## INTRODUCTION

Support Vector Machines, Decision Tree Classifiers and Random Forests are supervised machine learning algorithms that are widely used in classification problems. The dataset that has been used to implement the above algorithms is the Census Income Dataset that contains 14 columns that provides information regarding an individual about their country, race, sex, education, marital status etc. and can act as features and has an income column that can be considered as the target. The job of the algorithms is to try and accurately predict whether a person belongs to the income class which is greater than 50K or the income class which is less than 50K based on the 14 features available in the dataset. The 14 features include both numerical and categorical columns that has both nominal and ordinal data.

| SVM | Decision Tree Classifiers | Random Forest Classifier |
|---|---|---|
| This is a binary classification algorithm that uses kernel functions (polynomial or rbf) to systematically find support vector classifiers in higher dimensions. The advantages include increased class separation and reduced expected prediction error. | This algorithm can be used for both classification and regression problems and they have a root node and decision nodes. The partitioning of the tree is done in a recursive manner. The advantages include the speed of execution and the simplicity. | This algorithm can be used for both classification and regression problems and is an ensemble method and each of the smaller model in a random forest is a decision tree. The advantages include its versatility and efficient default hyperparameters. |

*Table 1: Explanation of 3 ML Models*

### Data Preprocessing :

The data preprocessing techniques that were used include converting the column datatypes from object to string so that the whitespaces in every column can be removed. Null values in the dataset which were represented as '?' were replaced with Not A Number (np.nan) and later replaced by the mode value of each column instead of removing them from the datasets. Replacing the null values with the mode value for each column showed a greater accuracy while training compared to just removing the null values from the dataset.

### Exploratory Data Analysis :

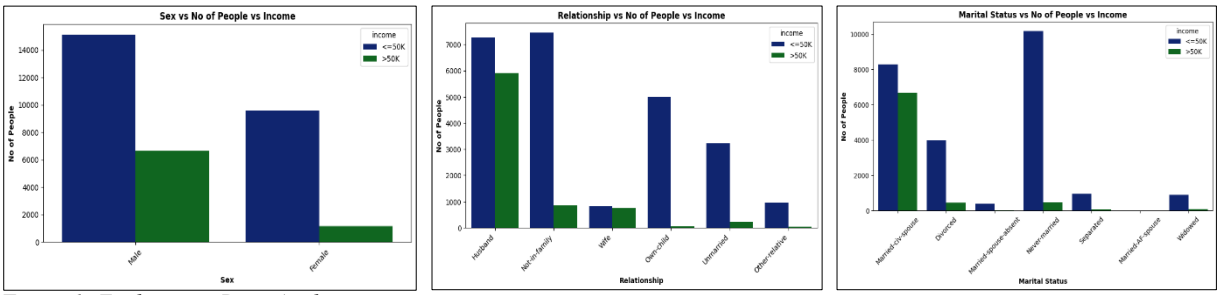


*Figure 1: Exploratory Data Analysis*

Exploratory Data Analysis were carried out on different columns with respect to the income class and various interesting patterns, observations and useful information were extracted.

### Feature Engineering :

| COLUMN NAMES | COLUMN TYPE | METHOD USED |
|---|---|---|
| Native_country, Workclass, Marital_status, occupation, relationship, race | Categorical/Nominal | Count/Frequency Encoding |
| Education | Categorical/Ordinal | Label Encoding |
| Sex | Categorical/Nominal | One Hot Encoding |

*Table 2: Feature Engineering Methods*

Dataset link : https://archive.ics.uci.edu/ml/datasets/Census+Income
Colab link : https://colab.research.google.com/drive/1vGW0PQMF6HxxGQT4Eo5GF74Kyun1UdM4#scrollTo=M458AfaP08Ac

Different techniques such as count/frequency encoding, label encoding, one hot encoding with multiple categories and mean encoding were carried out on the columns based on the column type and the best techniques were selected based on accuracy score using trial and error method.

**Feature Scaling :**

Three techniques of feature scaling which are normalization (MinMaxScaler), standardization (StandardScaler) and no scaling were implemented to see how each of the technique effect the performance of the model and the standardization technique using Standard Scaler turned out to improve the performance of the ML models and was implemented in all the columns.

**Feature Selection Methods :**

The techniques that were implemented are correlation coefficient method, recursive feature elimination method and chi square method and each method provided us with a list of columns that could be considered as potential features. Since the chi square method of feature selection works best for categorical features and categorical target columns, the features were selected based on the selection from chi square method, this was also verified by selecting the columns as features from correlation method and RFE method. Columns from chi square gave higher accuracy in all the three algorithms compared to the other 2 feature selection methods.

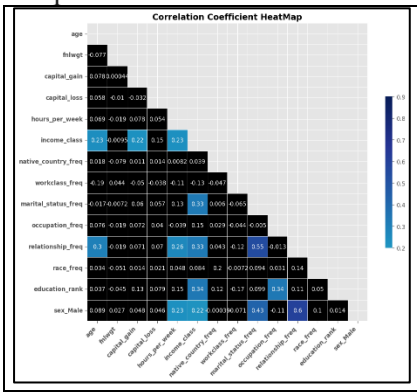


*Fig 2. Correlation Coefficient Method*


*Fig3. RFE Method*

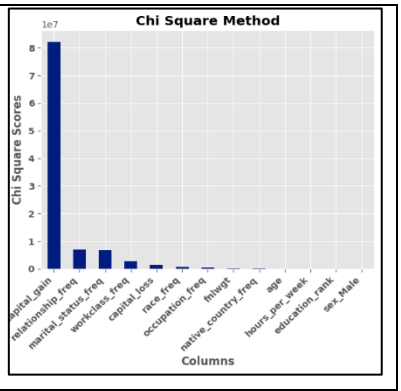


*Fig4. Chi Square Method*

Since the data preprocessing, feature engineering, feature scaling and feature selection techniques were carried out, machine learning models are now ready to be trained. In order to train the model, the data was split into 75 % of training and 25% of testing after various combinations of train and test split using trial and error method to check the better performance based on the split. Also, all the necessary libraries were imported to carry out the training and to evaluate the performance of the models.

**Performance of the three models :**

| Algorithm | Accuracy | Precision | F1 Score | Recall | ROC AUC | Log Loss |
|---|---|---|---|---|---|---|
| SVM | 0.804 | 0.753 | 0.396 | 0.269 | 0.620 | 7.05 |
| Decision Tree | 0.8460 | 0.743 | 0.628 | 0.544 | 0.742 | 5.54 |
| Random Forest | 0.8460 | 0.744 | 0.629 | 0.542 | 0.741 | 5.55 |

*Table 3: Performance Comparison of 3 models*

**Conclusion**

The study shows the performance of the three models on the income dataset. It can be observed that all the three algorithms performed well. Decision Tree and Random Forest algorithms came out with a higher accuracy score of 84% each when compared to the accuracy score of SVM which is only 80%. Precision scores, F1 scores and Recall Scores of Random Forest and Decision Tree algorithms were similar values and higher than the SVM performance metrics. Also, when trying to run the algorithms separately, execution time taken by SVM was greater than the execution time taken by the other two algorithms which proves that Random Forest and Decision Tree algorithms are faster than SVM in terms of execution time required. This shows that the income census dataset performs really well on tree models

Dataset link : https://archive.ics.uci.edu/ml/datasets/Census+Income
Colab link : https://colab.research.google.com/drive/1vGW0PQMF6HxxGQT4Eo5GF74Kyun1UdM4#scrollTo=M458AfaP08Ac