Introduction to Data Science report

Nguyen Tien Vuong

MSc Computer Science, Artificial Intelligence Specialization, Eötvös Loránd University
nguyentienvuong.scorpio13@gmail.com

1 Introduction

The dataset used in this report is related with direct marketing campaigns of a Portuguese banking institution.
The dataset has 21 attributes as shown in **Table 1**, the **y** attribute will be considered as the output and the other 19 attributes (age, job, marital etc.) except for 'duration' will be considered as the input.

Table 1. Details of attributes

Attribute	Number of values	Attribute	Number of values
age	78	campaign	42
job	12	pdays	27
marital	4	previous	8
education	8	poutcome	3
default	3	emp.var.rate	10
housing	3	cons.price.idx	26
loan	3	cons.conf.idx	26
contact	2	euribor3m	316
month	10	nr.employed	11
day_of_week	5	y	2

The aim of this report is to show how to develop a prediction model to classify if the client will subscribe (yes/no) a term deposit, cluster the client into various groups, and provide some ideas on how frequent pattern mining could be utilized to uncover some patterns in the data and/or to enhance the classification. Logistic regression, K-nearest neighbors, Random Forest, KModes clustering and Apriori can be applied to solve those problem.
This report is organized as follow: Section II shows the theoretical background, Section III presents the methodology used to evaluate the proposal. Conclusion will be founded as section IV.

2 Theoretical background

2.1 K-nearest neighbors

K-nearest neighbor is one of the simplest supervised-learning algorithms (which works in some cases) in Machine Learning. When training, this algorithm does not learn anything from the training data (this is also the reason this algorithm is classified as lazy learning), all calculations are performed when it needs to predict the outcome of new data. **K-nearest neighbor** can be applied to both types of Supervised learning problems, Classification and Regression. **KNN** is also known as an Instance-based or Memory-based learning algorithm.

With **KNN**, in the Classification problem, the label of a new data point is directly inferred from the nearest K data points in the training set. The label of a test data can be decided by major voting between the nearest points, or it can be inferred by assigning a different weight to each of the nearest points and then deducing it.

In a nutshell, **KNN** is an algorithm to find the output of a new data point by relying only on the information of K data points in the training set closest to it (*K-neighbor*), regardless of whether some of these closest data points are noisy.

Advantages of **KNN**:
- The computational complexity of the training process is zero.
- Predicting the outcome of new data is simple.
- There is no need to make any assumptions about the distribution of classes.

Disadvantages of **KNN**:
- **KNN** is very sensitive to noise when K is small.
- As mentioned, **KNN** is an algorithm where all calculations are in the testing stage. In which, calculating the distance to each data point in the training set will take a lot of time, especially with databases with large dimensions and many data points. With larger K, the complexity will also increase. In addition, storing all data in memory also affects the performance of **KNN**.

2.2 Random Forest

Decision Tree

The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree [2].

Random Forest

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. In the Random Forest algorithm, a numerous number of decision trees using the Decision Tree algorithm

are built, but each decision tree will be different (with random elements). Then the resulting predictions are aggregated from the decision bodies.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

ALgorithm

1.In Random forest n number of random records are taken from the data set having k number of records.

2. Individual decision trees are constructed for each sample.

3. Each decision tree will generate an output.

4. Final output is considered based on Majority Voting or Averaging for Classification and regression respectively [1].

2.3 Logistic Regression

Logistic Regression is a classification algorithm used to assign objects to a set of discrete values (like 0, 1, 2, ...). A typical example is email classification, including work email, family email, spam email,... Online transactions are safe or unsafe, benign or malignant tumors. The algorithm uses the logistic sigmoid function to make a probability estimate. For example: This tumor is 80% benign, this transaction 90% is fraudulent,...

Sigmoid Function A function that always has a value in the interval $[0, 1]$, continuous and easy to use, is a sigmoid function.

The function is continuous and always returns a value in the range $(0, 1)$, Furthermore, there are derivatives at every point, so we can use gradient descent.

2.4 KModes Clustering

The simplest idea about a cluster is a collection of points that are close together in a certain space (this space can have many dimensions in case the information about a data point is very large). The **Fig.2** below is an example of 3 data clusters.

Assume each cluster has a representative point (center) in yellow. And the points around each center belong to the same group as that center. In the simplest way, considering any point, we consider which point is closest to the center, then it belongs to the same group as that center. At this point, we have an interesting problem: On a large square sea, there are three islands of yellow square, triangle, and circle as shown in **Fig.2**. A point in the sea is said to be within the territorial sea of an island if it is located closer to one island than to the other two.

Fig.3 is an illustration of the division of the territorial sea if there are 5 different islands represented by black circles:

We see that the lines of delimitation between the territorial seas are straight lines (more precisely, they are the medians of pairs of adjacent points). So the territorial sea of an island would be a polygon.

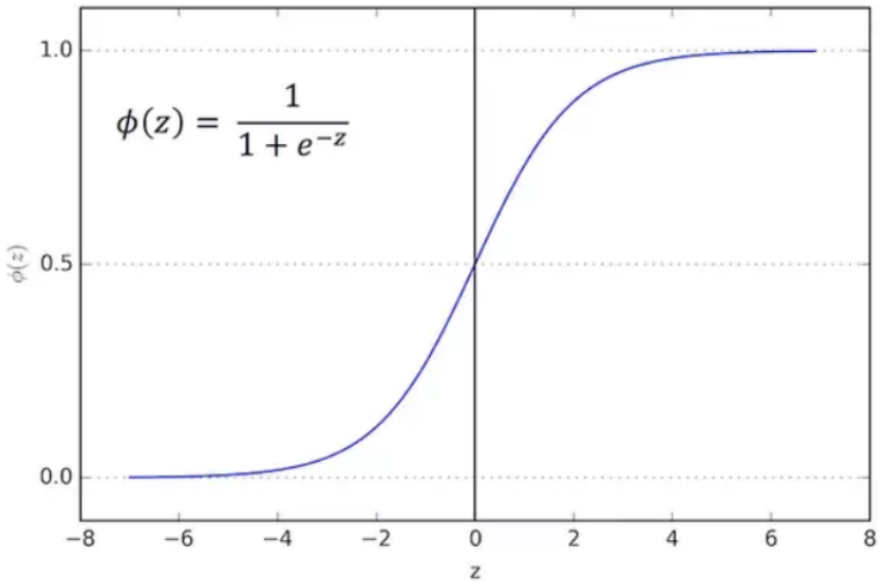


Fig. 1. Sigmoid function

This division in mathematics is called the Voronoi Diagram.

In three-dimensional space, for example planets, the (so-called) space of each planet will be a polyhedron. In more dimensional space, we will have hyperpolygons.

Algorithm

Input: Data X and the number of clusters to find K.

Output: M centers and label vectors for each data point Y.

1. Choose any K points as the initial centers.

2. Calculate the distances between each object and the cluster mode; assign the object to the cluster whose center has the shortest distance.

3. Repeat until all objects are assigned to clusters.

4. Select a new mode for each cluster and compare it with the previous mode. If different, go back to Step 2; otherwise, stop.

We can guarantee that the algorithm will stop after a finite number of iterations. Indeed, since the loss function is a positive number, and after every step 2 or 3, the value of the loss function is reduced. According to the knowledge of the sequence in the high school program: if a sequence is decreasing and bounded below, it converges! Furthermore, the number of possible groupings for the entire data is finite, so at some point the loss function will not be able to change, and we can stop the algorithm here.

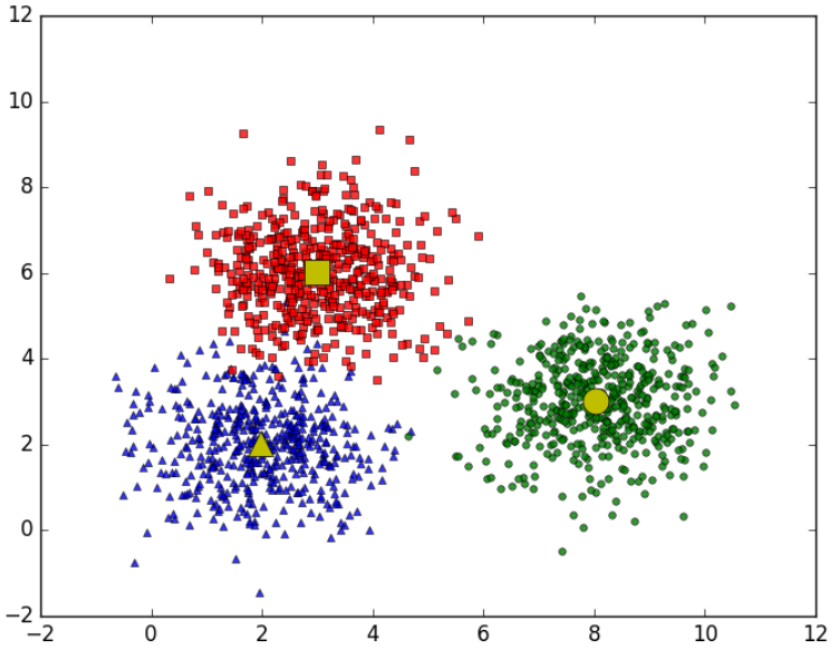


Fig. 2. Example of clusters

2.5 Apriori

The Apriori algorithm was published by R. Agrawal and R. Srikant in 1994 for finding frequent itemsets in a large dataset. The name of the algorithm is Apriori because it uses pre-existing knowledge (prior) about attributes and items that frequently appear in the database. To improve the efficiency of filtering frequent items by level, an important property is used called Apriori property which helps to reduce the search scope of the algorithm.

All non-empty subsets of the regular set must also be regular. This key concept of the Apriori algorithm is to combat the monotony of the support method.

The main idea of the Apriori algorithm is:

- Find all frequent itemsets:

k-itemset (itemset consisting of k items) is used to find (k+1) - itemset.

First find 1-itemset (symbol L_1). L1 is used to find L_2 (2-itemsets). L_2 is used to find L_3 (3-itemset) and continues until no k-itemset is found.

- From frequent itemsets generate strong association rules (association rules satisfy 2 parameters min_sup and min_conf).

Algorithm

1. Scan (Scan) the entire transaction database to get support S of 1-itemset, compare S with min_sup, to get 1-itemset (L_1)

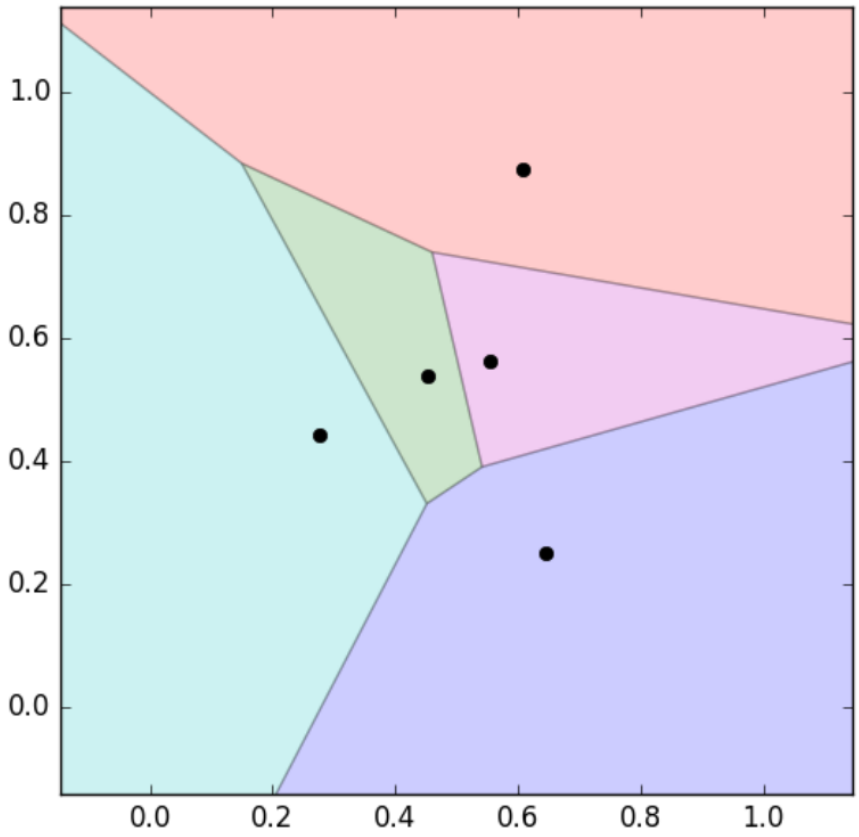


Fig. 3. Zoning of the territorial sea of each island. Different regions have different colors.

2. Use L_{k-1} to join L_{k-1} to generate candidate k-itemset. Eliminate itemsets that are not frequent itemsets to obtain k-itemset

3. Scan transaction database to get support of each candidate k-itemset, compare S with min_sup to get frequent k-itemset (L_k)

4. Repeat from step 2 until Candidate set (C) is empty (no frequent itemsets found)

5. For each frequent itemset I, generate all non-empty subsets s of I

6. For each non-empty subset s of I, generate rules s => (I-s) if its confidence (Confidence) >= min_conf

3 Method

3.1 Data preprocessing

First of all, the dataset contains 45307 rows and 21 attributes/columns with both numerical and categorical values. Furthermore, to have a deeper understanding about the data, we must know what is represented in each of the column in the dataset. In more detail, the meaning of them are *age, job, marital status, education, default credit, housing loan, personal loan, contact communication type, last contact month, last contact day of week, last contact duration, number of contacts performed during the campaign, number of days that passed by after the client was last contacted from a previous campaign, number of contacts performed before this campaign and for this client, outcome of the previous marketing campaign, employment variation rate, consumer price index, consumer confidence index, euribor 3 month rate, number of employees and output* respectively.

Next, it is necessary to investigate the dataset in order to check if there exist any missing values and those values will be treated for each column separately. Fortunately, after using `isnull()` function from `panda`, there does not exist any missing values, and because of this, we do not need to replace those missing data with new value. Furthermore, after being examined, there exists some duplicate rows in this dataset and those duplicate data are dropped. Because of that, the total number of remaining rows are 39404.

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
0	56	housemaid	married	basic.4y	no	no	no	telephone	may	mon	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
1	57	services	married	high.school	unknown	no	no	telephone	may	mon	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
2	37	services	married	high.school	no	yes	no	telephone	may	mon	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
3	40	admin.	married	basic.6y	no	no	no	telephone	may	mon	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no
4	56	services	married	high.school	no	no	yes	telephone	may	mon	1	999	0	nonexistent	1.1	93.994	-36.4	4.857	5191.0	no

Fig. 4.

Moreover, based on the dataset, there are 4598 clients subscribe to a term deposit while 34806 other people do not subscribe and the distribution of the outcome is shown below by counting the values of the output column.

As being mentioned above, the dataset contains both numerical and categorical values. So in order to use those categorical values for programming efficiently, *OneHotEncoder* is introduced to encode categorical features as a one-hot numeric array. Additionally, the numerical columns in this dataset are being standardized by removing the mean and scaling to unit variance. This standardized columns are concatenated with the values data from the categorical columns.

3.2 Classification using K-nearest neighbors

From the dataset, it is known that there are two classes of subscribing term which are "yes" and "no", but `n_neighbors` is generally an odd number if the

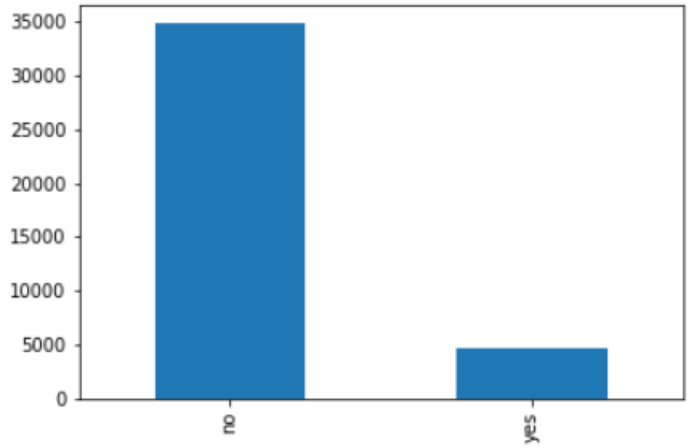


Fig. 5. Number of customers of 2 groups

number of classes is 2, so it can be set to 3 and the accuracy of the model on testing sample data is 0.38.

Moreover, **Cross validation** is used to evaluate the model and **10-Fold Cross validation** is run on the algorithm which is **KNN**. None the less, the full data of X and y are passed because *K-Fold* will split the data and automatically choose training and test sets. The the final avarage is 0.71 as shown in **Fig.6.**

```
Accuracy values for 10-fold Cross Validation:
 [0.82853462 0.8261585  0.76590927 0.81266017 0.79980778 0.72363724
 0.63966294 0.61459318 0.73673276 0.32844361]

Final Average Accuracy of the model: 0.71
```

Fig. 6. Cross validation for KNN

3.3 Classification using Logistic Regression

First of all, the parameters for `LogisticRegression()` are chosen with $C = 1, penalty = l2$ and $solver = newton - cg$. The accuracy of the model on testing sample data is 0.35, which is a little bit lower than the accuracy achieved by **KNN** method.

Similar to **Section 3.2**, **10-Fold Cross validation** is also used for **Logistic Regression**. The result is shown in **Fig.7.** where the final average is 0.75 which is higher than in **KNN**.

```
Accuracy values for 10-fold Cross Validation:
 [0.82853462 0.82853462 0.82853462 0.82853462 0.8313165  0.8349429
 0.85837059 0.47717298 0.88175152 0.34719436]

Final Average Accuracy of the model: 0.75
```

Fig. 7. Cross validation for LR

3.4 Classification using Random Forest

The accuracy after running the model on the test set is 0.38 which is similar to **KNN** and a little bit higher than **Logistic Regression**.

Similar to **Section 3.2**, **10-Fold Cross validation** is also used for **Random Forest**. The result is shown in **Fig.8.** where the final average is 0.5 which is significant worse than in **KNN** and **Logistic Regression**.

```
Accuracy values for 10-fold Cross Validation:
 [0.82853462 0.82587302 0.4388631  0.80001655 0.74088701 0.43686159
 0.27581423 0.24097398 0.29562283 0.16157953]

Final Average Accuracy of the model: 0.5
```

Fig. 8. Cross validation for RF

Furthermore, *RandomizedSearchCV* for hyperparameter tuning is used in order to achieve a better performance for **Random Forest** model. The result is slightly better compare to the original model as can be seen in **Fig.9.**.

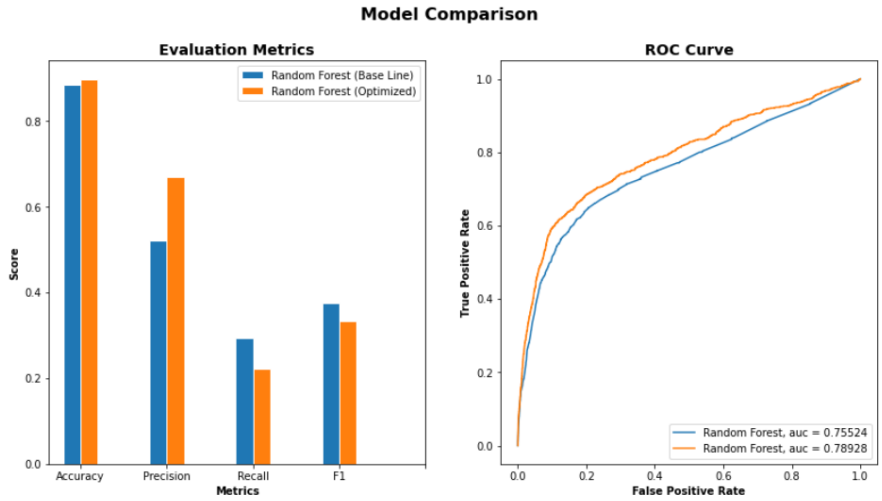
3.5 Clustering using KModes

In this section, KModes clustering is used to cluster customers into 2 groups. In addition, the main columns which are being focused on categorical columns.

Additionally, we use **Elbow Method** to determine the optimal k value for the $KModes$. As can be seen in **Fig.10.**, it shows that the most optimal value is 2, so k is set to that value.

After k is being determined, $KModes$ algorithm is run on the dataset in order to cluster the categorical data of the dataset as in **Fig.11.** For each categorical column, the values of that column is being split into 2 clusters.

From **Fig.12**, we can see each unique value in 'job' is clustered into 2 clusters 0 and 1. And the proportion of cluster 0 is higher than cluster 1 in most of the values. Furthermore, as can be seen for every other categorical columns in the notebook, cluster 0 is always has a larger amount than cluster 1.




```
Change of 1.14% on accuracy.
Change of 28.07% on precision.
Change of -24.44% on recall.
Change of -11.34% on F1 score.
Change of -8.35% on Kappa score.
Change of 4.51% on AUC.
```

Fig. 9. Cross validation for RF

3.6 Frequent pattern mining using Apriori

In this section, **Apriori** is applied in order to find frequent itemsets and association between different itemsets, that is, association rule. First of all, the "clean" dataset is converted into a list of lists because `apriori()` can only receive a list as a parameter. Furthermore, the first 100 rows is being converted because the dataset is too being so it will take a lot of time to transform and apply Apriori. Moreover, `min_support` is set to 0.02, `min_confidence` is set to 0.2, `min_lift` is set to 3 and the length is 2.

4 Conclusion

In this report, the aim is about solving real-world classification problem regarding direct marketing campaigns of a Portuguese banking institution dataset. With the help of Numpy and Sklearn dependencies in Python, **KNN**, **Logistic Regression**, **Random Forest**, **KModes**, and **Apriori** are implemented and discussed. The results showed that **KNN**, **Logistic Regression** and **Random Forest** do return the good accuracy for prediction. This probably because of the dataset is imbalanced.

In order to improve uncover some patterns in the data and/or to enhance the classification, there are many works can be done in the future. For example, we can group the data of dataset to an appropriate table, then other frequent pattern mining such as **FP Growth, Eclat** can improve the model.

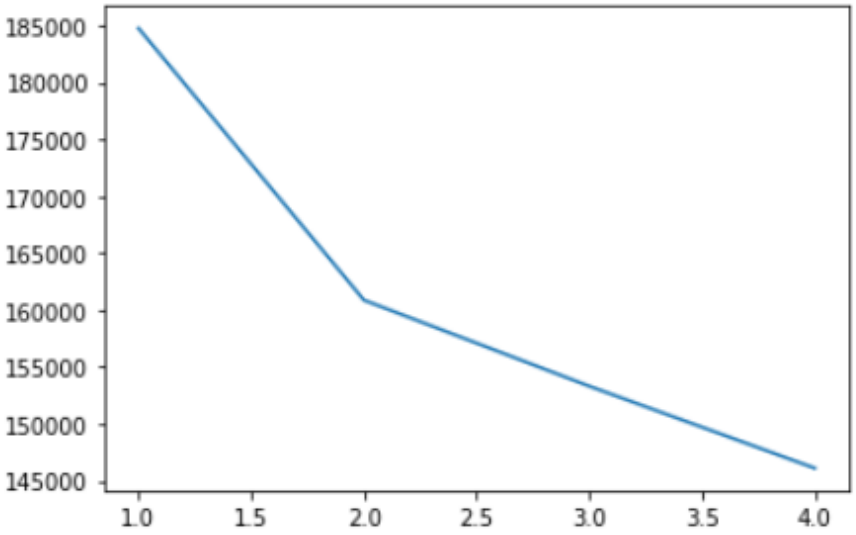


Fig. 10. Elbow Method

References

1. Sruthi E R
Understanding Random Forest.
https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
2. Akshay Sharma
Machine Learning 101: Decision Tree Algorithm for Classification
https://www.analyticsvidhya.com/blog/2021/02/machine-learning-101-decision-tree-algorithm-for-classification/

	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome	cluster_predicted
0	housemaid	married	basic.4y	no	no	no	telephone	may	mon	nonexistent	0
1	services	married	high.school	unknown	no	no	telephone	may	mon	nonexistent	0
2	services	married	high.school	no	yes	no	telephone	may	mon	nonexistent	1
3	admin.	married	basic.6y	no	no	no	telephone	may	mon	nonexistent	0
4	services	married	high.school	no	no	yes	telephone	may	mon	nonexistent	0
5	services	married	basic.9y	unknown	no	no	telephone	may	mon	nonexistent	0
6	admin.	married	professional.course	no	no	no	telephone	may	mon	nonexistent	0
7	blue-collar	married	unknown	unknown	no	no	telephone	may	mon	nonexistent	0
8	technician	single	professional.course	no	yes	no	telephone	may	mon	nonexistent	1
9	services	single	high.school	no	yes	no	telephone	may	mon	nonexistent	1

Fig. 11. Cluster

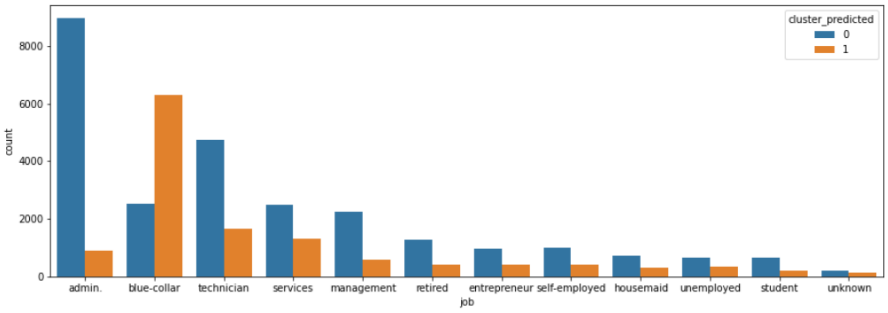


Fig. 12. Job clustering