A Project Report on

Prediction of Adult Income based on Census Data

Submitted in partial fulfillment for the completion of course

Data Mining Techniques(SWE2009)

in

M.Tech (SE)

By

Name

Debdatta Ray

Sruthi

Reg.No

20MIS0112

20MIS0269

Submitted to

Dr. B. PRABADEVI

Associate Professor

SCORE

Nov 2023

Table of Contents

Abstract

The prominent inequality of wealth and income is a huge concern especially in the United States. The likelihood of diminishing poverty is one valid reason to reduce the world's surging level of economic inequality. The principle of universal moral equality ensures sustainable development and improve the economic stability of a nation. Governments in different countries have been trying their best to address this problem and provide an optimal solution. This study aims to show the usage of machine learning and data mining techniques in providing a solution to the income equality problem. The UCI Adult Dataset has been used for the purpose. Classification has been done to predict whether a person's yearly income in US falls in the income category of either greater than 50K Dollars or less equal to 50K Dollars category based on a certain set of attributes. The Decision Tree Classifier Model was deployed which clocked the highest accuracy of 80.96%, eventually breaking the benchmark accuracy of existing works.

Acknowledgement

We would like to express our deepest gratitude to all those who helped and supported us during the course of this data mining project.

We extend our heartfelt appreciation to our teacher, Prabadevi B ma’am, for her invaluable guidance, continuous support, and constructive suggestions throughout this project. Her expertise and insights were instrumental in shaping the direction of our research.

We are also thankful to Vellore Institute of Technology for providing us with the necessary resources, data, and facilities, without which this project would not have been possible.

Our sincere thanks go to the participants who contributed their time and effort in providing the data used in this study. Their cooperation and willingness to share information were essential to the success of this research.

We are grateful to our families and friends for their understanding, encouragement, and unwavering support, which served as a constant source of motivation during the various stages of this project.

Lastly, we would like to express our appreciation to all the researchers, authors, and scholars whose insightful work laid the foundation for our research and provided us with a wealth of knowledge in the field of data mining.

Thank you all for your invaluable contributions to this project.

1. Introduction:

The project titled "Prediction of Adult Income based on Census Data" aims to utilize machine learning and data mining techniques to address the issue of income inequality. It leverages the UCI Adult Dataset, which contains census data, to predict whether an individual's income exceeds $50,000 per year. The project's primary objective is to develop a predictive model that can classify adults into two income groups: those earning more than $50,000 per year and those earning less. This classification is achieved by analyzing various attributes and features from the census data

The project is based on statistical and machine learning approaches, and it involves the use of classification algorithms to make predictions about an individual's income level based on available data. The dataset used in this project is sourced from the 1994 Census Bureau database and is intended to provide insights into income distribution.

1.1. Motivation

The motivation of a project focused on the prediction of adult income based on census data is to address the socioeconomic issue of income inequality and provide valuable insights into income distribution . Income inequality is a significant social and economic challenge. By predicting adult incomes, the project can shed light on the factors influencing this inequality, which can inform policy decisions and interventions to reduce disparities . Predictive models can assist individuals, organizations, and policymakers in making data-driven decisions. For instance, individuals can use the predictions to plan their finances, while organizations can use them for targeted marketing and government agencies for social programs.

1.2. Objective(s) of the proposed work

The primary objective of the project titled "Prediction of Adult Income based on Census Data" is to develop predictive models that can classify individuals into different income categories based on demographic and socio-economic features.

The project aims to predict whether an individual's income exceeds $50,000 per year or not. This classification is a fundamental goal, as it helps in understanding and addressing income disparities.

The project involves extensive data analysis of the UCI Adult Dataset, which includes various attributes such as age, education, occupation, and more. The objective is to identify which attributes are most influential in determining income levels.

1.3. Report Organization

2. Analysis and design of proposed work

2.1 Problem statement

The goal of this project is to predict if an individual's income exceeds 50K or not using machine learning classification algorithms and also finding patterns in the dataset using Association rules. This helps us to determine various things such as the lucrativeness of setting up a business in a city based on average income of the people, Real Estate preferences and bank loan eligibility for a particular person. In addition, we can also figure out what type of tourist places a particular strata of people would like to visit and whether that person's children would prefer a public or private college in future.

2.2 Stakeholder Identification

1. Tax officials- Overlook the tax paid by people. Also determine any tax evasion.
2. Working class citizens- Helps them to predict their financial growth and plan their investments accordingly.
3. Businessmen- Analyse areas of business establishment based on the finance structure of the residents .

2.3 Gaps Identified

[1] We can use data to make analysis and predictions to see how the trends change with time. We can make more data analysis by using some other machine learning techniques. We need ensemble techniques on the given problem to get better accuracy and is faster.

[2] An important application of this work is the prediction of regional and individual poverty levels in low HDI countries.As future work, we would like to investigate the performance implications of including temporal aspects of raw CDRs in our models and the data representation. In addition, we will work on finding a general representation of telecom data that can be used for various prediction tasks

[3] Machine learning algorithms can be employed to predict the individual's level of income. UsingRandom Forest (RT) classification, the features of age, capital gain, hours per week,relationship, education, occupation, marital status, work class, and capital loss can be used to nearly precisely predict income

[4] The last design gave Mean Squared Error = 0.18 for GBT Prediction. It was suggested to develop GBT algorithm to improve the MSE performance. The improvement can be using different set of rules (different kind of GT example without the use of GBT script but using random forest script) or combine boosted tree using ADAboost script.

[5] In this paper we proposed a salary prediction system by using a linear regression algorithm with second order polynomial transformation. For the proper salary prediction, we found out most relevant 5 features. The result of the system is calculated by suitable algorithm by

comparing it with another algorithms in terms of standard scores and curves like the classification accuracy, theF1score, the ROC curve, the Precision-Recallcurve etc.

[6] In this paper, a framework to predict salary spatially across economic activities and occupations is developed, presented, and demonstrated using training data from the Saudi labor market.Results suggests that MLR models do not provide the best fit; instead, the use of non-linear ML markedly improved the goodness of fit for the regression models. More specifically, the use of Bayesian ML by applying GPR performed with large and limited training data performed among the best. However, ANN also performed well when training data was limited. Use of statistical ML, can both reduces the cost of salary benchmarking and improves accuracy especially when estimating salary levels for similar occupations in different industries, or when estimating different occupations within the same sector. The use of MLR models did not produce accurate predictions.

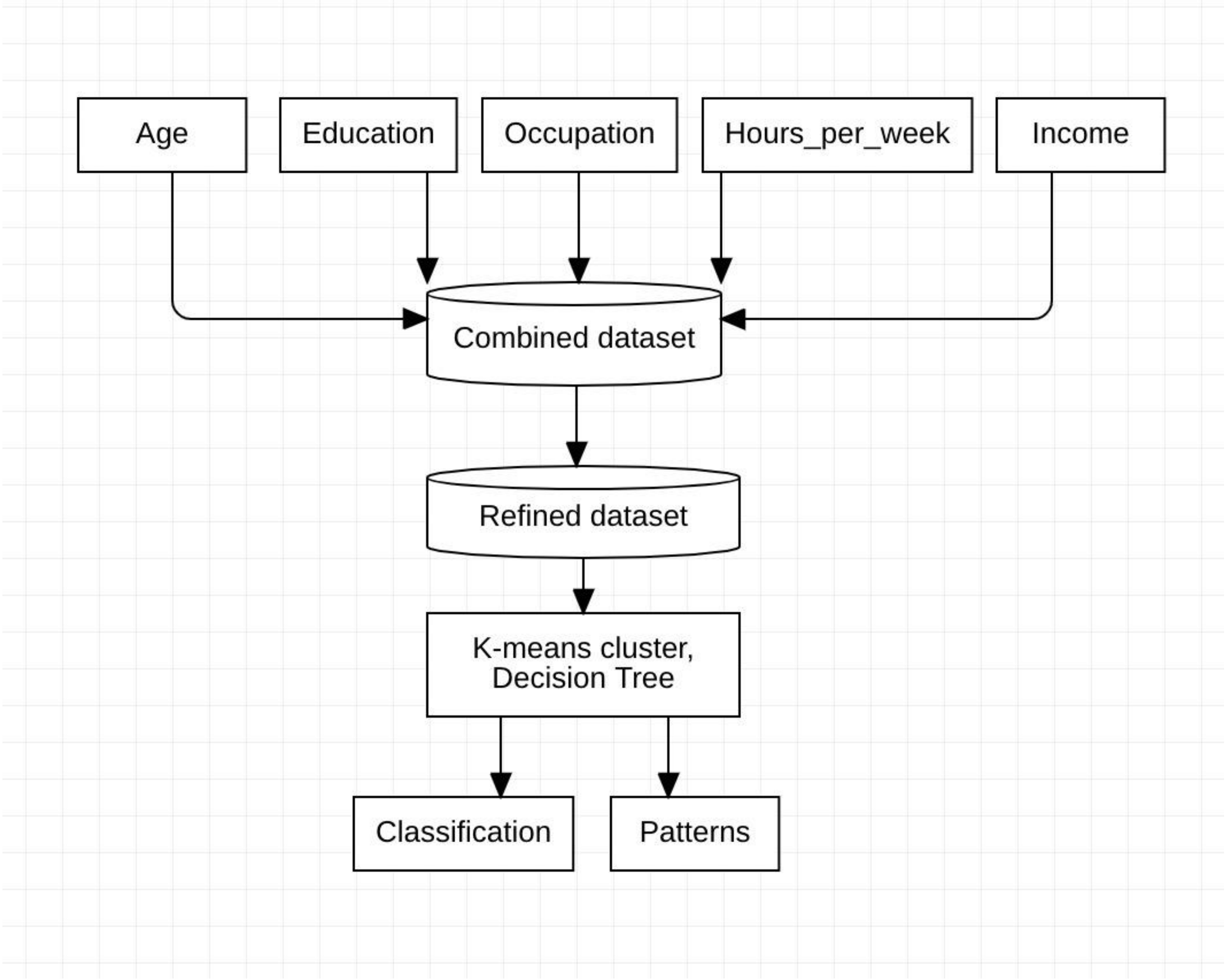
[7]This paper proposed the application of Ensemble Learning Algorithm, Gradient Boosting Classifier with extensive HyperParameter Tuning with Grid Search on Adult Census Data. Finally, the Validation Accuracy, so obtained, 88.16% which is, by the best of our knowledge, has been the highest ever numeric accuracy achieved by any Income Prediction Model so far.

[8]we measure the accuracy of various classification models directly by comparing the whole dataset. This new comparison gives the positive and improved result using the given metrics. The algorithm retrains itself each time an input variable is passed and compares itself to the previous scored label.

[9]We used the Adult dataset, which contains demographic and socio-economic features, and experimented with different classification algorithms, hyperparameter tuning, and feature engineeringThe results show that the Random Forest and ANN model has the highest probability of predicting the correct income level, while SVM and Logistic Regression have comparatively poor performance.

[10]The model was fitted with the optimal hyperparameters and the performance of the model was measured using the confusion matrix. From the results, we could observe that the model performs pretty well in predicting the income classes of individuals. An overall prediction accuracy of 85% was achieved by the model.

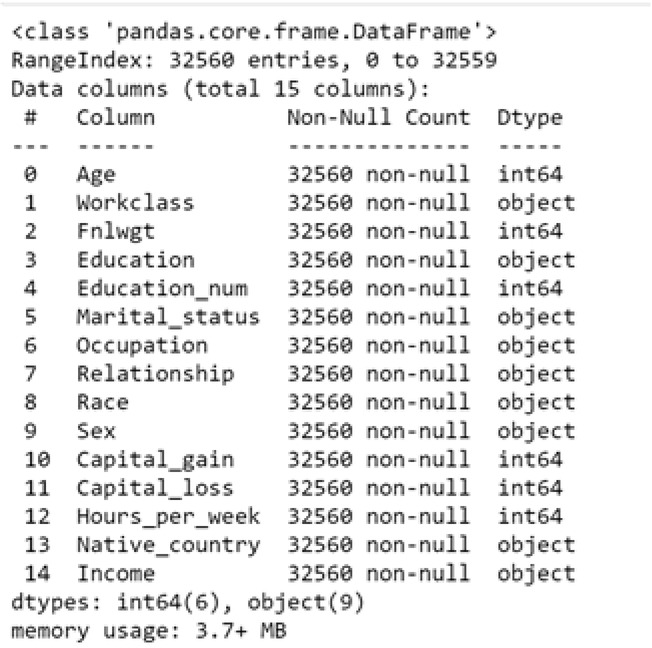
2.4 System Architecture



2.5 Module description

Exploratory Data Analysis

The first step that we do is to check the information about our data. We see the results shown in the image below:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32560 entries, 0 to 32559
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Age             32560 non-null  int64 
 1   Workclass       32560 non-null  object
 2   Fnlwgt          32560 non-null  int64 
 3   Education       32560 non-null  object
 4   Education_num   32560 non-null  int64 
 5   Marital_status  32560 non-null  object
 6   Occupation      32560 non-null  object
 7   Relationship    32560 non-null  object
 8   Race            32560 non-null  object
 9   Sex             32560 non-null  object
 10  Capital_gain    32560 non-null  int64 
 11  Capital_loss    32560 non-null  int64 
 12  Hours_per_week  32560 non-null  int64 
 13  Native_country  32560 non-null  object
 14  Income          32560 non-null  object
dtypes: int64(6), object(9)
memory usage: 3.7+ MB
```

We see that we have a mixture of categorical and numeric columns. We have 6 integer columns and 9 object type columns. We observe that the count of entries is 32560 for all columns, hence no NaN values are present in our dataset.

Data modeling

We now proceed to an important part of our process — data modeling. Based on our analysis above, we will fill the missing values in our data, and group certain categories logically, to allow our model to learn better.

Clustering

K-Means Clustering: K-Means clustering is an unsupervised learning algorithm that partitions n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is commonly used in image segmentation and market segmentation

Classification

Decision Tree:A decision tree is a tree-like model of decisions and their possible consequences. It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Decision trees are constructed using a top-down approach by recursively splitting the data into subsets based on the most significant attributes².

- K-Nearest Neighbor (KNN): KNN is a non-parametric algorithm that can be used for classification and regression. It works by finding the k-nearest neighbors to a given data point and then classifying the data point based on the majority class of its neighbors[1].

Outlier Analysis

We further proceed to detect outliers in our data and decide how to deal with them.

Team Contribution

Name	Contribution
Debdatta Ray	Analysis and design of proposed work, implementation, testing, comparison with existing system
Sruthi	Introduction, result and conclusion

3. Implementation

3.1 Software used with version

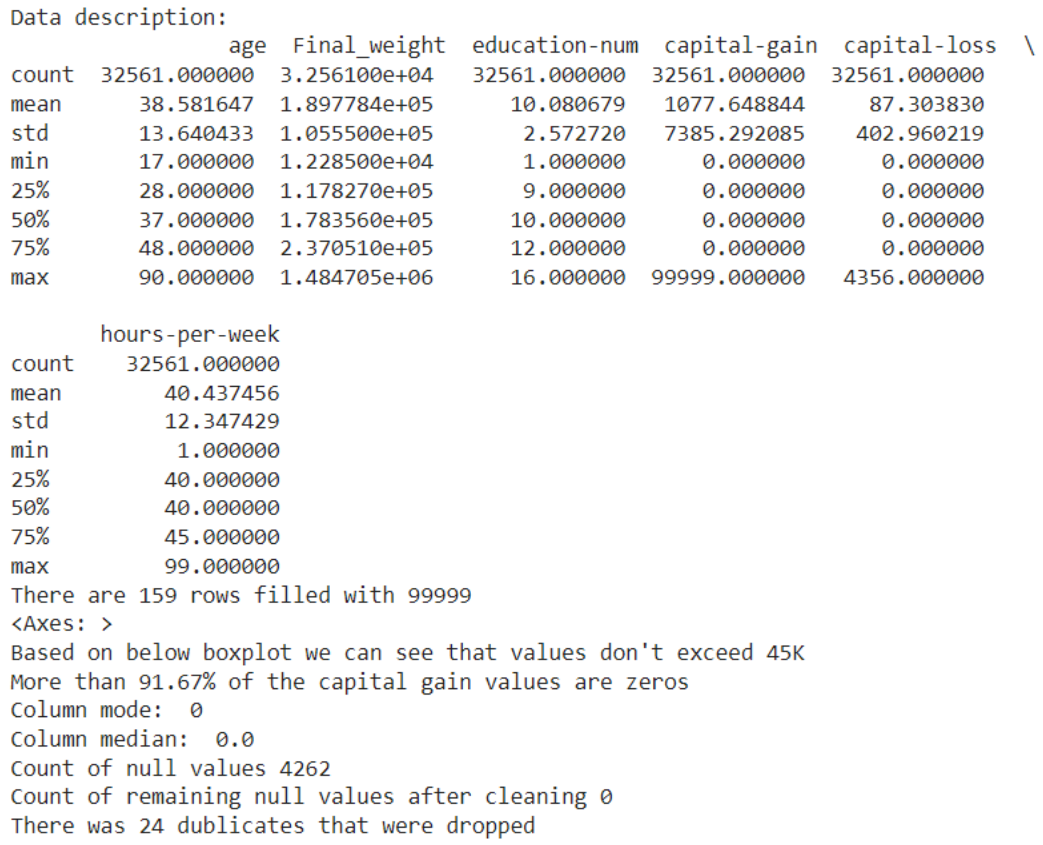
Operating system- Windows 11

IDE used - Colab

Runtime- 14.7 minutes

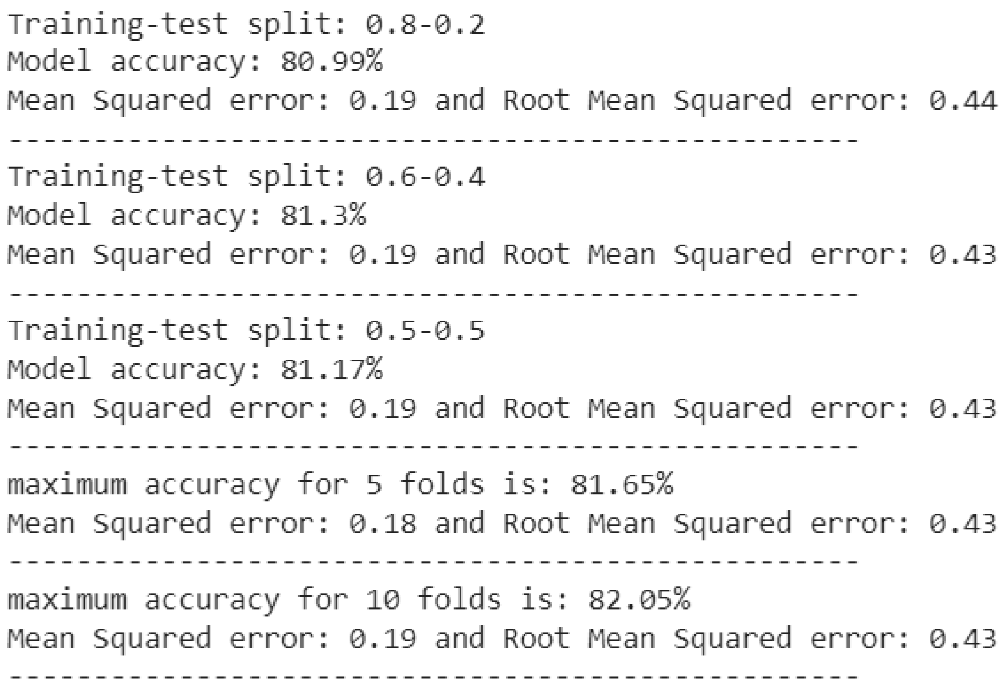
3.2 Screenshots

Data modelling

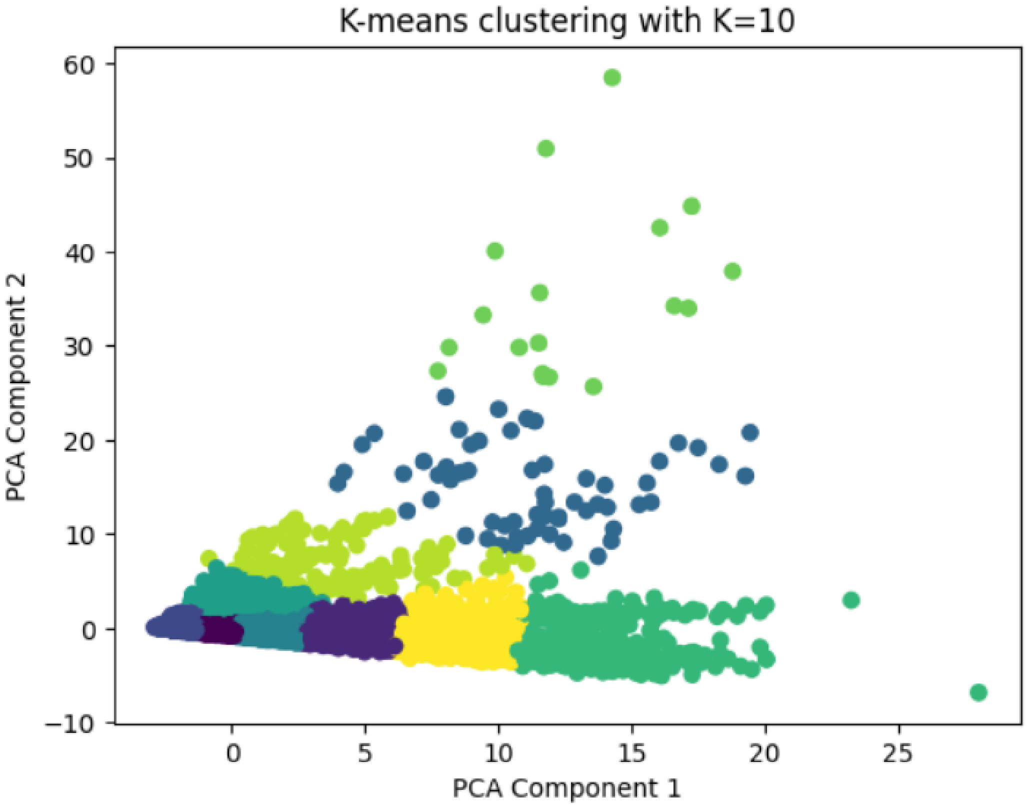


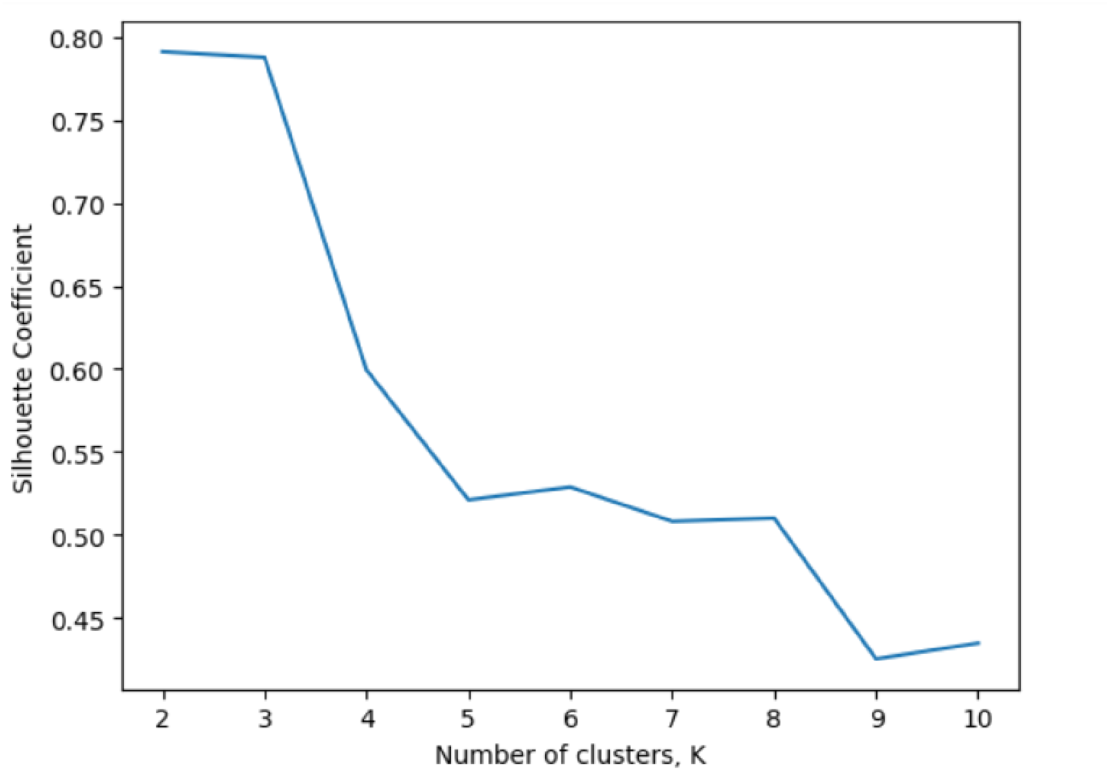
```
Data description:
                 age  Final_weight  education-num  capital-gain  capital-loss  \
count  32561.000000  3.256100e+04   32561.000000  32561.000000  32561.000000
mean      38.581647  1.897784e+05      10.080679   1077.648844     87.303830
std       13.640433  1.055500e+05       2.572720   7385.292085    402.960219
min       17.000000  1.228500e+04       1.000000      0.000000      0.000000
25%       28.000000  1.178270e+05       9.000000      0.000000      0.000000
50%       37.000000  1.783560e+05      10.000000      0.000000      0.000000
75%       48.000000  2.370510e+05      12.000000      0.000000      0.000000
max       90.000000  1.484705e+06      16.000000  99999.000000   4356.000000

       hours-per-week
count    32561.000000
mean        40.437456
std         12.347429
min          1.000000
25%         40.000000
50%         40.000000
75%         45.000000
max         99.000000
There are 159 rows filled with 99999
<Axes: >
Based on below boxplot we can see that values don't exceed 45K
More than 91.67% of the capital gain values are zeros
Column mode:  0
Column median:  0.0
Count of null values 4262
Count of remaining null values after cleaning 0
There was 24 dublicates that were dropped
```

Training and testing data split

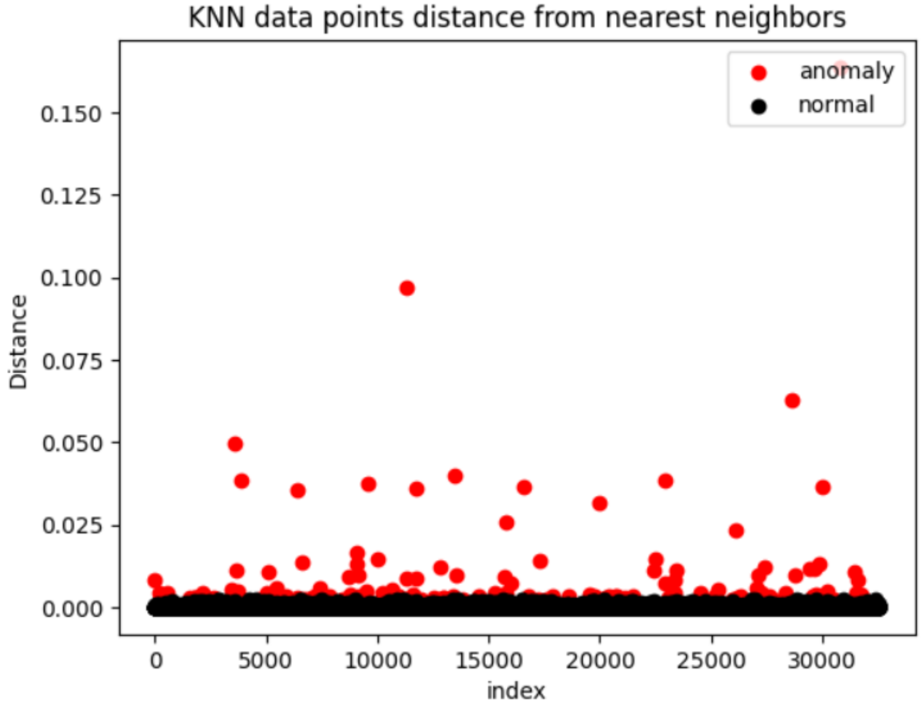


```
Training-test split: 0.8-0.2
Model accuracy: 80.99%
Mean Squared error: 0.19 and Root Mean Squared error: 0.44
--------------------------------------------------
Training-test split: 0.6-0.4
Model accuracy: 81.3%
Mean Squared error: 0.19 and Root Mean Squared error: 0.43
--------------------------------------------------
Training-test split: 0.5-0.5
Model accuracy: 81.17%
Mean Squared error: 0.19 and Root Mean Squared error: 0.43
--------------------------------------------------
maximum accuracy for 5 folds is: 81.65%
Mean Squared error: 0.18 and Root Mean Squared error: 0.43
--------------------------------------------------
maximum accuracy for 10 folds is: 82.05%
Mean Squared error: 0.19 and Root Mean Squared error: 0.43
--------------------------------------------------
```

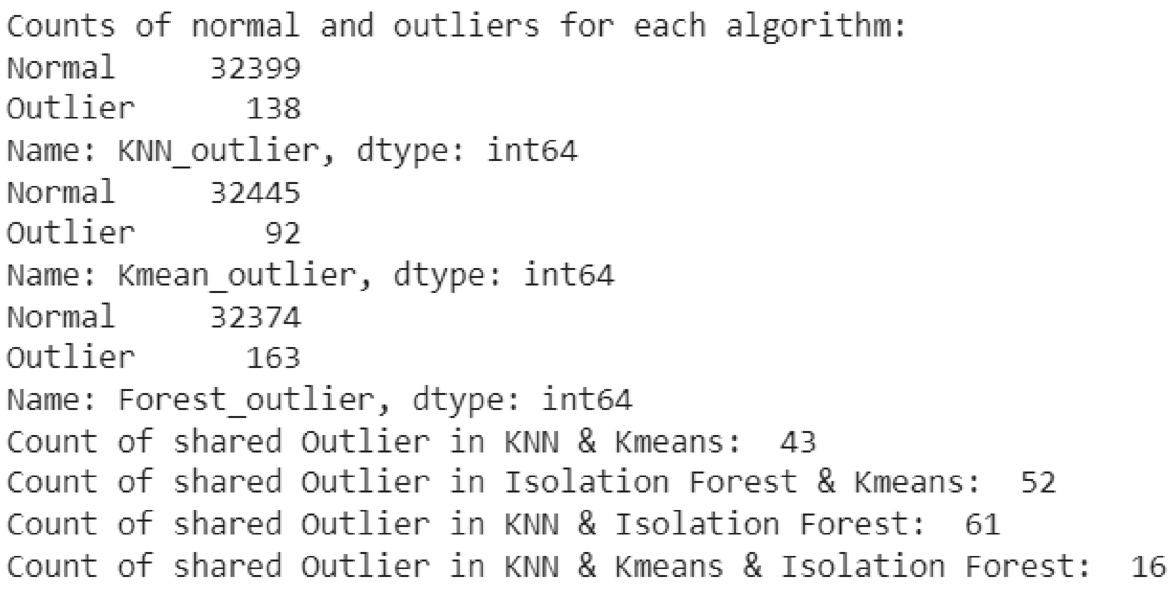




```
KNN autmatic threshold is: 0.00238 and any value further in distance will be labeled as anomaly
```



Outlier Analysis



```
Counts of normal and outliers for each algorithm:
Normal     32399
Outlier      138
Name: KNN_outlier, dtype: int64
Normal     32445
Outlier       92
Name: Kmean_outlier, dtype: int64
Normal     32374
Outlier      163
Name: Forest_outlier, dtype: int64
Count of shared Outlier in KNN & Kmeans:  43
Count of shared Outlier in Isolation Forest & Kmeans:  52
Count of shared Outlier in KNN & Isolation Forest:  61
Count of shared Outlier in KNN & Kmeans & Isolation Forest:  16
```

3.3 Source code

```
# Define a function to train and evaluate decision tree models

def create_classification_model(X,y): #function to create variables and split them

 # Define the training-test splits

 splits = [(0.8, 0.2), (0.6, 0.4), (0.5, 0.5)]

 # Train and evaluate models with diffrent splits

 for split in splits:

    print('Training-test split: {}-{}'.format(split[0], split[1]))

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=split[1], random_state=42)

    accuracy,mse,rmse,list1,list2 = DecisionTreeModel(X_train,X_test, y_train, y_test)

    evaluation(accuracy,mse,rmse)
 return(list1,list2)


# Functoin to build decision tree model
def DecisionTreeModel(x_train, x_test, y_train, y_test,K): ## creating a decision tree classifier
    # Train a decision tree classifier
    dtc = DecisionTreeClassifier()
    dtc.fit(x_train, y_train)
    # Evaluate the model on the test set
    y_pred = dtc.predict(x_test)
    accuracy = accuracy_score(y_test, y_pred)
    rounded_accuracy = round(accuracy*100,2)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    # Perform cross-validation and print mean accuracy and standard deviation
    cv_scores = cross_val_score(dtc, x_train, y_train, cv= K)
    #Storing accuracy and training data
    list1 = []
    list2 = []
    list1.append(accuracy)
    list2.append([x_train,y_train,y_test,y_pred])
```

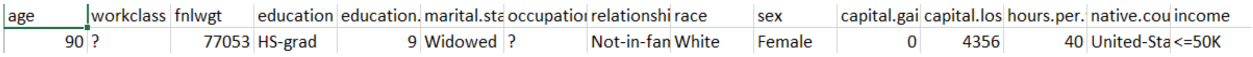
```
return(rounded_accuracy,round(mse,2),round(rmse,2),cv_scores.mean(),cv_scores.std(),list1,list2)


def DecisionTreeModel(x_train, x_test, y_train, y_test): ## creating a decision tree classifier
    # Train a decision tree classifier
    dtc = DecisionTreeClassifier()
    dtc.fit(x_train, y_train)
    # Evaluate the model on the test set
    y_pred = dtc.predict(x_test)
    accuracy = accuracy_score(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    #Storing accuracy and training data
    list1 = []
    list2 = []
    list1.append(accuracy)
    list2.append([x_train,y_train,y_test,y_pred])
    return(round(accuracy*100,2),round(mse,2),round(rmse,2),list1,list2)
 for train, test in kf.split(x): #split then find training accuracy
    x_train, x_test = x.iloc[train], x.iloc[test]
    y_train, y_test = y.iloc[train], y.iloc[test]
    accuracy,mse,rmse,list3,list4 = DecisionTreeModel(x_train, x_test, y_train, y_test)
    list1.append(train)
    list2.append(accuracy)
    n+=1
  print(f"maximum accuracy for {folds} folds is: {max(list2)}%")
  print(f"Mean Squared error: {mse} and Root Mean Squared error: {rmse}")
  print('-' * 50)
  return list1,list2,x_test,y_test


# returns the optimal k value determined by the silhouette coefficient
def get_optimal_k(X):
    # compute the Silhouette Coefficient for each K value
    k_values = range(2, 10)
    silhouette_scores = []
    for k in k_values:
        kmeans = KMeans(n_clusters=k, init='k-means++', max_iter=300, n_init=10, random_state=0)
        silhouette_scores.append(silhouette_score(X, kmeans.fit_predict(X)))


    # find the optimal K value
    optimal_k = k_values[np.argmax(silhouette_scores)]
    print('Optimal number of clusters:', optimal_k)
    return optimal_k



# Define a function to plot the clusters
def plot_clusters(df, labels, title):
    plt.scatter(df.loc[:, 0], df.loc[:, 1], c=labels)
    plt.xlabel('PCA Component 1')
    plt.ylabel('PCA Component 2')
    plt.title(title)
    plt.show()

# measure distance from data point to kmean centroid a
def distance_from_center(pred,income, label):
    cent_pred =  kmeans.cluster_centers_[label,0]
    cent_income =  kmeans.cluster_centers_[label,1]
    distance = np.sqrt((pred - cent_pred) ** 2 + (income - cent_income) ** 2)
    return np.round(distance, 3)
```

Link to complete code:

https://colab.research.google.com/drive/12vNFAPsHU_4apXXbj2Fbndggud0zf8z2?usp=sharing

4. Testing

Testcases



age	workclass	fnlwgt	education	education.	marital.sta	occupatio	relationshi	race	sex	capital.gai	capital.los	hours.per.	native.cou	income
90	?	77053	HS-grad	9	Widowed	?	Not-in-fan	White	Female	0	4356	40	United-Sta	<=50K

Accuracy- 81.96%

Training-test split - 60:40

Root mean square error- 0.43

5. Results

One aspect of interest in this dataset is its class imbalance, where there are significantly more individuals with income below $50,000 than those above it. Addressing class imbalance is a common challenge when working with this dataset The "Prediction of Adult Income based on Census Data" is a common machine learning task where the goal is to predict whether an individual's income exceeds $50,000 per year based on census data attributes. This problem is often used as a benchmark for machine learning and data mining techniques. Various datasets, including the "Adult Census Income" dataset, are available for this task on platforms like Kaggle

5.1 Comparison with existing systems

Ref no.	Methodology used	Technologies	System Description	Dataset	Performance analysis	limitations
1.	Naive Bayes	ML	Analysis of Income on the Basis of Occupation using Data Mining	Same dataset	Accuracy- 80.36% Error - 19.63% Correct classified- 7851	Takes longer time to generate results
2.	Deep neural network model	Deep learning	Deep Learning Applied to Mobile Phone Data for Individual Income Classification	Same dataset	AUC on training data- 80%	classic machine learning approaches have the advantage of being

					AUC on test data-77%	interpretable which automated approach lacks.
3.	Random Forest (RF), K Nearest Neighbor (KNN), Support Vector Machines (SVM), Logistic Regression and Naïve Bayes algorithm	Machine learning	Research on Income Forecasting based on Machine Learning Methods and the Importance of Features	Same data set	ACC: 0.97977 SVM ACC: 0.80524 NB ACC: 0.79717 class precision	More research in different parament of RT algorithm.
4.	GLM, Decision tree, random forest	Machine learning	Prediction of B2B sales using ML.	Sales data of 2016,2017,2018	Mape-GLM:0.28 Decision tree:0.7 Random forest:0.3	Need to improve mse performance.
5.	Linear regression	Machine learning	Salary prediction using machine learning.	Same dataset	Accuracy-76% Mse- 357	Considers only two attributes as relevant.
6.	Bayesian model	Machine learning		Salary data	Goodness of fit-0.12	Low goodness of fit might indicate need of larger training sets.
7.	Feature Engineering and Selection Kernel Density Estimation Exploratory Data Analysis	Machine learning algorithms	Adult Income Prediction Using various ML Algorithms	Same dataset	Naive Bayes train-81.9% test-82%	Machine Learning and Deep Learning (Neural Networks) - to produce better results overall while maintaining accuracy.
8	Neural network , Support vector machines, Poisson Regression, Boosted Decision Tree Regression	Machine learning algorithm	Comparative Analysis of Classification Models on Income Prediction	Same data set	Accuracy: 0.90	negative impact, enhance the model by carefully training it

9.	Decision tree, random forest, SVM	Machine learning algorithm	Predicting Annual Income of Individuals using Classification Techniques	Same data set	Accuracy : 85.36%	Our findings show the importance of considering imbalanced data and using appropriate metrics, such as the F1-score, when evaluating classification models.
10.	Decision tree analysis	Machine learning algorithm	Classification of Adult Income Using Decision Tree*	Same data set	accuracy is 0.85290	Features with higher importance scores tend to contribute more to improving

6. Conclusion

Predicting adult income based on census data is a well-established machine learning task. Researchers use datasets like the "Adult Census Income" dataset to predict whether an individual's income exceeds $50,000 per year based on attributes. Various studies and projects have tackled this problem using machine learning and data mining techniques. Notably, class imbalance in the dataset, where there are more individuals with incomes below $50,000, is a common challenge. The UCI Adult Dataset has been used for classification to determine income levels, and researchers have explored various machine learning models for this task. The task is to classify whether a given adult makes more than $50,000 a year based on attributes

7. References

[1]*Analysis of Income on the Basis of Occupation using Data Mining*. (2022, February 16). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/9759040

[2] Sundsøy, P., Bjelland, J., Reme, B., Iqbal, A., & Jahani, E. (2016). Deep Learning Applied to Mobile Phone Data for Individual Income Classification. *Deep Learning Applied to Mobile Phone Data for Individual Income Classification*. https://doi.org/10.2991/icaita-16.2016.24

[3]Wang, J. (2022). Research on Income Forecasting based on Machine Learning Methods and the Importance of Features. *Proceedings of the International Conference on Information Economy, Data Modeling and Cloud Computing, ICIDC 2022, 17-19 June 2022, Qingdao, China*. https://doi.org/10.4108/eai.17-6-2022.2322745

[4]Wisesa, O., Andriansyah, A., & Khalaf, O. I. (2020). Prediction Analysis for Business To Business (B2B) Sales of Telecommunication Services using Machine Learning Techniques. *Majlesi Journal of Electrical Engineering*, *14*(4), 145–153. https://doi.org/10.29252/mjee.14.4.145

[5]http://ijasret.com/VolumeArticles/FullTextPDF/842_47._SALARY_PREDICTION_USING_MACHINE_LEARNING.pdf

[6]Matbouli, Y. T., & Al-Ghamdi, S. (2022). Statistical machine learning regression models for salary prediction featuring economy wide activities and occupations. *Information*, *13*(10), 495. https://doi.org/10.3390/info13100495

[7]Thapa, S. (2023). Adult Income Prediction Using various ML Algorithms. *Social Science Research Network*. https://doi.org/10.2139/ssrn.4325813

[8]Kakulapati, V. (2017). Comparative analysis of classification models on income prediction. *ResearchGate*. https://www.researchgate.net/publication/316968182_Comparative_Analysis_of_Classification_Models_on_Income_Prediction

[9]Shuvo, S. S., Mohanty, J., & Patel, D. (2023). Predicting Annual Income of Individuals using Classification Techniques. *ResearchGate*. https://doi.org/10.13140/RG.2.2.33330.99529

[10]*Salary Classification & Prediction based on Job Field and Location using Ensemble Methods*. (2023, February 16). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/10127828