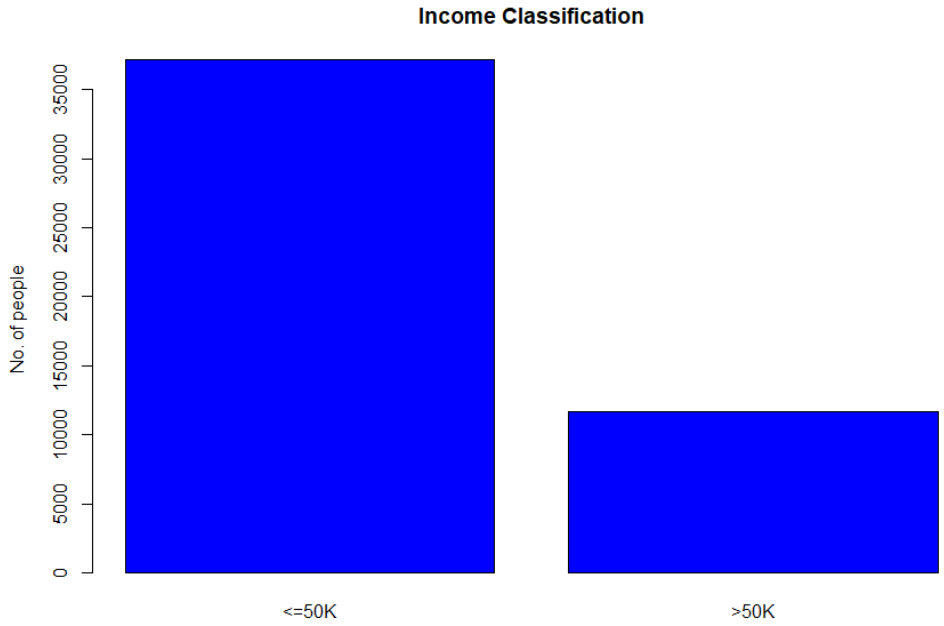

Fig. 7: Income Classification

Converting '?' to NA:

Some of the values in columns are marked as '?' as shown in Fig. 8. Convert these to NA while loading the data itself.

kNN_project_19075570013.R* | dataset | dataset1

Filter

	age	workclass	fnlwgt	education	educational.num	marital.status	occupation	relationship	race
1	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black
2	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
3	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White
4	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black
5	18	?	103497	Some-college	10	Never-married	?	Own-child	White
6	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White
7	29	?	227026	HS-grad	9	Never-married	?	Unmarried	Black
8	63	Self-emp-not-inc	104626	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White
9	24	Private	369667	Some-college	10	Never-married	Other-service	Unmarried	White
10	55	Private	104996	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White
11	65	Private	184454	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
12	36	Federal-gov	212465	Bachelors	13	Married-civ-spouse	Adm-clerical	Husband	White
13	26	Private	82091	HS-grad	9	Never-married	Adm-clerical	Not-in-family	White
14	58	?	299831	HS-grad	9	Married-civ-spouse	?	Husband	White

Showing 1 to 14 of 48,842 entries, 15 total columns

Fig. 8: Dataset with '?' values

kNN_project_19075570013.R* | dataset1

Filter

	age	workclass	fnlwgt	education	educational.num	marital.status	occupation	relationship	race
1	25	Private	226802	11th	7	Never-married	Machine-op-inspct	Own-child	Black
2	38	Private	89814	HS-grad	9	Married-civ-spouse	Farming-fishing	Husband	White
3	28	Local-gov	336951	Assoc-acdm	12	Married-civ-spouse	Protective-serv	Husband	White
4	44	Private	160323	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	Black
5	18	*NA*	103497	Some-college	10	Never-married	*NA*	Own-child	White
6	34	Private	198693	10th	6	Never-married	Other-service	Not-in-family	White
7	29	*NA*	227026	HS-grad	9	Never-married	*NA*	Unmarried	Black
8	63	Self-emp-not-inc	104626	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White
9	24	Private	369667	Some-college	10	Never-married	Other-service	Unmarried	White
10	55	Private	104996	7th-8th	4	Married-civ-spouse	Craft-repair	Husband	White
11	65	Private	184454	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White
12	36	Federal-gov	212465	Bachelors	13	Married-civ-spouse	Adm-clerical	Husband	White
13	26	Private	82091	HS-grad	9	Never-married	Adm-clerical	Not-in-family	White
14	58	*NA*	299831	HS-grad	9	Married-civ-spouse	*NA*	Husband	White

Showing 1 to 14 of 48,842 entries, 15 total columns

Fig. 9: Dataset1 with NA values

The replaced missing values in our data marked as '?' with 'NA' is shown in Fig. 9. Another way to check the number of NAs in our data column wise is given in Fig. 10.

```
> colSums(is.na(dataset1))
            age       workclass          fnlwgt       education educational.num  marital.status
              0            2799               0               0               0               0
     occupation    relationship            race          gender    capital.gain    capital.loss
           2809               0               0               0               0               0
 hours.per.week  native.country          income
              0             857               0
```

Fig. 10: Columns with NA values

Before applying the various classification algorithms:

It is observed that some variables are not self-explanatory because of the following reasons:

i. *capital_gain* and *capital_loss* is income from other sources like investments other than salary which have no relevance here.
ii. The continuous variable *fnlwgt* represents final weight, which is the number of units in the target population that the responding unit represents.
iii. The variable *education_num* stands for the number of years of education in total, which is a continuous representation of the discrete variable education.
iv. The variable *relationship* represents the responding members' role in the family.

For simplicity of this analysis, the following variables are removed *education.num*, *relationship*, *fnlwgt*, *capital.gain* and *capital.loss* as shown in Fig. 11.

dataset1 × dataset × kNN_project_19075570013.R* ×

Filter

	age	workclass	education	occupation	gender	hours.per.week	native.country	income
1	25	Private	11th	Machine-op-inspct	Male	40	United-States	<=50K
2	38	Private	HS-grad	Farming-fishing	Male	50	United-States	<=50K
3	28	Local-gov	Assoc-acdm	Protective-serv	Male	40	United-States	>50K
4	44	Private	Some-college	Machine-op-inspct	Male	40	United-States	>50K
5	18	NA	Some-college	NA	Female	30	United-States	<=50K
6	34	Private	10th	Other-service	Male	30	United-States	<=50K
7	29	NA	HS-grad	NA	Male	40	United-States	<=50K
8	63	Self-emp-not-inc	Prof-school	Prof-specialty	Male	32	United-States	>50K
9	24	Private	Some-college	Other-service	Female	40	United-States	<=50K
10	55	Private	7th-8th	Craft-repair	Male	10	United-States	<=50K
11	65	Private	HS-grad	Machine-op-inspct	Male	40	United-States	>50K
12	36	Federal-gov	Bachelors	Adm-clerical	Male	40	United-States	<=50K
13	26	Private	HS-grad	Adm-clerical	Female	39	United-States	<=50K
14	58	NA	HS-grad	NA	Male	35	United-States	<=50K

Showing 1 to 15 of 48,842 entries, 8 total columns

Fig. 11: Dataset with dropped columns

KNN imputation to replace NAs:

Here library(VIM) is required to impute missing values. KNN imputation is preferred over the conventional method of replacing with mean, median and mode as it is supposed to be more justified. It may occur that a person whose age is missing and earns >50k is allotted a median age which may not be true. KNN imputation will consider all the observations and based on the historical data will assign a better value.

```
C:/Users/Himani/OneDrive/Desktop/Sixth Sem/Data Mining/
> #We have taken k as 221 because sqrt of nrow(dataset1) comes as 221 approx.
> dataset2<-kNN(dataset1,variable = c("workclass","occupation","native.country"),k=221)
> # to verify if NAs removed
> colSums(is.na(dataset2))
                age           workclass           education          occupation              gender
                  0                   0                   0                   0                   0
     hours.per.week      native.country              income      workclass_imp     occupation_imp
                  0                   0                   0                   0                   0
native.country_imp
                  0
```

Fig. 12: Dataset with no missing values

More EDA:

Let's check income with respect to *age*.

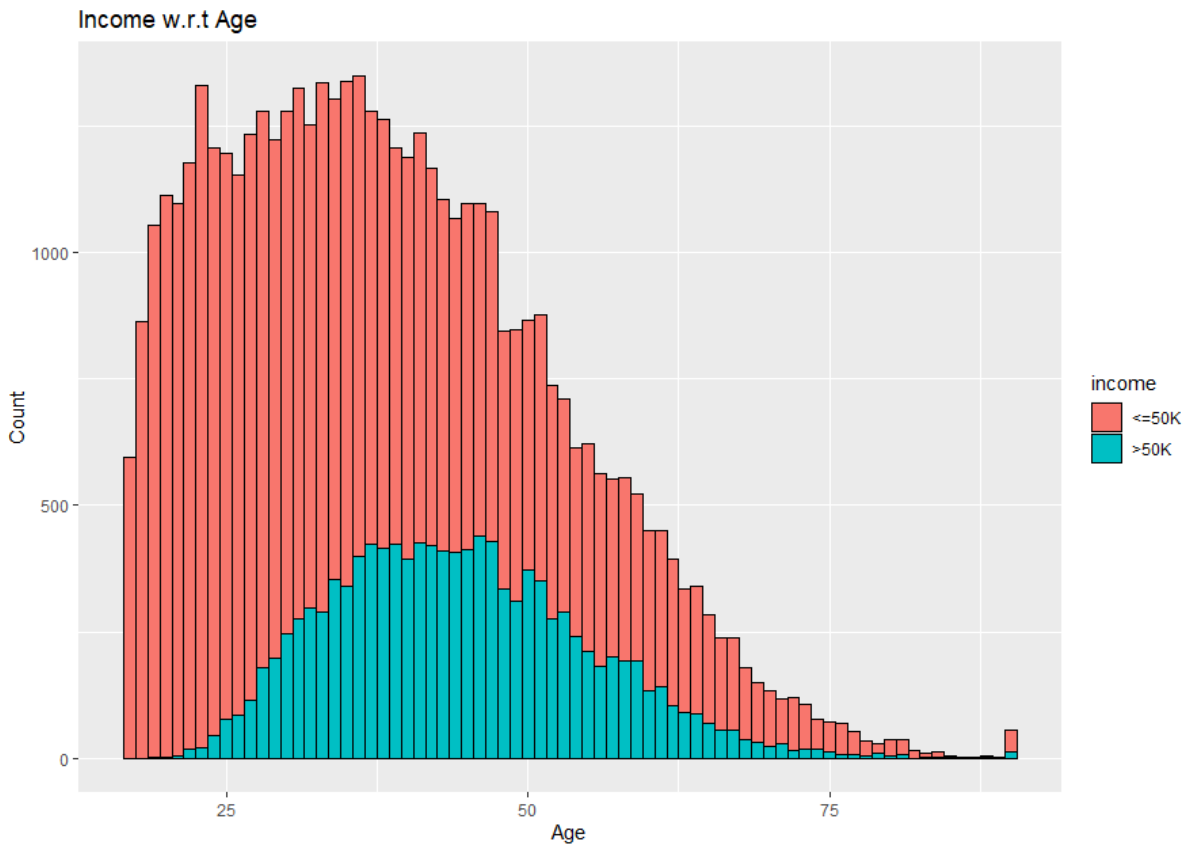

Fig. 13: Age VS Income

Majority of the people make less than <50k a year. However, observe people earning >50k are in their mid-career. Thus, we shall make this hypothesis based on the age.

Let’s check the same for *workclass*.

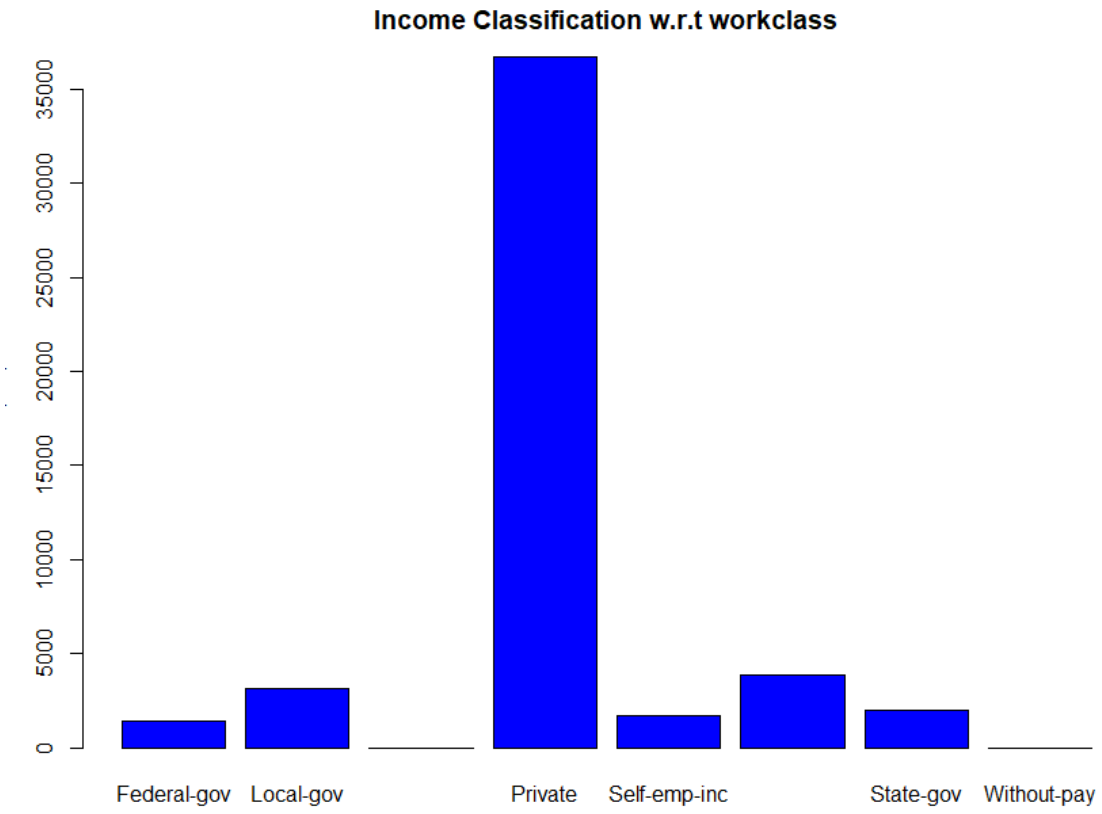

Fig. 14: Income VS Workclass

We can conclude that people working in private sector earn significantly better than the ones in other classes.

Dividing data in Training and Testing Datasets:

Let’s put 75% data in training and 25% in testing dataset.

Then we will train the model. [Appendix]

Load Library Receiver Operating Characteristic (ROC):

An ROC curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate. False Positive Rate.

i. Compare predicted values with actual values in training dataset.
ii. Store the measures with respect to which we want to plot the ROC graph.

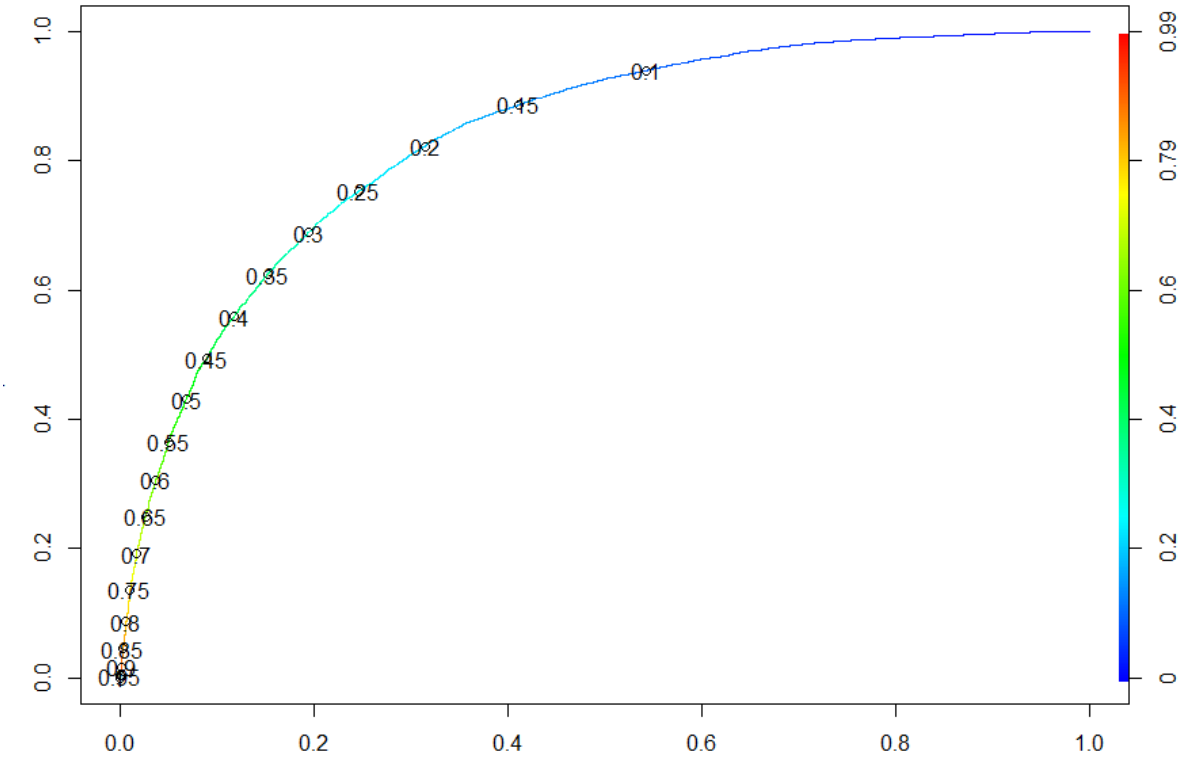

Fig. 15: ROC Curve

From the above observations and graphs, the resulted confusion matrix is shown below:

```
> table1<-table(train_adult$income,train_adult$pred_income)
> table1

            0     1
<=50K   22387  5401
>50K     2744  6101
```

Fig. 16: Confusion Matrix for Train dataset

```
> table2<-table(test_adult$income,test_adult$pred_income)
> table2

           0    1
<=50K   7539 1764
>50K     909 1997
```

Fig. 17: Confusion Matrix for Test dataset

Accuracy checking:

```
> accuracy_train<-100*(sum(diag(table1))/sum(table1));
> accuracy_train
[1] 77.83692
> accuracy_test<-100*(sum(diag(table2))/sum(table2));
> accuracy_test
[1] 78.10632
```

Fig. 18: Accuracy of train & test sample dataset

To check how much of our predicted values lie inside the curve:

```
> auc<-performance(pred,"auc")
> auc@y.values
[[1]]
[1] 0.8349227
```

Fig. 19: Accuracy of predicted values

We can analyse from the above observations and graphs that our model is good. We can conclude that we are getting an accuracy of 78.10% with 83.49% of our predicted values lying under the curve.

CONCLUSION

This study presents a proposed methodology to predict whether an individual earns more than USD 50,000 (50K) or less in a year using the 1994 US Census Data sourced from the UCI Machine Learning Repository (Lichman, 2013) [9]. The purpose of this work was to train and test the model with a large dataset. The performance measures such as accuracy, ROC, and confusion matrix were employed to confirm the validity of the method. The model was designed in layers with each layer solving one of the basic issues of supervised learning. The model was optimized for an extremely large data set in real-time. In that case, the optimization focuses on the reduction of execution time as well as further improvement of inaccuracy. Due to the large dataset, we concluded that KNN Classification Algorithm takes much more time than usual to perform and with slightly less accuracy.

REFERENCES

1. https://ieeexplore.ieee.org/document/6783471
2. https://www.geeksforgeeks.org/basic-concept-classification-data-mining/
3. https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d#:~:text=knn%20(K%20%E2%80%94%20Nearest%20Neighbors),(a%20vector)%20from%20other%20.
4. https://rpubs.com/Mr_President/income_prediction
5. https://www.hindawi.com/journals/tswj/2014/313164/
6. https://towardsdatascience.com/k-nearest-neighbors-knn-algorithm-23832490e3f4
7. Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Pearson Education.
8. http://theprofessionalspoint.blogspot.com/2019/02/advantages-and-disadvantages-of-knn.html
9. https://www.featureranking.com/tutorials/machine-learning-tutorials/case-study-predicting-income-status/
10. https://www.kaggle.com/datasets/wenruliu/adult-income-dataset

APPENDIX

```
install.packages("class")
library(class)
install.packages("caret")
library(caret)
#read dataset
dataset<-read.csv("adult.csv", header=T, stringsAsFactors=T)
# to check the first 6 observations in the data
head(dataset)
#EDA of the dependent variable
library(ggplot2)
barplot(table(dataset$income),main = 'Income Classification',col='blue',ylab ='No. of people')
#summary of the data
summary(dataset)
# This will replace '?' with 'NA'
dataset1<-read.csv("adult.csv",na.strings = c("?","NA"))
# Let's check summary again
summary(dataset1)
View(dataset1)
colSums(is.na(dataset1))
#-----RULES-----
#numerical rules
age>=17
age<=90
educational.num>=1
educational.num<=16
hours.per.week>=1
```

```
hours.per.week<99

#categorical rules

gender %in% c("Male","Female")

income %in% c("<=50K",">50K")

install.packages("editrules")

library(editrules)

rules<- editfile("knn_rules.txt")

print(rules)

ve<-violatedEdits(rules,dataset1)

ve

summary(ve)

par(mar=c(3,3,3,3))

plot(ve)

#dropping columns

drop <-
c("educational.num","marital.status","relationship","race","capital.loss","capital.gain","fnlwg
t")

dataset1 = dataset1[,!(names(dataset1) %in% drop)]

View(dataset1)

#removing NA values

install.packages("VIM")

library(VIM)

# as it is observed only the following columns have NAs in them, we specifically perform
kNN on these 3 variables

nrow(dataset1)

#We have taken k as 221 because sqrt of nrow(dataset1) comes as 221 approx.

dataset2<-kNN(dataset1,variable = c("workclass","occupation","native.country"),k=221)

# to verify if NAs removed
```

```
colSums(is.na(dataset2))

#now we create another data set excluding the dummy variables

dataset3<-dataset2[,1:10]

head(dataset3) # to verify if dummy variables removed

dim(dataset3) # gives the number of variables and columns in our dataset

#Let’s check income with respect to age

library(ggplot2)

ggplot(dataset3) + aes(x=as.numeric(age), group=income, fill=income) +

  geom_histogram(binwidth=1, color='black')+

  labs(x="Age",y="Count",title = "Income w.r.t Age")

#Let’s check the same for workclass

barplot(table(dataset3$workclass),main = 'Income Classification w.r.t
workclass',col='blue',ylab ='No. of people')

#Dividing data in Training and Testing Datasets

index<-createDataPartition(dataset3$age,p=0.75,list = F)

# argument 'list=F' is added so that it takes only indexes of the observations and not make a
list row wise

train_adult<-dataset3[index,]

test_adult<-dataset3[-index,]

dim(train_adult)

dim(test_adult)

# model implementation

adult_blr<-glm(as.factor(income)~.,data = train_adult,family = "binomial") # argument
(family = "binomial") is necessary as we are creating a model with dichotomous result

# To check how well our model is built we need to calculate predicted probabilities

train_adult$pred_prob_income<-fitted(adult_blr)# this column will have predicted
probabilities of being 1

head(train_adult) # run the command to check if the new column is added
```

```
# receiver operating characteristic

install.packages("ROCR")

library(ROCR)

# compares predicted values with actual values in training dataset

pred<-prediction(train_adult$pred_prob_income,train_adult$income)

# stores the measures with respect to which we want to plot the ROC graph

perf<-performance(pred,"tpr","fpr")

# plots the ROC curve

plot(perf,colorize=T,print.cutoffs.at=seq(0.1,by=0.05))

# we assign the threshold where sensitivity and specificity have almost similar values after observing the ROC graph

train_adult$pred_income<-ifelse(train_adult$pred_prob_income<0.3,0,1)

# this column will classify probabilities we calculated and classify them as 0 or 1 based on our threshold value (0.3) and store in this column

head(train_adult)

#Creating confusion matrix and assessing the results:

table1<-table(train_adult$income,train_adult$pred_income)

table1

dim(train_adult)

test_adult$pred_prob_income<-predict(adult_blr,test_adult,type = "response")

# an extra argument(type = "response") is required while using 'predict' function to generate response as probabilities

test_adult$pred_income<-ifelse(test_adult$pred_prob_income<0.3,0,1)

# we take the same threshold to classify which we considered while classifying probabilities of training data

head(test_adult)

dim(test_adult)

table2<-table(test_adult$income,test_adult$pred_income)
```

```
table2
#accuracy
accuracy_train<-sum(diag(table1))/sum(table1);
accuracy_train
accuracy_test<-sum(diag(table2))/sum(table2);
accuracy_test
auc<-performance(pred,"auc")
auc@y.values
```