Submitters:
Shay Doner 316117399 | Oz Hatuka 204066880
Mark Voskovitch 208411850 | Guy Cohen 209046077

Final Project of Data Mining Course

Abstract

Wage is one of the most important factors considered when organization offers a job offer. Also wage is important consideration for a candidate choosing whether to accept a job offer. However football players earn a varied wages. If clubs could accurately and analytically determine the player's wage based on his attributes, then clubs who are considering hire a new player could get a rough idea about his wage, and if they have room to negotiate.

Background

Today football players have a varied levels of wages which are hard to predict. Furthermore, the football clubs determine the wages of the players not by accurate measures but through popularity and demand in other clubs.
However, player's value is easy to predict, it is determined by player's attributes and achievements.
This paper describes an implementation of DM project. Real world data were collected from the FIFA 18 database.
The goal is to find a model that can predict and evaluate the player's wage by their attributes.

Problems addressed

The problem is how to predict the player wage according to his stats and abilities. Football clubs today don't have analytic tools to determine player's wage. It causes false evaluations of wages and even loss of money for the club.

Process

The preprocessing steps includes the following:

- Cleaning unimportant data for the addressed problem: nationality, club logo, flag and picture of each player.
- Delete incomplete data: players with missing attributes.
- Normalization: transform all data of the "value" and "wage" attributes into millions.
- Discretization: divide the wage attribute to 10 levels of wage by the WEKA discretization preprocess.
- We removed the nominal attributes to avoid over fitting (name, club).

After this process the data includes 17724 objects and 65 attributes (including the output target - wage).

Methods

During the Data Understanding phase, we analyzed the data main characteristics. The output presented in the report was composed from the attributes: player's features stats, age and club.

Table 1 - target attribute discretization

No.	Label	Count
1	'(-inf-0.0015]'	4198
2	'(0.0015-0.0025]'	2276
3	'(0.0025-0.0035]'	1535
4	'(0.0035-0.0045]'	1196
5	'(0.0045-0.0065]'	1700
6	'(0.0065-0.0095]'	1568
7	'(0.0095-0.0135]'	1252
8	'(0.0135-0.0205]'	1304
9	'(0.0205-0.0355]'	1368
10	'(0.0355-inf)'	1327

Table 2 - Examples of some of the 68
. player attributes

Attribute name	Description and value
name	Name of the player (nominal)
age	Age of the player (numeric≥16)
club	Player's club (nominal)
wage	Player's wage (numeric>0)
Agility	Player's agility (0≤numeric≤100)

Results

First we chose to classify the algorithms according to a minimum 23% precision because by a simple calculate – if we decide to classify all the players to the first wage group ([0-0.0015]) we will get a 23% precision. The chosen method has to have better precision. We used the Cross Validation method (10 folds) in the entire 3 algorithms.

Table 3 – algorithms results

	J-48	NaivBayes	BayesNet
TP rate	0.298	0.283	0.334
FP rate	0.093	0.099	0.095
Precision	0.295	0.272	0.314
Recall	0.298	0.283	0.334
F measure	0.296	0.271	0.321
ROC area	0.624	0.737	0.78
MAE	0.1422	0.1438	0.1336
RMSE	0.3519	0.3357	0.3327

- J48 – we chose the best output of this algorithm according to a different confidence level and min num objects of each tree branch, and eliminated those with low precision (under 23%)

- Naivbayes – this algorithm didn't produce the best result in any category.

- BayesNet - this algorithm that based on Bayes' Theorem produced the best results. It is top rated in all categories, except FP rate.

RMSE is a main criterion in selecting estimators, so we decide to relay on its result, but the differences between the algorithms are negligible so we also decided to relay on this parameters:

- ROC area, Precision and F measure are parameters we considered in selecting the best algorithm.
- The best results for FP rate, MAE, RMSE should be the lowest.

According to Table 3, the best algorithm is the BayesNet.

We focused on the first and last wage groups because they are more useful in business models (the most expensive players and the cheapest players)

Table 4 – first wage group ROC curve

Wage level: [-inf-0.0015]

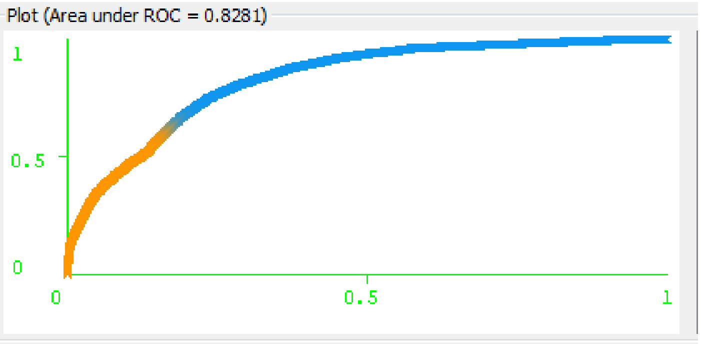
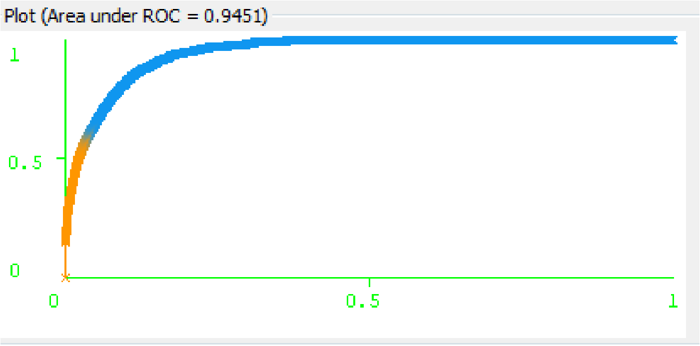


Table 5 – last wage group ROC curve

Wage level: [0.0355-inf]



Business meaning

According to the results, a football club management could use this algorithm to classify the players by their skills, and determine their wage in analytic form. In addition a club management could use this data to negotiate with players about their wage, and when they appealing to hire a player in their contract.

Conclusions

Football club management with low budget or interested in low wage players can use our model (table 4) to evaluate the players wage according to their skills.

Likewise football club management who are interested in skilled players and that can afford themselves to hire those players with high wages demands as our model shows (table 5).

Appendix process

- Select a football player's database.
- Choose a business goal that will compatible with the database. The goal: determine the player's wage.
- Determine the problem type: clustering or classification. We found out that this is a classification problem.
- Preprocess - we removed the: nominal attributes that caused over-fitting, irrelevant attributes to the business goal, incomplete data. Also wage normalization, and 10 bins discretization.
- Run the data in weka using relevant algorithms (bayesnet, naivbayes, j48, etc.)

J48

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5275               29.7619 %
Incorrectly Classified Instances     12449               70.2381 %
Kappa statistic                          0.1968
Mean absolute error                      0.1422
Root mean squared error                  0.3519
Relative absolute error                 81.1421 %
Root relative squared error            118.8654 %
Total Number of Instances            17724

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.566     0.149      0.542     0.566     0.554      0.732    '(-inf-0.0015]'
                 0.253     0.114      0.246     0.253     0.249      0.599    '(0.0015-0.0025]'
                 0.16      0.085      0.151     0.16      0.156      0.538    '(0.0025-0.0035]'
                 0.094     0.065      0.094     0.094     0.094      0.515    '(0.0035-0.0045]'
                 0.166     0.085      0.172     0.166     0.169      0.548    '(0.0045-0.0065]'
                 0.166     0.078      0.172     0.166     0.169      0.553    '(0.0065-0.0095]'
                 0.148     0.061      0.155     0.148     0.151      0.561    '(0.0095-0.0135]'
                 0.166     0.061      0.178     0.166     0.172      0.592    '(0.0135-0.0205]'
                 0.24      0.061      0.247     0.24      0.243      0.641    '(0.0205-0.0355]'
                 0.521     0.035      0.544     0.521     0.532      0.783    '(0.0355-inf)'
Weighted Avg.    0.298     0.093      0.295     0.298     0.296      0.624

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j   <-- classified as
 2377  784  374  214  187  108   46   45   45   18 |    a = '(-inf-0.0015]'
  825  576  354  206  165   91   37   15    7    0 |    b = '(0.0015-0.0025]'
  388  341  246  161  149  112   62   54   17    5 |    c = '(0.0025-0.0035]'
  253  216  152  112  144  118   94   67   33    7 |    d = '(0.0035-0.0045]'
  200  220  164  152  283  254  165  140   99   23 |    e = '(0.0045-0.0065]'
  119  112  136  120  266  261  190  187  145   32 |    f = '(0.0065-0.0095]'
   80   50   84   97  151  197  185  152  167   89 |    g = '(0.0095-0.0135]'
   63   29   77   74  173  177  160  216  203  132 |    h = '(0.0135-0.0205]'
   51   11   32   55   99  145  174  199  328  274 |    i = '(0.0205-0.0355]'
   32    3    5    3   31   57   84  138  283  691 |    j = '(0.0355-inf)'
```

NaivBayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5021               28.3288 %
Incorrectly Classified Instances     12703               71.6712 %
Kappa statistic                          0.179
Mean absolute error                      0.1438
Root mean squared error                  0.3357
Relative absolute error                 82.0334 %
Root relative squared error            113.3914 %
Total Number of Instances            17724

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.51      0.171      0.48      0.51      0.495      0.788    '(-inf-0.0015]'
                 0.239     0.112      0.239     0.239     0.239      0.728    '(0.0015-0.0025]'
                 0.196     0.107      0.148     0.196     0.169      0.68     '(0.0025-0.0035]'
                 0.012     0.007      0.106     0.012     0.021      0.644    '(0.0035-0.0045]'
                 0.166     0.104      0.145     0.166     0.155      0.654    '(0.0045-0.0065]'
                 0.133     0.063      0.17      0.133     0.15       0.671    '(0.0065-0.0095]'
                 0.125     0.067      0.124     0.125     0.125      0.695    '(0.0095-0.0135]'
                 0.074     0.024      0.193     0.074     0.107      0.733    '(0.0135-0.0205]'
                 0.355     0.118      0.201     0.355     0.257      0.782    '(0.0205-0.0355]'
                 0.595     0.041      0.537     0.595     0.565      0.917    '(0.0355-inf)'
Weighted Avg.    0.283     0.099      0.272     0.283     0.271      0.737

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j   <-- classified as
 2141  957  567    8  253   97   61   15   67   32 |    a = '(-inf-0.0015]'
  732  544  473   16  314  101   74    8   14    0 |    b = '(0.0015-0.0025]'
  360  273  301   18  281  123  111    7   60    1 |    c = '(0.0025-0.0035]'
  254  181  215   14  193   92  115   22  108    2 |    d = '(0.0035-0.0045]'
  287  170  213   23  283  208  232   54  213   17 |    e = '(0.0045-0.0065]'
  233   98  135   21  233  209  201   72  326   40 |    f = '(0.0065-0.0095]'
  150   37   63    8  162  145  157   63  377   90 |    g = '(0.0095-0.0135]'
  153   17   42   11  116  138  146   96  442  143 |    h = '(0.0135-0.0205]'
   90    1   19   13   97   89  120   98  486  355 |    i = '(0.0205-0.0355]'
   57    0    1    0   17   24   51   63  324  790 |    j = '(0.0355-inf)'
```

Bayesnet

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        5927               33.4405 %
Incorrectly Classified Instances     11797               66.5595 %
Kappa statistic                          0.2332
Mean absolute error                      0.1336
Root mean squared error                  0.3327
Relative absolute error                 76.2314 %
Root relative squared error            112.393  %
Total Number of Instances            17724

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.624     0.174      0.526     0.624     0.571      0.828    '(-inf-0.0015]'
                 0.327     0.117      0.291     0.327     0.308      0.762    '(0.0015-0.0025]'
                 0.22      0.096      0.178     0.22      0.197      0.712    '(0.0025-0.0035]'
                 0.047     0.029      0.104     0.047     0.064      0.67     '(0.0035-0.0045]'
                 0.191     0.087      0.188     0.191     0.19       0.708    '(0.0045-0.0065]'
                 0.157     0.056      0.213     0.157     0.181      0.731    '(0.0065-0.0095]'
                 0.141     0.048      0.184     0.141     0.16       0.749    '(0.0095-0.0135]'
                 0.158     0.05       0.202     0.158     0.177      0.789    '(0.0135-0.0205]'
                 0.328     0.066      0.293     0.328     0.309      0.843    '(0.0205-0.0355]'
                 0.579     0.036      0.563     0.579     0.571      0.946    '(0.0355-inf)'
Weighted Avg.    0.334     0.095      0.314     0.334     0.321      0.78

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j   <-- classified as
 2619  883  349   59  111   49   23   27   51   27 |    a = '(-inf-0.0015]'
  840  744  429   67  135   38   12   10    1    0 |    b = '(0.0015-0.0025]'
  386  380  338   94  184   86   35   18   13    1 |    c = '(0.0025-0.0035]'
  270  208  254   56  186   87   61   51   22    1 |    d = '(0.0035-0.0045]'
  246  199  240  103  324  245  143  124   68    8 |    e = '(0.0045-0.0065]'
  185  104  164   68  298  246  169  185  123   26 |    f = '(0.0065-0.0095]'
  119   30   75   44  223  140  177  169  196   79 |    g = '(0.0095-0.0135]'
  131    8   34   38  154  154  170  206  278  131 |    h = '(0.0135-0.0205]'
   87    2    8   11   95   88  135  171  449  322 |    i = '(0.0205-0.0355]'
   95    0    3    1    9   20   36   61  334  768 |    j = '(0.0355-inf)'
```

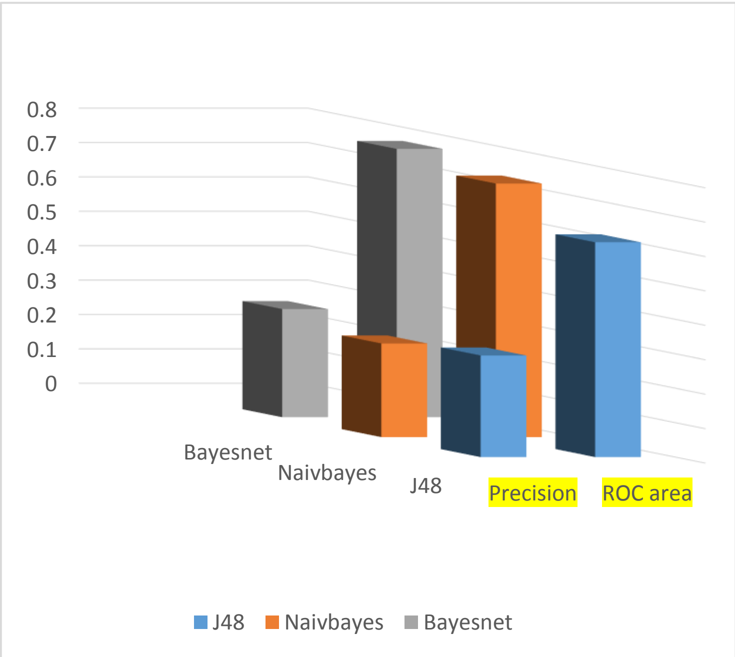
Appendix detailed results

According to our business meaning we need these parameters to be the highest:

Table 6 – main parameters considered

	J48	Naivbayes	Bayesnet
Precision	0.295	0.272	0.314
ROC area	0.624	0.737	0.78

Table 7 – main parameters graph



As shown in table 7 we can see that the bayesnet algorithm giving the best results, so we concluded that this is the best algorithm for our business meaning.

Table 8 – multiple roc curves

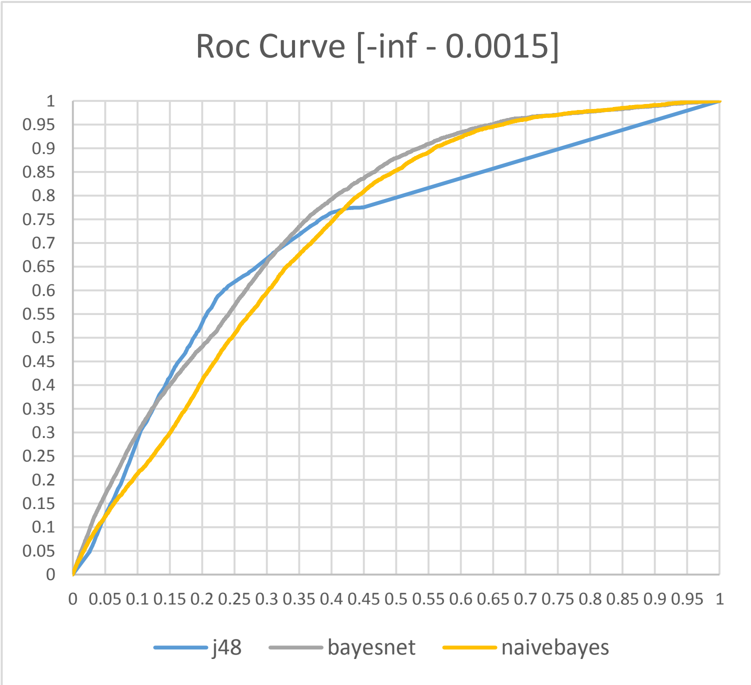
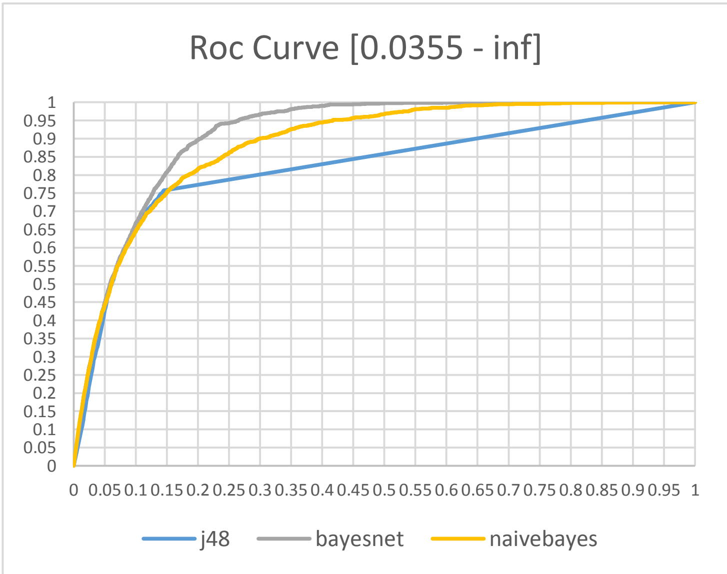


Table 9 – multiple roc curves



As we can see in tables 8-9 the Bayesnet has the largest roc curve area compare to the other algorithms. Those graphs verify our claim that Bayesnet is the best algorithm for this business meaning.