Brief Description of Report Organization

The report will be organized as follows:

1. Introduction

a. Problem Description

b. Motivation

c. Organization

2. Data Exploration

3. Methodology

a. Preprocessing

b. Mining the Data

c. Model Performance

4. Final Updated Logic of Problem

a. LOP

b. Summary

c. Pros and Cons

d. Recommendations

5. Conclusion

6. Appendix and References

Introduction:

Problem Description

The problem this project sets out to resolve is the issue of information and decision-making when it comes to job hunting. Because of an abundance of data in the modern world, job seekers are inundated with excess information and have no good parameters to use in order to compare and contrast various jobs. Based on solely job description, the user is unable to decide if a job will be a good fit for his/her career. The problem we want to analyze is whether particular types of jobs are focused around certain areas, and how these jobs would contrast in terms of salary, job requirement, job type (full-time vs part-time), etc.

Motivation

Several factors such as job fit, company location, or job type among others are factors that are essential for job seekers, and the problem they often face is that online employment services provide far too many options that are often meaningless and provide users of these services with too many options to choose from. This often makes it difficult for job seekers to make an educated decision from the data that is presented to him or her. Thus, the main motivation behind tackling this problem is to simplify the whole process of job hunting.

Data Exploration

The ideal database we came across for job hunting was Reed\_UK.csv. However, the .csv file was not supported by our data mining tool, WEKA. Hence, we converted the .csv file into .xsl excel format and from .xsl format we then converted it into Reed\_UK.arff dataset. On analyzing the dataset we found that geo and job\_board are having the same values throughout all instances. Hence, we eliminated these two attributes. Also, we got rid of 6000 instances with reason being explained in the preprocessing step. Currently the dataset consists of following 10 attributes as headings and 14000 rows as instances to choose from.

| **Attribute** | **Data Type** | **Description** |
| --- | --- | --- |
| Category | String | The type of job falls under job category Eg. law job, hr job |
| City | String | It specifies the job requirement is for which city |
| Company Name | String | The position is offered by which company is mentioned here. |
| Job Title | String | The role requirement at the company is given as Job Title. |
| Job Type | String | The type of job offered in the company. E.g. full-time, part-time |
| Salary Offer | String | This field is used to indicate the salary offered for that position |
| State | String | It specifies the job requirement is for which state |
| Posting Date | Date | The date on which the job was posted |

It consists of all the jobs available at different organizations in the UK. This dataset is easily accessible on the below mentioned link : https://data.world/promptcloud/50000-job-board-records-from-reed-uk/workspace/file?filename =reed\_uk.csv

Parameters like job\_category, job\_type, and salary will help us classify the job postings and make our dataset decision friendly. However, we noticed some missing values, redundant instances, irrelevant attributes and duplicate data in the dataset. Therefore, we decided to clean our data using preprocessing steps.

Methodology:

1) Date Preprocessing

Some of the data present in this dataset consisted of duplicate, missing or repetitive values. All these inconsistencies were pre-processed and removed using the filters in Weka.

a) Duplicate Data

As with above, there were many duplicate job postings present in the dataset; they were all removed using the ‘RemoveDuplicate’ filter in Weka.

b) Unnecessary Attributes

Attributes such as “Geo” and “Job Board” were not needed since they only had a single value for all the instances which were UK and Reed respectively, and did not contribute much to the classification methods to be implemented. Thus, they were removed.

c) Missing Value

Some of the instances had values missing in them. Hence, they were removed using ‘SubsetByExpression’ which can be used to remove instances with missing values in an attribute. This can be done by using the expression ‘not ismissing(ATT3)’ which will remove the instance with missing values in attribute index 3.

d) Repetitive Values

Some instance values were repetitive. For example, one instance value was entered in the ‘New York’ format while other was in ‘New york’ format. To solve this issue, we used the ‘MergeTwoValues’ filter. So, both the instances will combine and appear as a single value ‘New York\_New york’

2. Mining the Data

The algorithms that were implemented in this project are done using Classification Techniques.

Classification algorithms:

We have implemented the following classification algorithms on the dataset and have obtained the following results:

i) J48

J48 is an algorithm used to generate a decision tree. J48 builds its decision trees using information gain. Here the node splits the classes based on the “gain” of the information. The attribute with the highest normalized gain is used as the splitting criteria. J48 is well known for good accuracy and high precision, hence it will be a good algorithm to implement.

ii) Naïve Bayes

Naïve Bayes is considered a probabilistic classification algorithm. Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. All of the features in our dataset such as job\_type, salary\_offered, company\_name etc. are independent. Naïve Bayes model is easy to build and particularly useful for very large datasets. Hence, we choose Naïve Bayes classifier.

iii) Lazy IBk

Lazy IBk is a K-Nearest Neighbors (KNN) algorithm. Lazy IBk is a lazy learner because it doesn't learn a discriminative function from the training data but memorizes the training dataset instead. The value of K is the number of nearest neighbors that needs to be decided. The points that will be closest or similar to each other are found using Euclidean distance. KNN works best with a lower number of features hence the value of K is 1 which is the default value of K.

All these classification algorithms have been run using their default settings in weka. The accuracies of the above-mentioned algorithms have been displayed below in the appendix section using a table.

3. Models Performance

The following are the performance measures of the model that were taken into consideration: Correctly Classified Instances, Incorrectly Classified Instances, Kappa statistic, Mean absolute error, TP Rate, FP Rate, Precision, Recall, F-Measure, ROC (Receiver Operating Characteristics), Confusion Matrix. Using these performance measures helps in determining how many instances were correctly classified; this is very helpful since this project makes use of classification algorithms like J48, Naïve Bayes and Lazy IBk. Correctly classified instances and incorrectly classified instances from the name tell us how many instances were correctly and incorrectly classified from the total number of instances. Kappa statistics tell how much better the classifier is performing comparatively. A value of 0 indicates that the classifier is useless. Mean absolute error can help determine the quality of the model by calculating the average of all errors. TP Rate is the rate of true positives (instances correctly classified as a given class). FP rate is the rate of false positives (instances falsely classified as a given class). Precision is the percentage of instances that are relevant. Recall tells us the number of records that were correctly classified. A high recall value indicates that the algorithm is returning relevant results. Confusion matrix tells us the number of classes that were correctly and incorrectly predicted. F-measure is a combined measure of precision and recall which measure a model’s accuracy on the dataset.

The main use of ROC is checking the performance of the classifier by creating a graph of True Positive vs False Positives for every classification threshold. All these measures help in determining the accuracy of the overall model. The dataset was divided into 2 parts one for training and the other for testing. 9000 records were used for training while 4000 records were used for testing. A 10-fold cross-validation method was used to train the model the way this method works is 9 folds are used for training and 1fold is used for testing. This process is repeated 10 times each time a different fold is used for testing. The performance of these models are mentioned in the tables below.

Logic of the Problem:

We are focused on evaluating how a Job seeker finds a better suited job based on the user preferences. To achieve this we have to think critically and not just go with the assumptions. We need to arrive at the best possible solution and using Critical thinking we have come up with the problems the Job hunters face by getting irrelevant job postings or the most visited postings according to the recent activity of the site, where in fact the right job posting according to the person's portfolio might be buried beneath.

Pros:

- We found the links between the ideas and relevance of the arguments.

- We approached this issue in a systematic fashion.

- We were able to find the inconsistencies in our initial arguments and ideas.

Cons:

- Critical thinking requires more in-depth knowledge of the problem to address the issue and not the symptom and thus is time consuming.

To assess the quality of our reasoning of the problem we used the Critical thinking standards where Clarity of the problem by giving examples of real life job hunting scenarios; Precision of addressing the issue and its relevance in the recent Job recruitment sites; Depth of the issue which is finding the right type of job based on the user parameters. Breadth helped us look at the issue from not only the user’s perspective but also the Job site's perspective.

Conclusion:

For classification, J48 gave the best accuracy compared to the other algorithms used. The dataset for job\_type was skewed more toward permanent, full-time job type hence the FP rate was more for this value. Both Lazy IBk and Naive Bayes got a higher accuracy in the testing dataset than compared to the training dataset.

In case of classification of class the accuracy rate was a little low an average of 50% and didn’t perform well on the testing dataset indicating that underfitting was taking place, but the ROC was high and the FP rate was low for each class.

All the models were able to classify the jobs based on the state and job\_type which can be seen from the training dataset accuracy table and testing dataset accuracy table. The value of Precision and Recall for state is 0.691 and 0.520 indicating that the TP rate is high.In this project we mainly made use of classification algorithms.

The problem is finding the right job is difficult and the current state of the world due to the pandemic has left thousands jobless. In this case how can we enable the Job search to be more efficient and valid to the user who is visiting the site instead of that person wasting his time and feeling unsatisfied by the relevancy of the job postings he sees which doesn't compliment his/her skills properly. As it is seen these classification models enable us to read the dataset properly by classifying the instances and making us realise which models worked efficiently with our dataset. We used J48 as it worked best for both training and testing data.

In the future we would like to implement a clustering algorithm where we would like to cluster each job based on their salary and job type. We would also like to implement association rule mining to which will generate rules to classify jobs based on the job title.

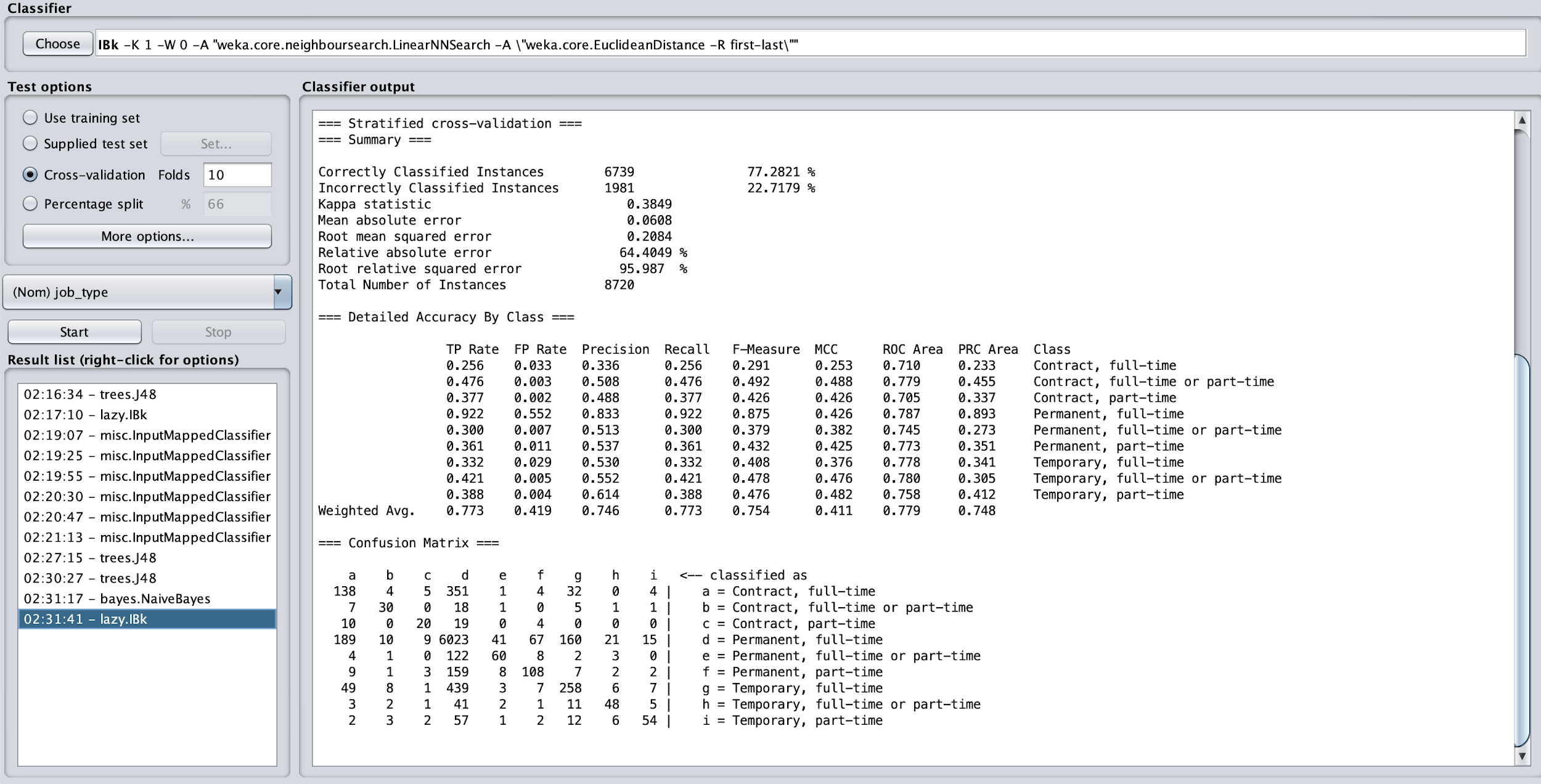
Appendix:

Training dataset accuracy:

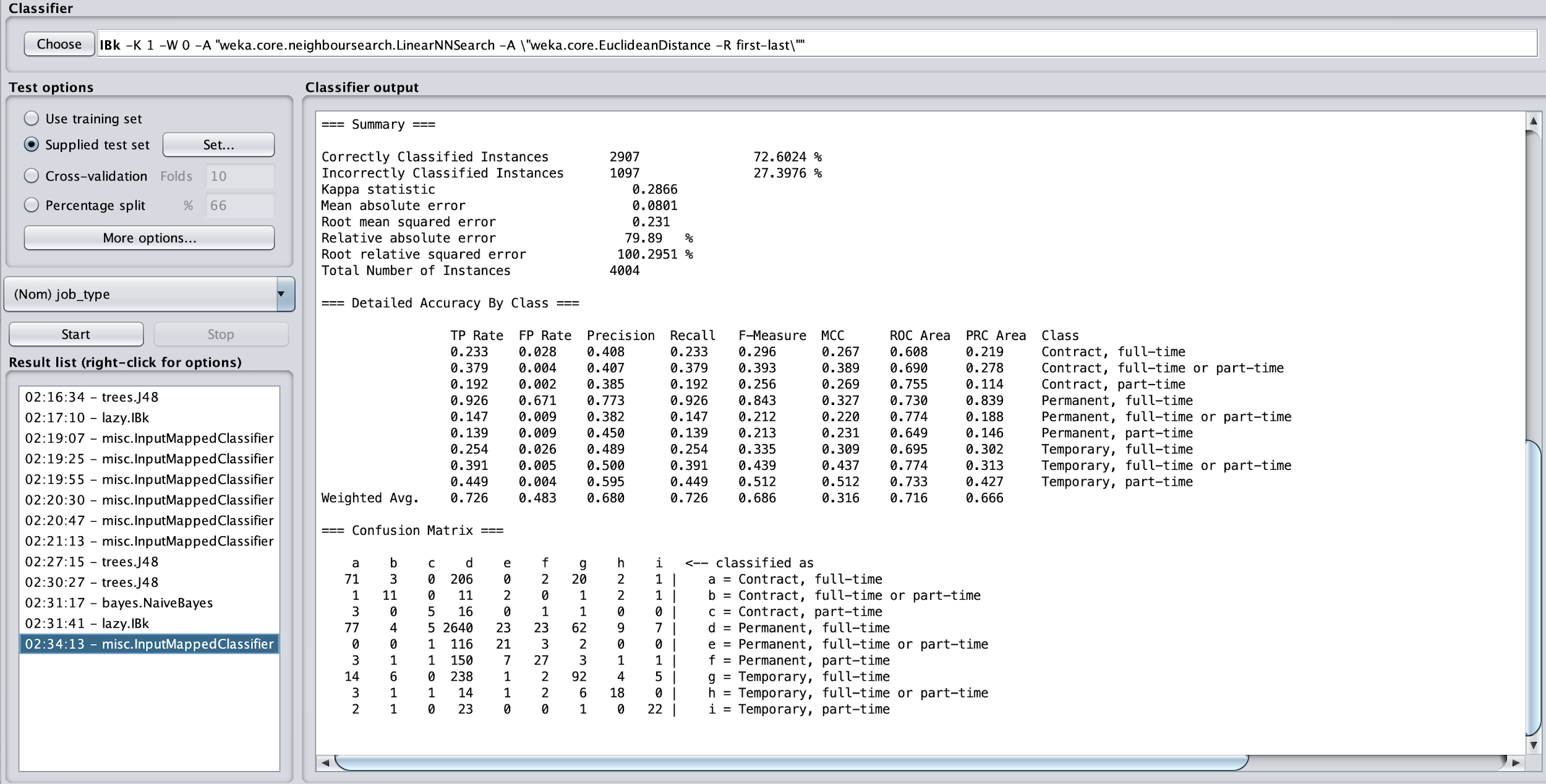
| Algorithm | state | job\_type | class |
| --- | --- | --- | --- |
| Naive Bayes | 61.99% | 78.15% | 60.81% |
| Lazy IBK | 51.79% | 77.28% | 50.75% |
| J48 | 89.98% | 74.94% | 50.20% |

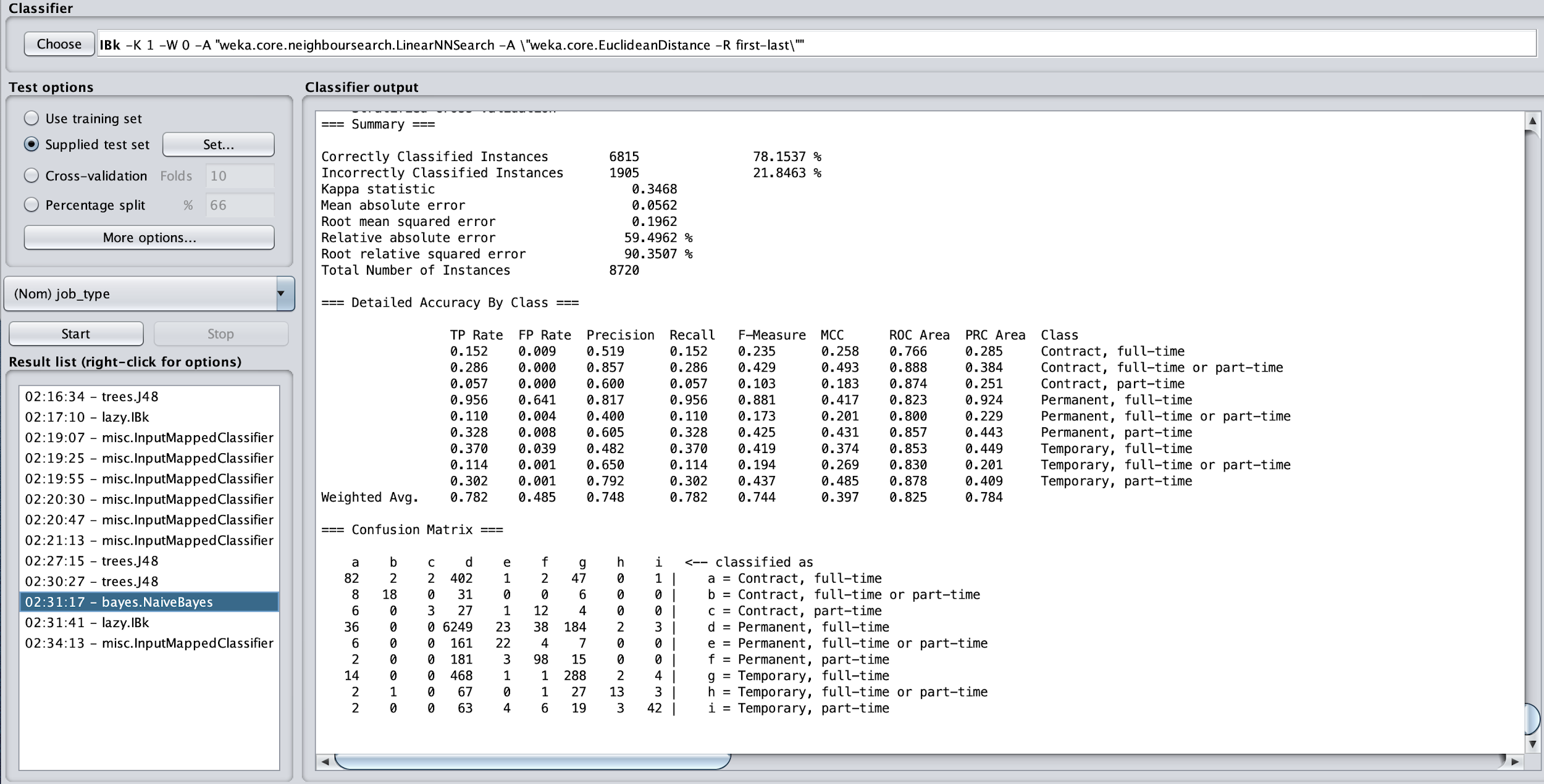
Testing dataset accuracy

| Algorithm | state | job\_type | class |
| --- | --- | --- | --- |
| Naive Bayes | 68.54% | 57.74% | 23.53% |
| Lazy IBK | 60.10% | 72.60% | 20.55% |
| J48 | 84.88% | 71.17% | 27.3227% |

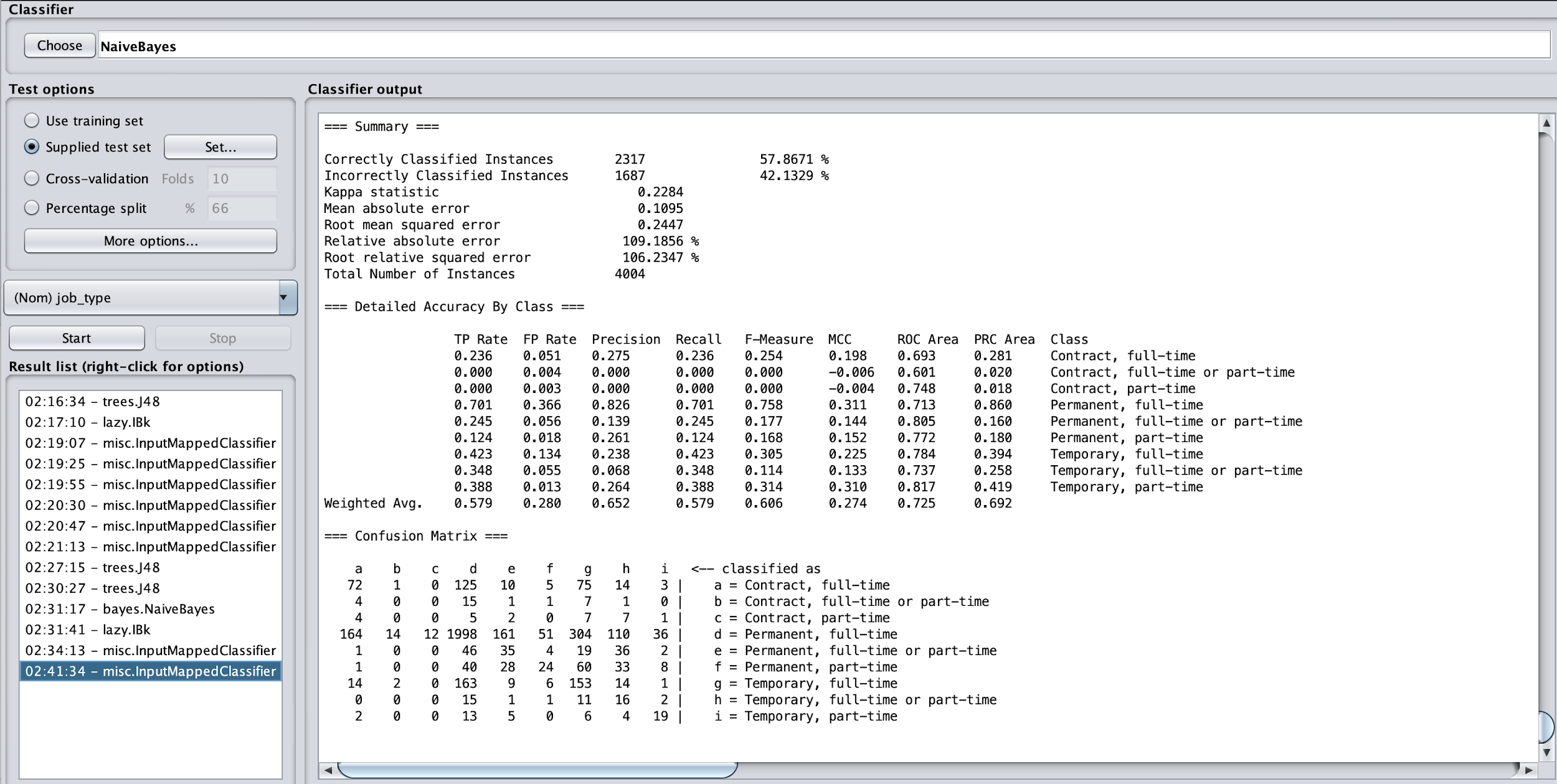


Fig[1] Screenshot of classification of job\_type using Lazy IBk algorithm on training dataset

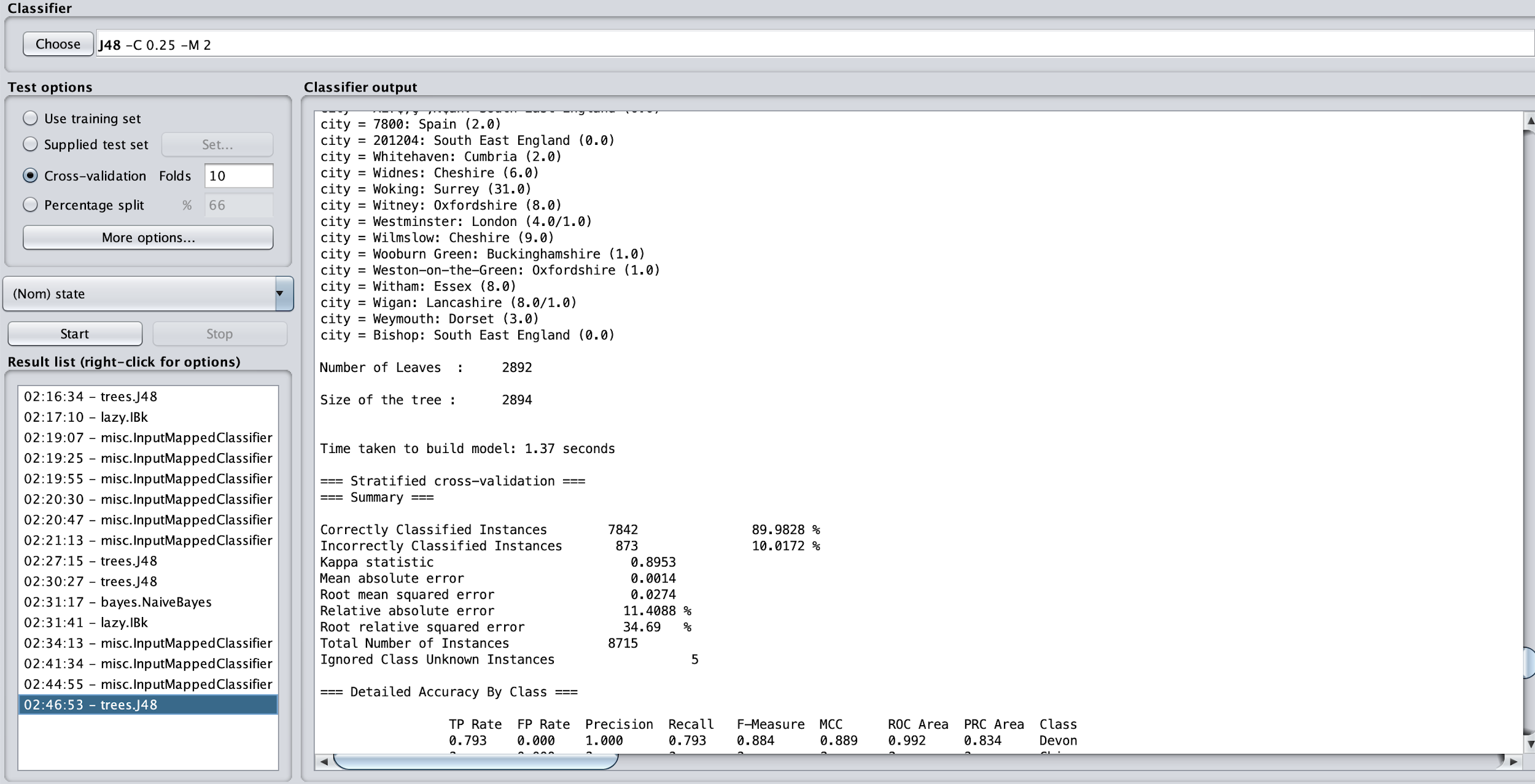
Fig[2] Screenshot of classification of job\_type using Lazy IBK algorithm on testing dataset



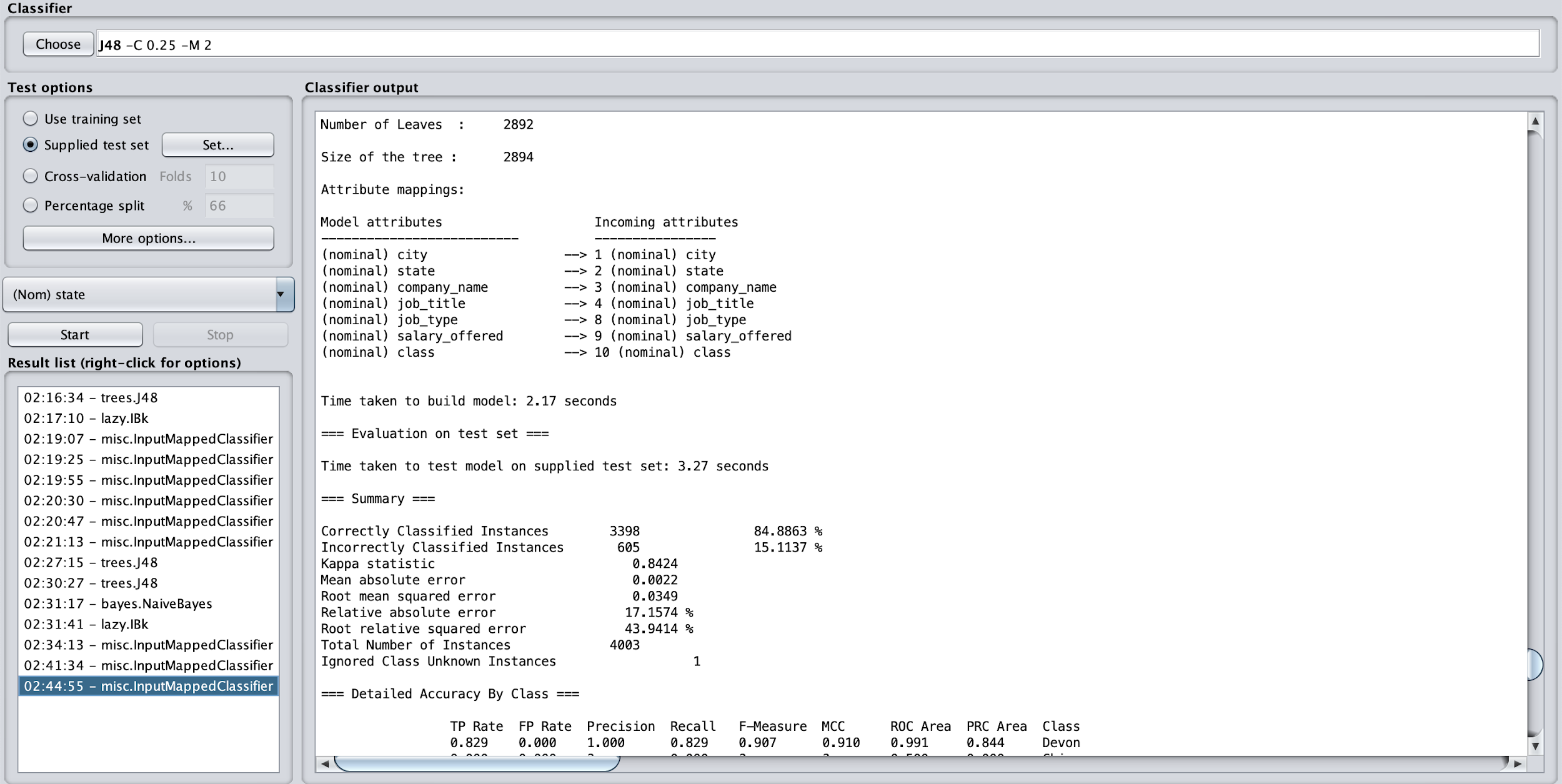
Fig[3] Screenshot of classification of job\_type using Naive Bayes algorithm on training dataset



Fig[4] Screenshot of classification of job\_type using Naive Bayes algorithm on testing dataset



Fig[5] Screenshot of classification of state using J48 algorithm on training dataset



Fig[6] Screenshot of classification of state using J48 algorithm on testing dataset

References:

[1] Karakatsanis, I., AlKhader, W., MacCrory, F., Alibasic, A., Omar, M. A., Aung, Z., & Woon, W. L. (2017). Data mining approach to monitoring the requirements of the job market: A case study. *Information Systems*, *65*, 1-6.

[2] Cheng, H. C., Lu, C. L., & Chiu, S. Y. (2012, October). Hybrid Multi-Objective PSO with Solution Diversity Extraction for job-shop scheduling management. In *2012 6th International Conference on New Trends in Information Science, Service Science and Data Mining (ISSDM2012)* (pp. 711-716). IEEE.

[3] Liu, X., Jia, S., Liu, E., & Zhang, Z. (2009, September). Application of Web-Based Data Mining in Personalized Online Recruiting System. In *2009 International Conference on Management and Service Science* (pp. 1-4). IEEE.

[4] Ding, S., & Ding, H. (2010, May). Research and development of attribute reduction algorithm based on rough set. In *2010 Chinese Control and Decision Conference* (pp. 648-653). IEEE.