Project report

Data Mining I - Project 1
Income Prediction

Full Name:

Iman Shahmoradi Najafabadi
(10018737)

Summer Semester

2019

Contents

Abstract

The Census adult database is being explored in this task to predict whether a new example make more than 50k a year. The research is conducted on the basis of steps of KDD procedures and four models of classification (DT, KNN, SVM, NB) are applied and tuned by grid searching to discover the optimal parameters of classification and balancing.

1- Introduction

There are 39072 training entries and 9769 sample cases in this Census database. It is analyzed for the first step to discover missing information and to eliminate two attributes (capital gain and capital loss) because they contain more than 92% miss leading data.

In the next step, univariate analysis will attempt to comprehend the data set characteristics and address extreme multi-class imbalances and construct new attribute for native country feature. Then the correlation matrix and chi-square test are used to analyze the interactions of attributes in order to select the suitable features to build the subsequent tailored dataset based on the Bivariate analysis' outcome.

The latest refined dataset has distinct types of data with values in distinct ranges, meaning a feature is weighted more than others. Therefore, transforming all data to the same range and type is necessary in order to make sure that different weights do not affect the algorithms of the datamining.

2- Data Understanding

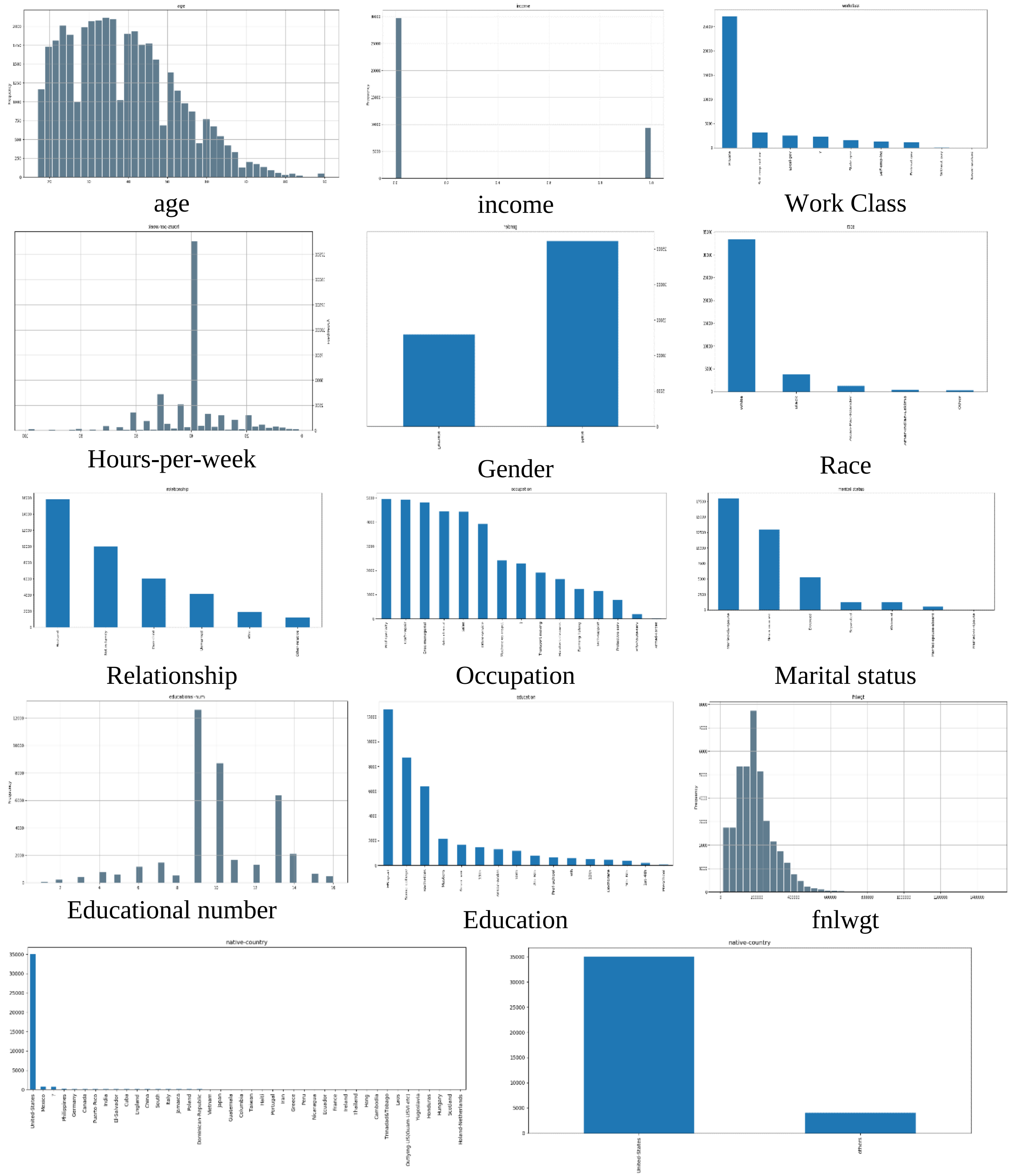
2-1- Univariate Analysis

As Shows in fig [1], All of the features in this dataset are highly imbalanced except age and occupation and this will reduce the model performance as discussed in the study of class imbalance on classification performance metrics by Amalia [1].

When attempting to classify unbalanced data sets, Data Mining classification algorithms tend to generate unsatisfactory outcomes. Compared to the complete amount of observations, the amount of observations in the interest class is very small on such datasets as they favor the majority class, leading to a high rate of misclassification for the minority interest class.

In many journals, several overviews on these problems were discussed. Such as [2], [3], [4], [5], [6], [7] that study the impact of several balancing methods such as under-sampling, over-sampling, Smote-Tomek combination, SMOTE-ENN combination.

Some technics such as over sampling and under sampling just produce or cut instances in dataset and it can be said that they are not too effective to overcome the imbalance problem but some other technics such as Tomek and ENN consider data spread properties such as classes overlap to balance the dataset. However, in this assignment just Smote-Tomek combination technic do effective to enhance classification performance metrics.



age

income

Work Class

Hours-per-week

Gender

Race

Relationship

Occupation

Marital status

Educational number

Education

fnlwgt

Native Country

Native country

Fig. (1)

2-2- Bivariate Analysis

The correlation matrices shown in Fig [2] correspond to numerical features before and after balancing illustrates that there is no correlation between fnlwgt and other features such as income so we can eliminate this attribute because it will have no effect on the learning algorithm. On the other hand, in both balanced and imbalanced correlation matrices income is strongly correlated with age, educational-num and hours-per-week.

The Chi-square test result values are organized in the upper part of table [1] and corresponded Degree Of Freedom values (Chi reference value) in the lower part (bold numbers) of the table [1] in bolded format. Comparing Chi value with the corresponded Chi reference based on Degree Of Freedom values indicates that all the categorical features relate together so all of them will affect the learning algorithm to predict the personal income. Strong relationship between gender and income as well as race and income illustrate that there still some discrimination exists against specific gender and races in this society.

	education	marital-status	occupation	relationship	race	gender	native-country	workclass	income
education		1919	18952	2909	819	333	3162	3075	5187
marital-status	**90(113)**		4059	46569	1080	8235	452	1702	7759
occupation	**210**	**84(106)**		6251	1016	6946	447	50121	4736
relationship	**75(96)**	**30(44)**	**70(90.5)**		1568	16423	565	1997	7972
race	**60(79)**	**24(36.4)**	**56(74.4)**	**20(31.4)**		533	5899	499	387
gender	**15(25)**	**6(12.6)**	**14(24)**	**5(11)**	**4 (9.49)**		4.5	901	1743
native-country	**15(25)**	**6(12.6)**	**14(24)**	**5(11)**	**4 (9.49)**	**1 (4)**		628	48
workclass	**120**	**48(65)**	**112**	**40(56)**	**32(46)**	**8 (15.51)**	**8 (15.51)**		1291
income	**15(25)**	**6(12.6)**	**14(24)**	**5(11)**	**4(9.49)**	**1 (4)**	**1(4)**	**8(15.51)**	

Table(1)

For Degree of freedom = (n-1) * (n-1) and P = 0.05 (95% confidence level) : cell
Format: **DOF (Chi reference value)**

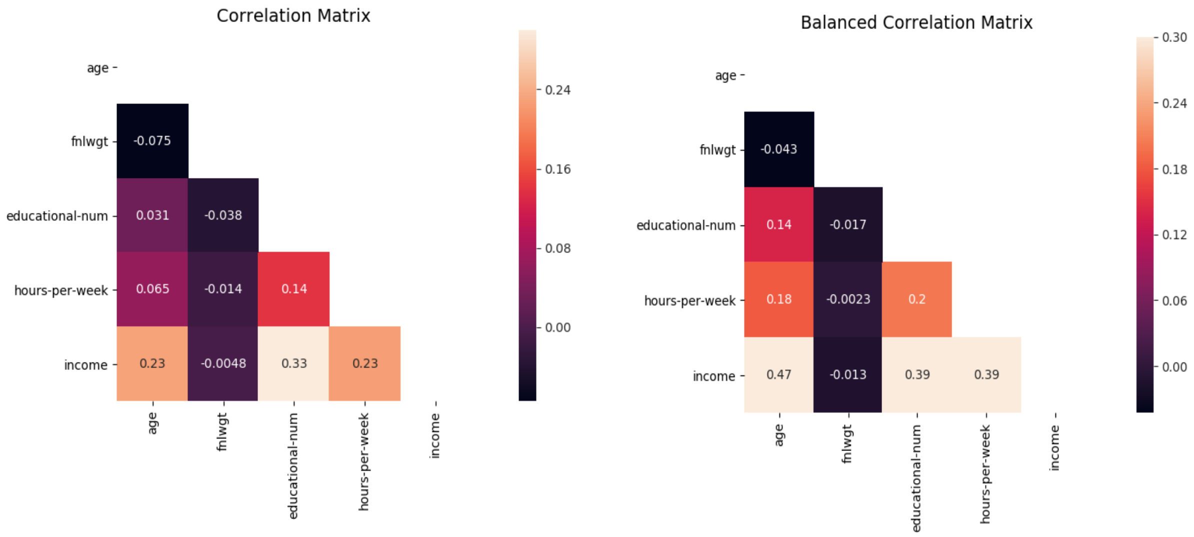


Fig (2)

3- Dataset Preparation

As discussed before the Capital-gain and Capital-loss attributes are eliminated because of 92% miss leading data and since most of the data for native country come from United states and a few data from 41 other countries, this attribute is reorganizing to two new categories United states and Others. Furthermore, the correlation matrices show that the interested attribute to predict (income) and fnlwgt not related to each other so it can be eliminated without any effect on the learning algorithm.

In this assignment we also study the effect of recategorizing the “educational-num”, “age” and “hours-per-week" to new categorizes **age**: '<30', '31-40', '41-60', '61<', for **educational-num**: '<8', '9-13', '14<' and for **hours-per-week:** '40 > ', '40', '40 < ‘. that this new categorization enhances the Decision Tree learning algorithm performance comparing to original categorization but for other Learning algorithms don’t see a noticeable effect on performance of algorithm.

Since still the original dataset is extremely imbalanced. As it is explained before the ensemble balancing Technique using **SMOTETomek [8**] python package is employed to overcome this problem.

Since the four remaining numerical attributes have a different values’ range meant different weights lead to different effect on the learning algorithm, these features’ values have been scaled to same range using **Sklearn** [9] python package.

For the last steps to made the data ready to extract corresponded learning algorithms, the categorical features should be encoded to corresponded numerical values using **Sklearn**[9] python package and furthermore the sklearn.feature_selection[9] package is employed in order to drop ineffective attributes using linear support vector machine classifier and the dataset ended up with 10 features include **['age', 'workclass', 'education', 'educational-num', 'marital-status', 'occupation', 'relationship', 'race', 'gender', 'hours-per-week'].**

4- Classification

4-1- Models

In order to extract the prediction algorithm 6 classification methods, include Decision Tree (DT), k-nearest neighbors (KNN), Complement Naive Bayes (CNB), Multinomial Naïve Bayes (MNB), Nu-Support Vector (NuSVC) and Support Vector Machine (SVC) have been employed and as ahown in table [2] the SVC has the best performance while the CNB has the worst w.r.t roc_auc score and accuracy.

The corresponding decision boundaries for all six classifiers are depicted in fig [4 a] and fig [4 b] with all and test data points respectively. Since the interested dataset still has 10 attribute even after feature reduction, we need to project them to a 2D space using Sklearn decomposition [9] python package. The decision boundaries clearly depicted the different behavior of different classification methods except CNB and MNB which have resulted approximately analogous boundaries.

4-2- Model Tuning

Every classifier has its specific parameters which affect the corresponding decision boundaries and model performance. The GridSearchCV **[9]** python package could be hired to find the best parameters combination which maximize the model performance based on preferred ranking metric (for example roc_auc_score) using cross validation technic with preferred folds (for instance 10-fold cross validation).

Comparing training and test values recorded in Table [2], SMOTEENN balancing method cause model overfitting furthermore corresponding training scores are close to 1.00 so not appropriate for parameter tuning. Therefore, we use SMOTETomek method to balance the dataset and then tune the classifier parameters.

For the sake of computation costs, for KNN, NuSVC and SVC, the tuning process is done in several steps, first with coarse range of parameter values and then with finer ranges. The process is depicted in fig [3].

4-2-1- Best Classification Parameters value

- **Decision Tree:** Max_depth= 18, min_samples_leaf=9, Criterion= gini.
- **K-Neighbors:** Minkowski power= 1, n_neighbors=12.
- **Complement Naive Bayes**: alpha=5, norm=True
- **Multinomial Naive Bayes:** alpha=2e-10, fit_prior= True
- **NuSVC:** gamma=scale, nu=0.28, kernel=rbf
- **SVC:** gamma=auto, C=100, kernel=rbf

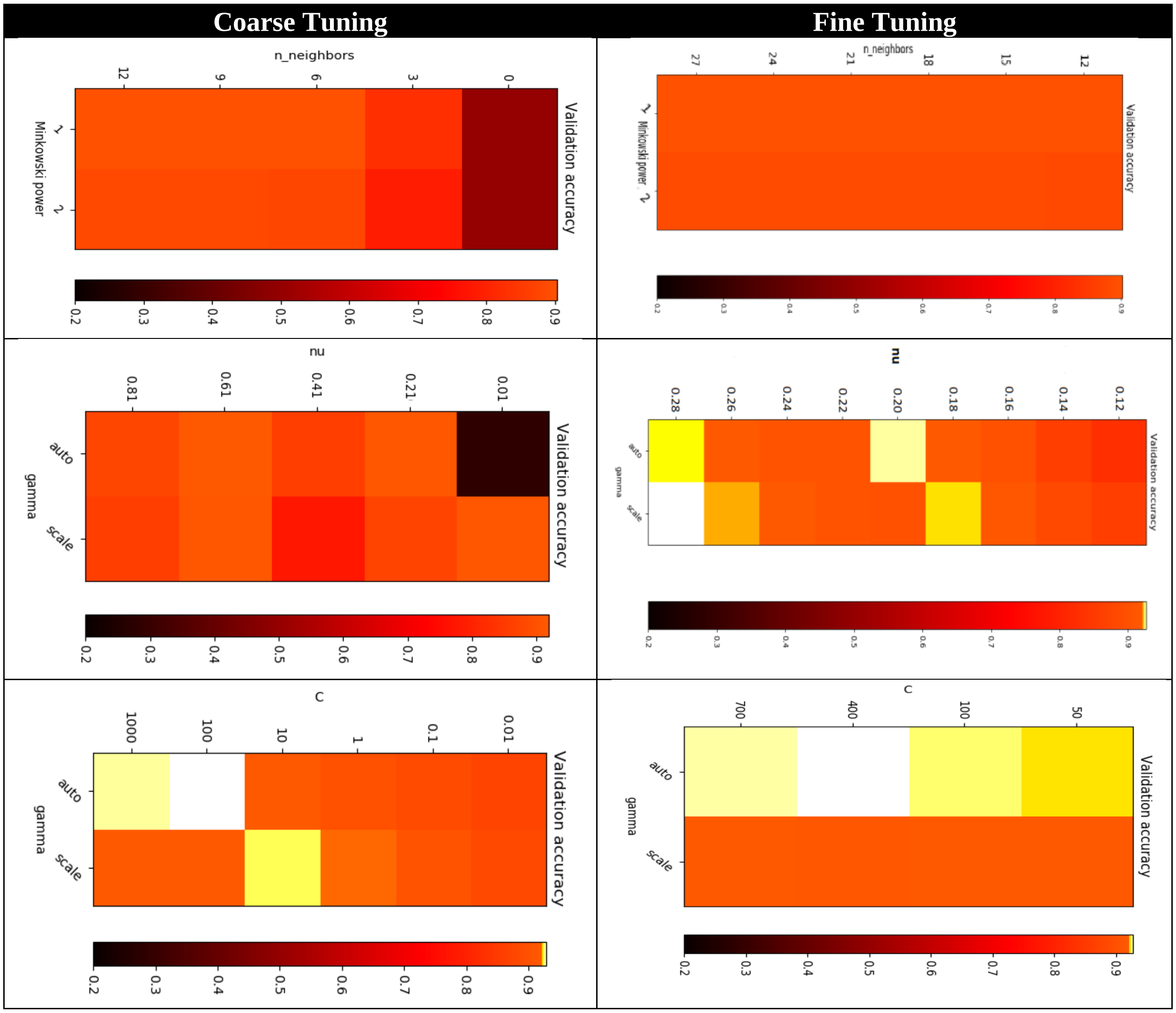


Fig (3)

Classifier	Balancing method	Accuracy	roc_auc_score (training)	roc_auc_score (test)
DT	SMOTEENN	0.78	0.99	0.79
DT	SMOTETomek	0.79	0.91	0.78
KNN	SMOTEENN	0.77	0.99	0.79
KNN	SMOTETomek	0.80	0.90	0.79
CNB	SMOTEENN	0.66	0.98	0.76
CNB	SMOTETomek	0.63	0.88	0.75
MNB	SMOTEENN	0.75	0.98	0.80
MNB	SMOTETomek	0.78	0.88	0.80
NuSVC	SMOTEENN	0.77	1.00	0.79
NuSVC	SMOTETomek	0.79	0.92	0.76
SVC	SMOTEENN	0.78	1.00	0.80
SVC	SMOTETomek	0.81	0.93	0.80
SVC	Balanced Bagging Classifier	0.79	------	0.82

Table(2)

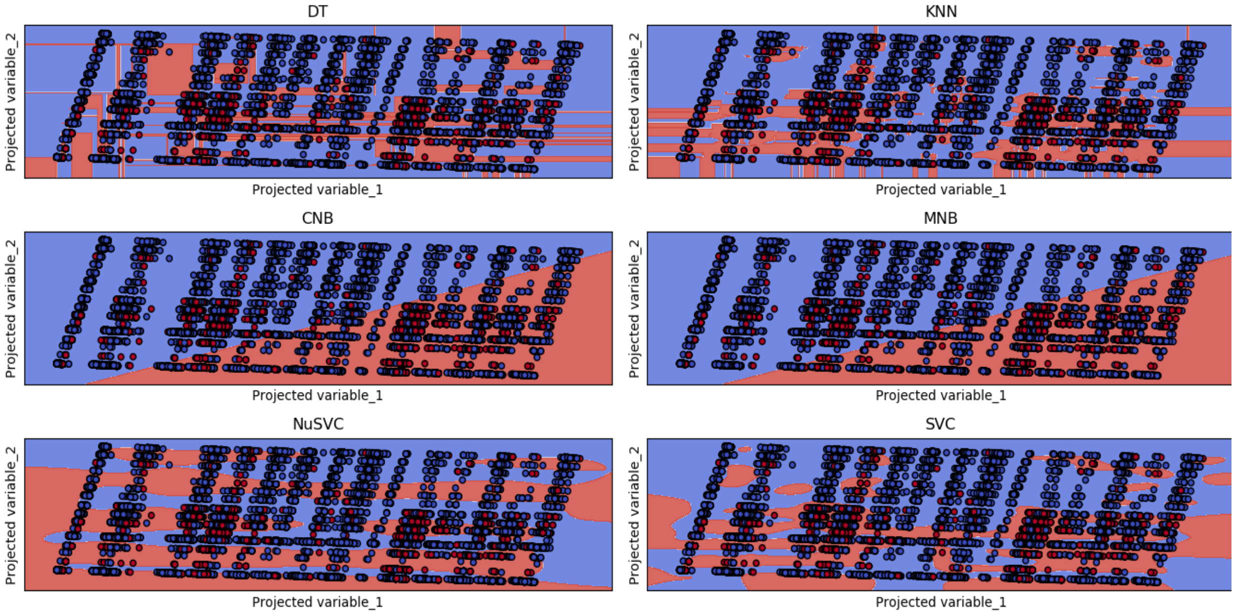


Fig (4 a)

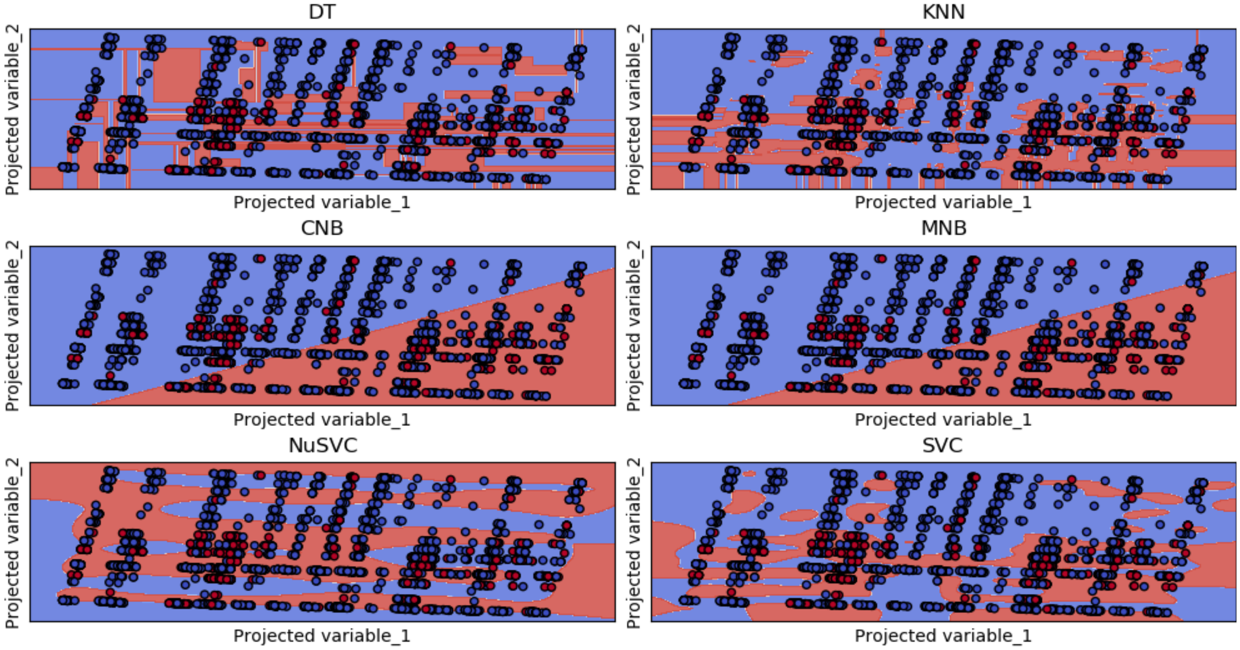


Fig (4 b)

4-2-2- Decision Tree Classifier tuning parameters

- **Criterion**: The function of measuring a split's quality. Supported criteria are the Gini impurity "gini" and the data gain "entropy."

- **Splitter:** The strategy used at each node to select the split. Supported approaches are "best" in selecting the best split and "random" in selecting the best split random.
- **max_depth :** The tree's highest depth. If None, nodes will be extended until all leaves are pure or until all leaves contain less than samples of min samples split.

- **min_samples_split:** Minimum number of samples needed to divide an internal node
- **min_samples_leaf** : The minimum amount of samples that a leaf node requires
- **min_weight_fraction_leaf** : The minimum weighted fraction of the complete amount of weights (of all input samples) that a leaf node requires. Samples have the same weight when there is no sample weight.

- **max_features** : The number of attributes to consider in search of the best split.
- **max_leaf_nodes** : Create the best-first-fashion tree with max leaf nodes. Best nodes are described as impurity relative decrease. If None, the amount of leaf nodes is unlimited.
- **min_impurity_decrease** : A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
- **min_impurity_split** : Early stop threshold in tree development. If its impurity is above the limit, a node will split, otherwise it will be a leaf.
- **class_weight** : Weights in the form { class label: weight } tied with classes. If not specified, all classes should have a same weight.

4-2-3- K-Neighbors Classifier tuning parameters

- **n_neighbors** : Number of neighbors for kneighbors queries to be used by default.
- **weights:** Function of weight used in estimates. Probable values: uniform, distance, user-defined function.
- **algorithm**: The Algorithm that employed to determine the closest neighbors.
- **p**: Minkowski metric power parameter. If p = 1, this means using manhattan distance (l1) and euclidean distance (l2) for p = 2. Use minkowski distance (l p) for arbitrary p.

- **metric**: the distance metric to use.

4-2-4- Complement Naive Bayes classifier tuning parameters

- **alpha**: smoothing coefficient.
- **norm**: Whether a second weight normalization is performed or not.

4-2-5- Multinomial Naive Bayes classifier tuning parameter

- **alpha**: smoothing coefficient.

4-2-6- Support Vector Classification tuning parameters

- **C**: Penalty parameter.
- **kernel**: Specifies the type of the kernel to use in the algorithm.
- **gamma:** Kernel coefficient.

- **shrinking**: Whether to use heuristic shrinking.
- **probability**: Whether to allow estimates of likelihood.

- **tol**: Tolerance for the criterion to stop.
- **class_weight**: The "balanced" mode uses values to adjust weights in the input data in inverse proportion to the class frequencies.

4-2-7- Nu-Support Vector Classification tuning parameters

- **nu**: An upper limit on the training error fraction and a lower limit of the support vector fraction.
- **kernel**: Specifies the type of the kernel to use in the algorithm.
- **gamma:** Kernel coefficient.
- **shrinking**: Whether to use heuristic shrinking.
- **probability**: Whether to allow estimates of likelihood.

- **tol**: Tolerance for the criterion to stop.
- **class_weight**: The "balanced" mode uses values to adjust weights in the input data in inverse proportion to the class frequencies.

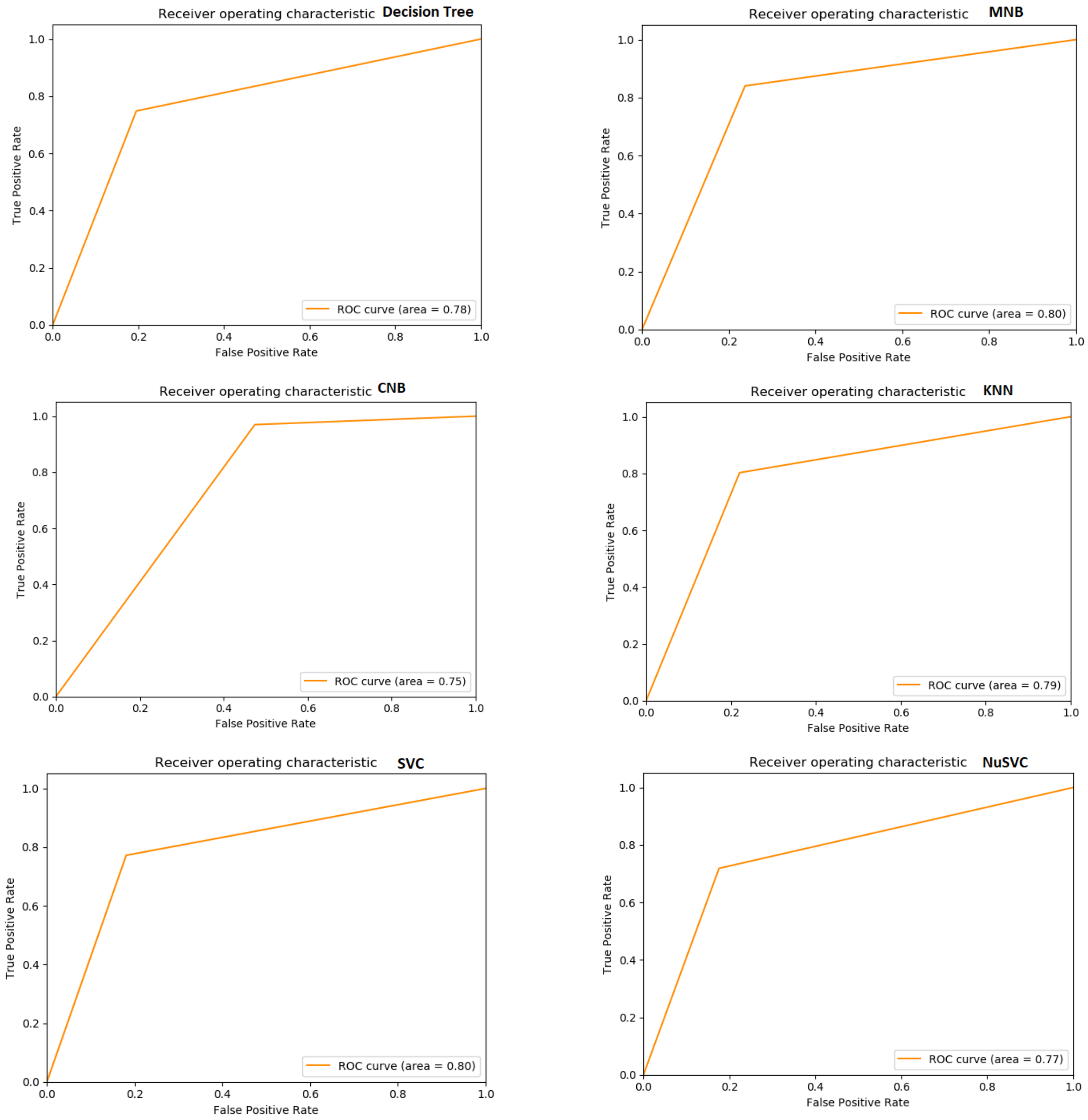
For the sake of computation speed just a few numbers of classifier parameters which are more effective to increase model performance considering overfitting could be selected for model tuning. Therefore, for **DT** tuning the **max_depth** and **min_samples_leaf** is selected and to tune **KNN** classifier the n_neighbors also for **CNB** tuning purpose we select the smoothing coefficient as well for **MNB** tuning and finally to tune **SVC** and **NuSVC** the **Penalty parameter** and **nu** have been used respectively as well **gama** for both of them.

4-3- Models Overfitting

- **Decision Trees:** Learners from the decision-tree can generate over-complex trees that do not properly generalize information. This is called being overfitted. To prevent this issue, mechanisms such as pruning, setting the minimum number of samples needed at a leaf node or setting the maximum tree depth are required.
- **KNN:** With a lower number of features KNN performs better than a large number of features. You can say that you need more data when the number of features increases. Dimensional growth also leads to the problem of overfitting.
- **SVC:** A narrower margin will be considered for higher **C** values and the decision function is able to properly classify all training points. A reduced C will support a wider margin at the expense of training precision, hence a simpler decision function. In other words," C" acts in the SVM as a parameter of regularization. On the other hand, the model's behavior is very susceptible to the **gamma**. If gamma is too large, only the support vector itself is included in the radius of the area of influence of the support vectors and no amount of regularization with C can prevent overfitting.

Conclusion

Comparing test and training scores recorded in table [2], since in all classification methods along with SMOTEENN balancing method, the test and training score values have big difference, it can be said that the SMOTEENN method cause Overfitting on this dataset. on the other hand, the SVC method has the best accuracy and best score on the test dataset so this classifier along with the SMOTETomek balancing method is the best combination to extract the prediction model of the Census adult database.



AUC plots

References

1- Amalia L, Alejandro C, Alejandro M, Ana D.L., “The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Published by Elsevier Ltd.” February 2019, DOI: 10.1016/j.patcog.2019.02.023

2- G.E. Batista , R.C. Prati , M.C. Monard , A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newslett. 6 (1) (2004) 20–29, DOI: 10.1145/1007730.1007735

3- V. Ganganwar , An overview of classification algorithms for imbalanced datasets, Int. J. Emerg. Technol. Adv. Eng. 2 (4) (2012) 42–47 .

4- H. He , E.A. Garcia , Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284 .

5- B. Krawczyk , Learning from imbalanced data: open challenges and future di- rections, Prog. Artif. Intell. 5 (4) (2016) 221–232, DOI: 10.1007/s13748-016-0094-0

6- N.V. Chawla , K.W. Bowyer , L.O. Hall , W.P. Kegelmeyer , SMOTE: synthetic mi- nority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.

7- Y. Sun , M.S. Kamel , A.K. Wong , Y. Wang , Cost-sensitive boosting for classifica- tion of imbalanced data, Pattern Recognit. 40 (12) (2007) 3358–3378.

8- imbalanced-learn python package, *https://github.com/scikit-learn-contrib/imbalanced-learn*.

9- scikit-learn python package, *https://scikit-learn.org*