Performance Analysis of Machine Learning Models for Income Prediction on UCI Adult dataset

Ankush Jauhari
Department of Information Technology
ABES Engineering College, Ghaziabad, UP, India
ankushjauhari013@gmail.com

Abstract:
Money is not everything, but money is something very important**.** In this paper, we analyse the classic US Adult Income Dataset Our aim is to predict whether the income of U.S. population exceeds $50K/year or not based on census data provided by Census bureau database, while considering different factors such as age, work class, education, marital status, country, occupation etc. We also aim to measure the accuracy of different models using Logistic Regression, Decision Tree Classifier, Svc etc. For the final output, the result of all the models will be considered.

Keywords- machine learning, income prediction, adult dataset, performance analysis, variables, dataset, Prediction, Classification, Classification report.

1. Introduction

Carrie Wilkerson once said "The longer you're not taking action the more money you're losing." Money is the most important thing in life, without money life does not last. Everything in an adult's life with financial responsibilities has to do with income (money). It is a means of providing comfort and safety for their children and their families. So money represents success in many aspects.
Our dataset contains 48243 records with various attributes such as age, relationship, occupation, education, income, country and so on.
In our first section, we explore the data in order to understand the trends. In this set all the attributes are not relevant for our analysis. The selection of the useful attributes will be based on the outcomes of the various algorithms.

2. Literature Review

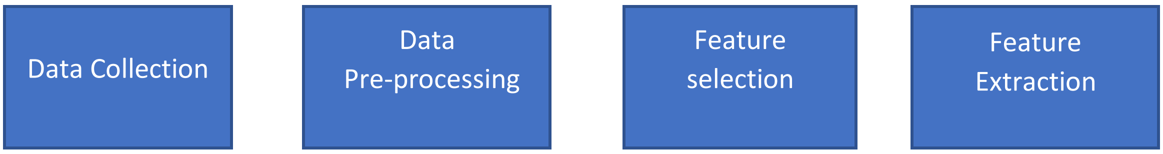
Certain efforts using machine learning models have been made in the past by researchers for predicting income levels.

•Chockalingam et. al. [1] explored and analysed the Adult Dataset and used several Machine Learning Models likeLogistic Regression, Stepwise Logistic Regression, NaiveBayes, Decision Trees, Extra Trees, k-Nearest Neighbor,SVM, Gradient Boosting and 6 configurations of Acti-vated Neural Network. They also drew a comparativeanalysis of their predictive performances.

•Bekena [2] implemented the Random Forest Classifieralgorithm to predict income levels of individuals

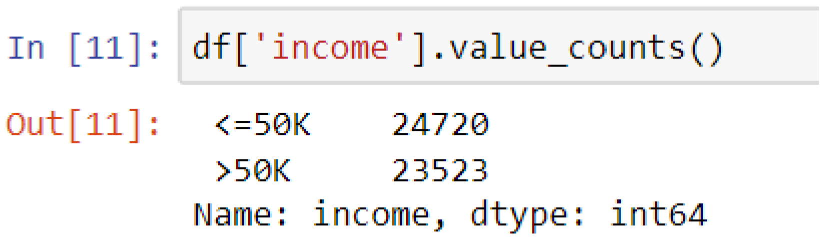
•Topiwalla [3] made the usage of complex algorithms likeXGBOOST, Random Forest and stacking of models forprediction tasks including Logistic Stack on XGBOOST and SVM Stack on Logistic for scaling up the accuracy.

3. Proposed methodology



- **Data Collection**

For this project the data has been downloaded from UCI's Machine Learning repository (http://archive.ics.uci.edu/ml/datasets/Adult). It consist of 12 classes namely age, work class, education, marital status, country, occupation etc. There are total of 48243 number of samples and divided in 2 classes as shown in figure 1.



```
In [11]: df['income'].value_counts()

Out[11]:  <=50K    24720
          >50K     23523
         Name: income, dtype: int64
```

Figure 1

- **Data Pre-processing**

The data is not pre-processed it require many changes in dataset. There are 12 missing values for country and 8 columns having categorical data as shown in fig 2.

```
age               0
workclass         0
education         0
marital-status    0
occupation        0
race              0
sex               0
capital-gain      0
capital-loss      0
hours/week        0
country          12
income            0
dtype: int64
```

```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             48243 non-null  int64
 1   workclass       48243 non-null  object
 2   education       48243 non-null  object
 3   marital-status  48243 non-null  object
 4   occupation      48243 non-null  object
 5   race            48243 non-null  object
 6   sex             48243 non-null  object
 7   capital-gain    48243 non-null  int64
 8   capital-loss    48243 non-null  int64
 9   hours/week      48243 non-null  int64
 10  country         48231 non-null  object
 11  income          48243 non-null  object
dtypes: int64(4), object(8)
```

Figure 2

```
In [10]: def data_cleanup(df):
             '''
             df: pandas dataframe
             '''
             if type(df)!=pd.core.frame.DataFrame:
                 raise ValueError('input is not a pandas dataframe')
             working_df = df.copy()
             cols = working_df.columns
             converted_columns = {}
             for col in cols:
                 if working_df[col].dtype == 'O':
                     unique_values = working_df[col].unique()
                     converted_values = {v:k for k,v in enumerate(unique_values)}
                     for value in unique_values:
                         working_df[col] = working_df[col].replace(value, converted_values[value])
                     converted_columns[col] = converted_values
             return working_df, converted_columns
         clean_df,coverted=data_cleanup(df)
         clean_df.head()
         df=clean_df
         df
```

Out[10]:

	age	workclass	education	marital-status	occupation	race	sex	capital-gain	capital-loss	hours/week	country	income
0	52	0	0	0	0	0	0	0	0	45	0	0
1	31	1	1	1	1	0	1	14084	0	50	0	0
2	42	1	2	0	0	0	0	5178	0	40	0	0
3	37	1	3	0	0	1	0	0	0	80	0	0
4	30	2	2	0	1	2	0	0	0	40	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
48238	32	1	1	1	2	2	0	0	0	11	8	1
48239	22	1	3	1	6	0	0	0	0	40	0	1
48240	27	1	7	0	2	0	1	0	0	38	0	1
48241	58	1	0	4	3	0	1	0	0	40	0	1
48242	22	1	0	1	3	0	0	0	0	20	0	1

45178 rows × 12 columns

```python
print (df.shape)
df['country'] = df['country'].replace(' ?',np.nan)
df['workclass'] = df['workclass'].replace(' ?',np.nan)
df['occupation'] = df['occupation'].replace(' ?',np.nan)
df.dropna(how='any',inplace=True)
print (df.shape)
print (df.head(10))
```

```
(48243, 12)
(45178, 12)
```

```
    age          workclass     education      marital-status  \
0    52   Self-emp-not-inc       HS-grad  Married-civ-spouse
1    31            Private       Masters       Never-married
2    42            Private     Bachelors  Married-civ-spouse
3    37            Private  Some-college  Married-civ-spouse
4    30          State-gov     Bachelors  Married-civ-spouse
6    43   Self-emp-not-inc       Masters            Divorced
7    40            Private     Doctorate  Married-civ-spouse
8    56          Local-gov     Bachelors  Married-civ-spouse
11   57        Federal-gov     Bachelors  Married-civ-spouse
12   47            Private   Prof-school  Married-civ-spouse
```

- **Feature selection**

For feature selection we take all the parameters with mean values and draw a heat-map between these parameters of the dataset.

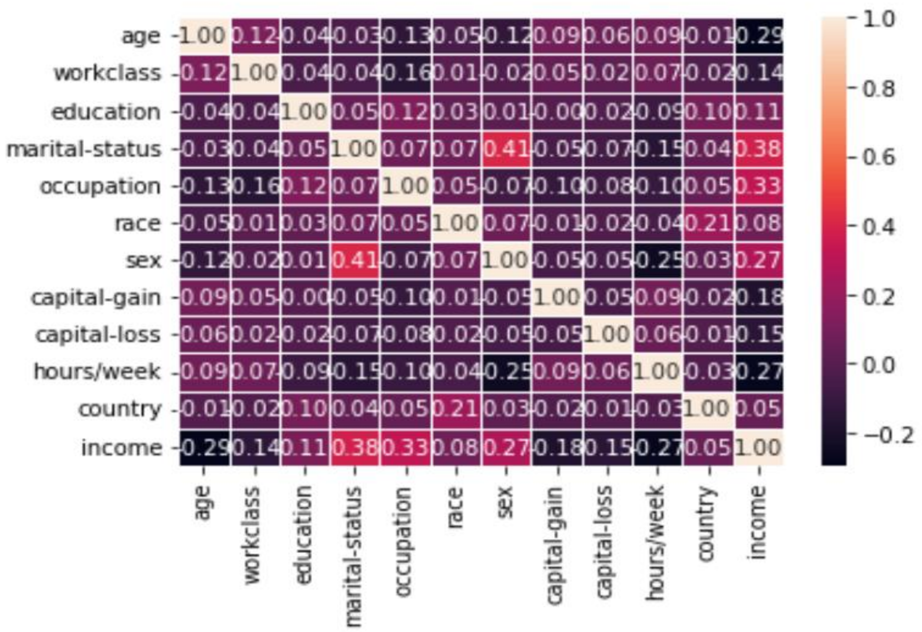
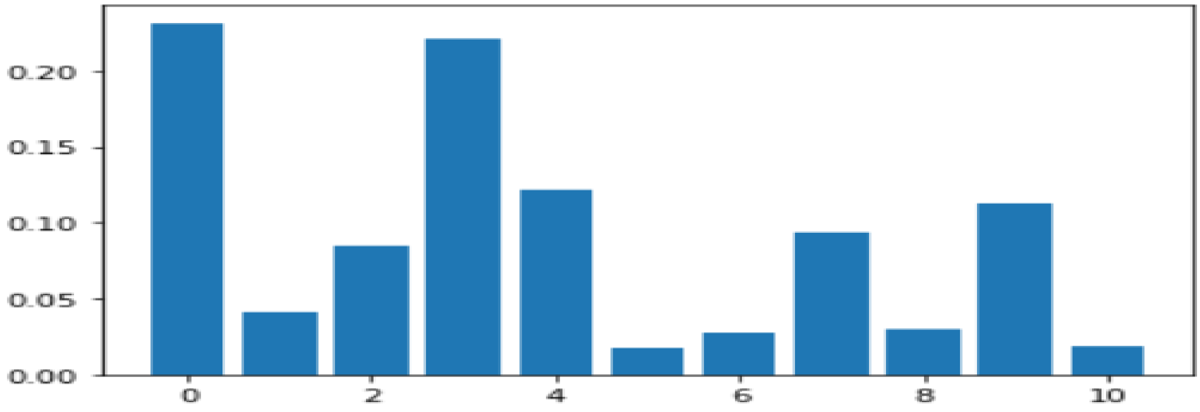


Figure 3

- **Feature Extraction**

PCA (Principle Component Analysis) is used to extract features by reducing the numbers of input variables, we use Random forest with PCA and got 90.94% accuracy score, Feature importance is also used to find importance of every parameter as shown in figure 4.

Figure 4



4. Result & Discussion

All experiment has been carried out in widows 10 with Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz, 2.00 GHz, RAM 8 GB. In this analysis we worked on twelve Machine Learning Models providing accuracy 73% by Perceptron, 80% by Logistic Regression, 82% by KNN, 68.8% by Gaussian NB, 81.8% by SVC, 90.8% by Random Forest, 89% by PCA, 89% by LDA, 90% by Decision Tree, 82.95% by Ada Boost and highest 91.5% by Extra Tree.

Table 1. Shows the result comparision of some ML techniques

Model	Accuracy	Recall	Precision
SVC	0.8182	0.90	0.84
Naïve Bayes	0.6881	0.90	0.89
PCA	0.8902	0.96	0.87
LDA	0.8906	0.96	0.95
Random Forest	0.9094	0.95	0.96
Ada boost	0.8295	0.84	0.82
Extra tree	0.9154	0.96	0.96

5. **Conclusion** : In this study we illustrate twelve machine learning models in which Extra Tree classifier provides the best accuracy score of 91.54% followed by Random forest classifier.

```
In [73]: from sklearn.ensemble import ExtraTreesClassifier
         cl=ExtraTreesClassifier()
         cl.fit(x_train,y_train)
         y_pred=cl.predict(x_test)
         from sklearn.metrics import accuracy_score
         print((accuracy_score(y_test,y_pred)))

         0.9154493138556884

In [74]: from sklearn.metrics import classification_report
         print(classification_report(y_pred,y_test))

                       precision    recall  f1-score   support

                    0       0.96      0.88      0.92      7372
                    1       0.87      0.96      0.91      6182

             accuracy                           0.92     13554
            macro avg       0.92      0.92      0.92     13554
         weighted avg       0.92      0.92      0.92     13554
```

References:

[1] Vidya Chockalingam, Sejal Shah and Ronit Shaw: “Income Classificationusing Adult Census Data”, https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a120.pdf.

[2] Sisay Menji Bekena:“Using decision tree classifier to predict incomelevels”, Munich Personal RePEc Archive 30th July, 2017

[3] Mohammed Topiwalla: “Machine Learning on UCI Adult data Set UsingVarious Classifier Algorithms And Scaling Up The Accuracy UsingExtreme Gradient Boosting”, University of SP Jain School of GlobalManagement

[4]. Besse P, del Barrio E, Gordaliza P, Loubes JM, Risser L. A survey of bias in machine learning through the prism of statistical parity.The American Statistician. 2021 Aug 24:1-1.

[5]. Chet Lemon (A10895241)
Chris Zelazo (A10863450)
Kesav Mulakaluri (A10616114

[6]. Bricker, J., Ramcharan, R. and Krimmel, J. (2014), Signaling status: the impact of relative income on household consumption and financial decisions, Federal Reserve Board, Finance and Economics Discussion Series (FEDS) Working Paper no. 2014-76.

[7]. Kennickell, Arthur B. "Using income data to predict wealth." Federal Reserve Board. Retrieved on April 22 (1999): 2005.

[8]. https://archive.ics.uci.edu/ml/datasets/adult

[9] https://www.kaggle.com/code/overload10/income-prediction-on-uci-adult-dataset/data

[10] https://www.youtube.com