# Project work on Intelligent Systems

## Introduction

In this project, I began with a dataset of candidates for a job with the goal of training several models, finding the best one and evaluating the fairness looking for biases and discriminated categories.

## Dataset

The first thing to do is explore the dataset to get an idea of what I got in hand. There were 865 different jobs and for each one the 10 best candidates.

Each row has the following features:

* cand\_id: candidate identifier
* job\_id: job position identifier
* distance\_km: distance (km) between candidate domicile and work location
* match\_score: score provided by matching algorithm [0-100]; higher the score better the match
* match\_rank: rank based on match\_score
* cand\_gender: candidate gender [Male,Female]
* cand\_age\_bucket: candidate age bucket ["15-24", "25-34", "35-44", "45-54", "55-74", blank]
* cand\_domicile\_province: candidate domicile province abbreviation
* cand\_domicile\_region : candidate domicile region
* cand\_education: candidate list of educational qualifications
* job\_contract\_type: job position contract type ["Lavoro subordinato", "Ricerca e selezione", "Other"]. "Lavoro subordinato" means temporary work meanwhile "Ricerca e selezione" means permanent work
* job\_professional\_category: Job position professional categoryjob\_sector:
* job\_work\_province: job position work location province abbreviation

The dataset contained 8647 values, making it impractical to analyze manually. Therefore, I looked at the missing values. For example, three candidates had missing “cand\_domicile\_province”. Additionally, a lot of candidates haven’t specified their education so only in that column I replaced missing values with "no info".

The "cand\_id" column was not useful for the matching, as the candidate’s details were described by other columns. Same for "job\_id".

After the preprocessing, the dataset had 8639 entries.

Another important aspect was that columns contained string data, only “distance\_km” and “match\_score” were numerical. In order to solve this problem I applied a lebel encoder toa ll categorical columns.

Number of categories per column:

* cand\_gender: 2
* cand\_age\_bucket: 5
* cand\_domicile\_province: 78
* cand\_domicile\_region: 18
* cand\_education: 434
* job\_contract\_type: 3
* job\_professional\_category: 247
* job\_sector: 26
* job\_work\_province: 53

Some columns had a lot of categories which will have a negative impact on the results, as I will show later.

At this point, the data was ready to be used.

## Classification

The first thing I tried was classifying the data on the “match\_rank” column. This resulted in a classification problem where the accuracy was around 10%. The poor results suggested that the classifier made a random guess among the 10 classes every time.

## Regression

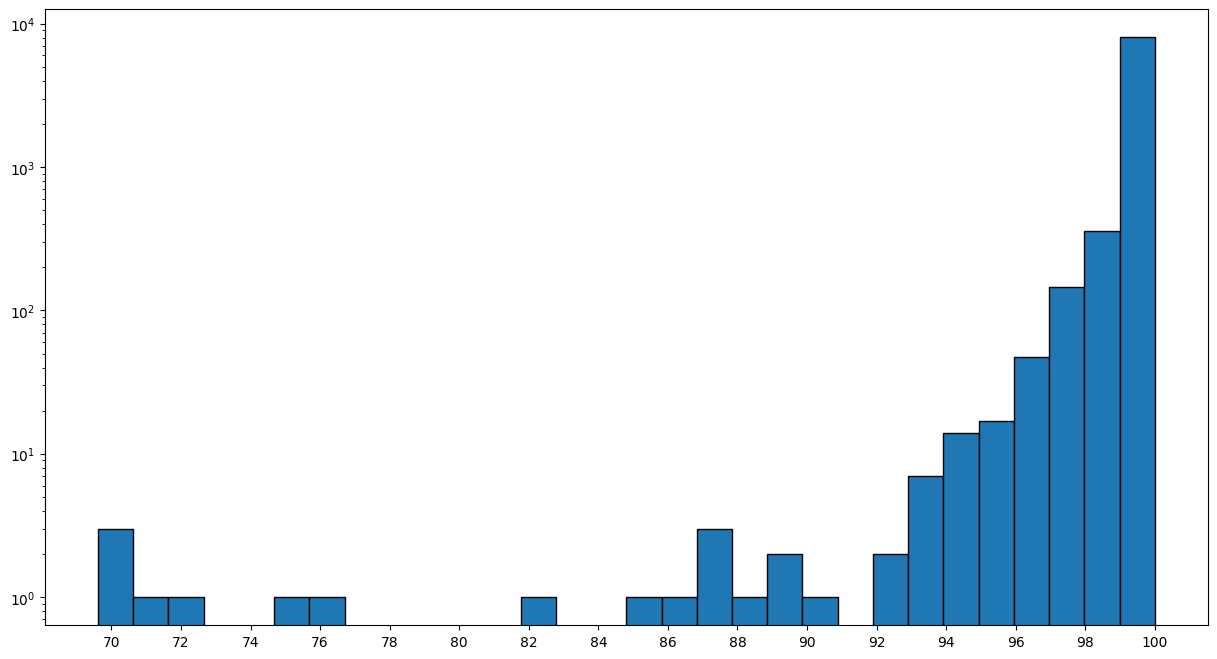
After the initial failure my idea was to use the only other target column “match\_score”. This time it was a regression problem where a model tried to guess the match score with minimal error.

The goal for now was finding the best regressor among 5 models:

* DecisionTreeRegressor
* RandomForestRegressor
* LinearRegression
* GradientBoostingRegressor
* SVR

After splitting the dataset into train and test, I tested a simple DecisionTreeRegressor:

* R^2 Score: -0.25
* Mean Squared Error: 1.20
* Root Mean Squared Error: 1.10
* Mean Absolute Error: 0.24

Looking at the results, the R^2 Score is very poor. To evaluate how good was the MAE (Mean Absolute Error) I looked into the target column “match\_score”.

Even when plotted on a logarithmic scale, the values were skewed towards the upper range (97-100). I thought that eliminating the outliers could improve the performance of the model. Therefore, I used the IQR technique eliminating 928. As a result, the dataset was reduced to 7711 entries. Immagine che contiene schermata, diagramma, Diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

Another step was identifying the least useful features in order to eliminate them.

Immagine che contiene testo, schermata, schermo, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

This is the result of DecisionTreeRegressor.

So, comparing the 5 models all agreed on three columns to eliminate:

* Cand\_education
* Cand\_gender
* Cand\_age\_bucket

As for education, as I hypothesized earlier, too many categories had a bad impact on the regression. Meanwhile, gender and age bucket weren’t that relevant for the prediction. This meant that the models didn’t use them that much which is a good thing (the model was unbiased by these two features).

The solution was to drop the columns , which indeed led to improved results.

Now I’ll show the results of each one with the corresponding MAE and a graph of all the error distribution.

DecisionTreeRegressor (Mean Absolute Error: 0.070)

Immagine che contiene diagramma, Diagramma, schermata, testo

Il contenuto generato dall'IA potrebbe non essere corretto.

RandomForestRegressor (Mean Absolute Error: 0.063)

Immagine che contiene diagramma, Diagramma, schermata, testo

Il contenuto generato dall'IA potrebbe non essere corretto.

LinearRegression (Mean Absolute Error: 0.115)

Immagine che contiene diagramma, testo, Diagramma, schermata

Il contenuto generato dall'IA potrebbe non essere corretto.

GradientBoostingRegressor (Mean Absolute Error: 0.094)

Immagine che contiene diagramma, Diagramma, testo, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

SVR (Mean Absolute Error: 0.109)

Immagine che contiene diagramma, testo, Diagramma, linea

Il contenuto generato dall'IA potrebbe non essere corretto.

## Fairness

To evaluate the fairness of the best model I used two metrics: Disparate Impact and Statistical Parity. I adapted the original formulas for the regression, where “A” rappresents a category and “nonA” all the other categories in a coumn:

* Disparate impact = mean(A) / mean(nonA)
* Statistical parity = mean(A) – mean(nonA)

Before applying the formulas, there was one more step to do: “distance\_km” contained continuous values so the column didn’t really have categories. In order to solve this problem I discretized the column in 5 classes:

* "very close": 0-20
* "close": 20-40
* "mid": 40-60
* "far": 60-80
* "very far": 80-100

The final step was looking at the results. Here I’ll show the fairness of two columns: “distance\_km” and “cand\_domicile\_regiorn”.

Immagine che contiene testo, schermata, Carattere, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

Immagine che contiene testo, schermata, menu, numero

Il contenuto generato dall'IA potrebbe non essere corretto.

In both cases there are some categories that have a mean score at least 10% higher or lower than the others. For example, “very far” is the only category in “distance\_km” with a 16% higher score, while several categories are discriminated in “cand\_domicile\_regiorn”:

* FRIULI VENEZIA GIULIA, MARCHE, UMBRIA, SARDEGNA have a lower score
* LOMBARDIA, VALLE D’AOSTA MOLISE have a higher score

Some of the categories have very few occurrences so that could explain the low score. However, LOMBARDIA is the category with the most occurrences and has a score 11% higher compared to the other categories.