Report on learning practice # 3

Sampling of multivariate random variables

Performed by:

Grandilevskii Aleksei

J4133c

Dubinin Ivan

Sorokin Mikhail

J4132c

Saint-Petersburg

2020

Table of contents:

1. Substantiation of chosen sampling.

```
# Used features and data preparation
source_df = source_df[[ 'blueTotalHeal', # 1st target variable
                        'blueTotalGold', # 2nd target variable
                        'blueChampionDamageDealt', # 3rd target variable
                        'blueTotalMinionKills', # This and the rest are predictors
                        'blueWardPlaced',
                        'gameDuraton',
                        'blueWardkills',
                        'blueKills',
                        'blueDeath',
                        'blueJungleMinionKills',
                        'blueObjectDamageDealt',
                        'blueWins']]
```

Pic.1. Dataset preparation.

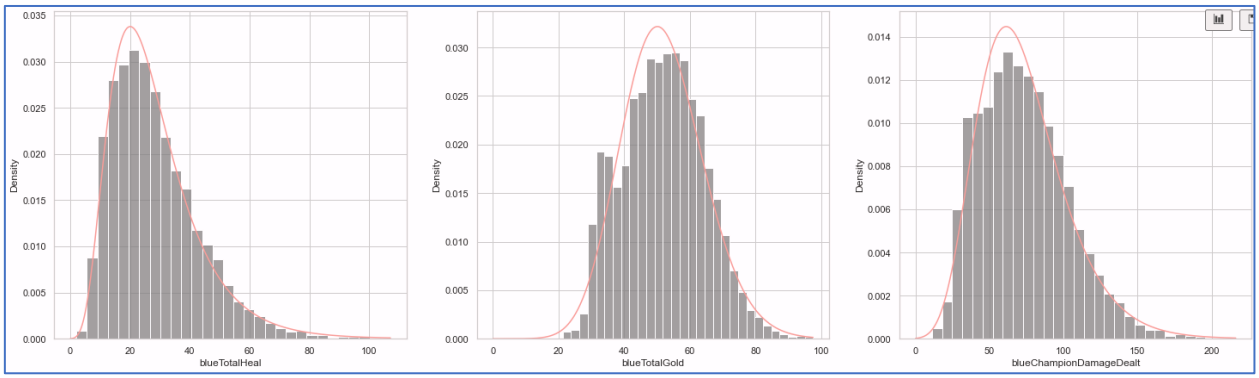
In this lab, 12 features will be considered, 3 of those are target variables and 9 are predictors (including 1 classificator).

From Lab 1: "Our dataset is game statistics data from the League of Legends game for 2020 from rated games in the "challanger" rank. The dataset is built using Riot.API (open public API for various in-game parameters from online games from Riot Games). The dataset contains many statistical parameters of past matches, including damage done, in-game currency earned, data on victories and defeats, etc. More details can be found in the README.MD file in the datasets folder."

2. Sampling of chosen target variables using univariate parametric distributions (from practice #2) with 2 different sampling methods.

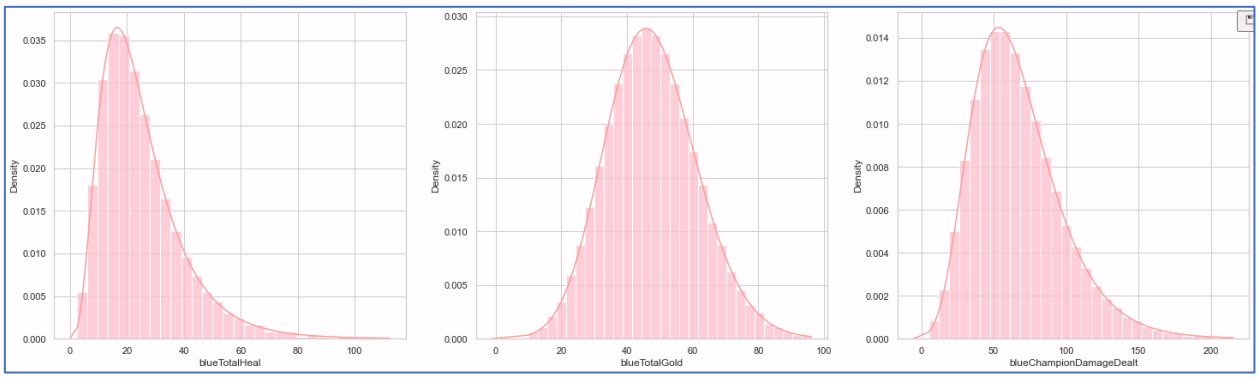
Based on the classes, we chose Inverse Transform and Accept-Reject methods to sample our target variables. The results are as follows:

1) Base data distribution:



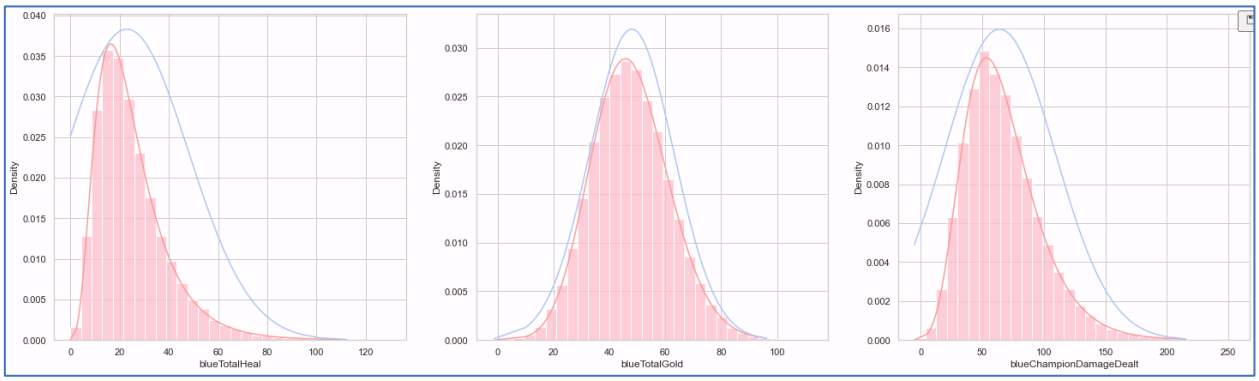
Pic.2. Base data distribution.

2) Inverse Transform sampling:



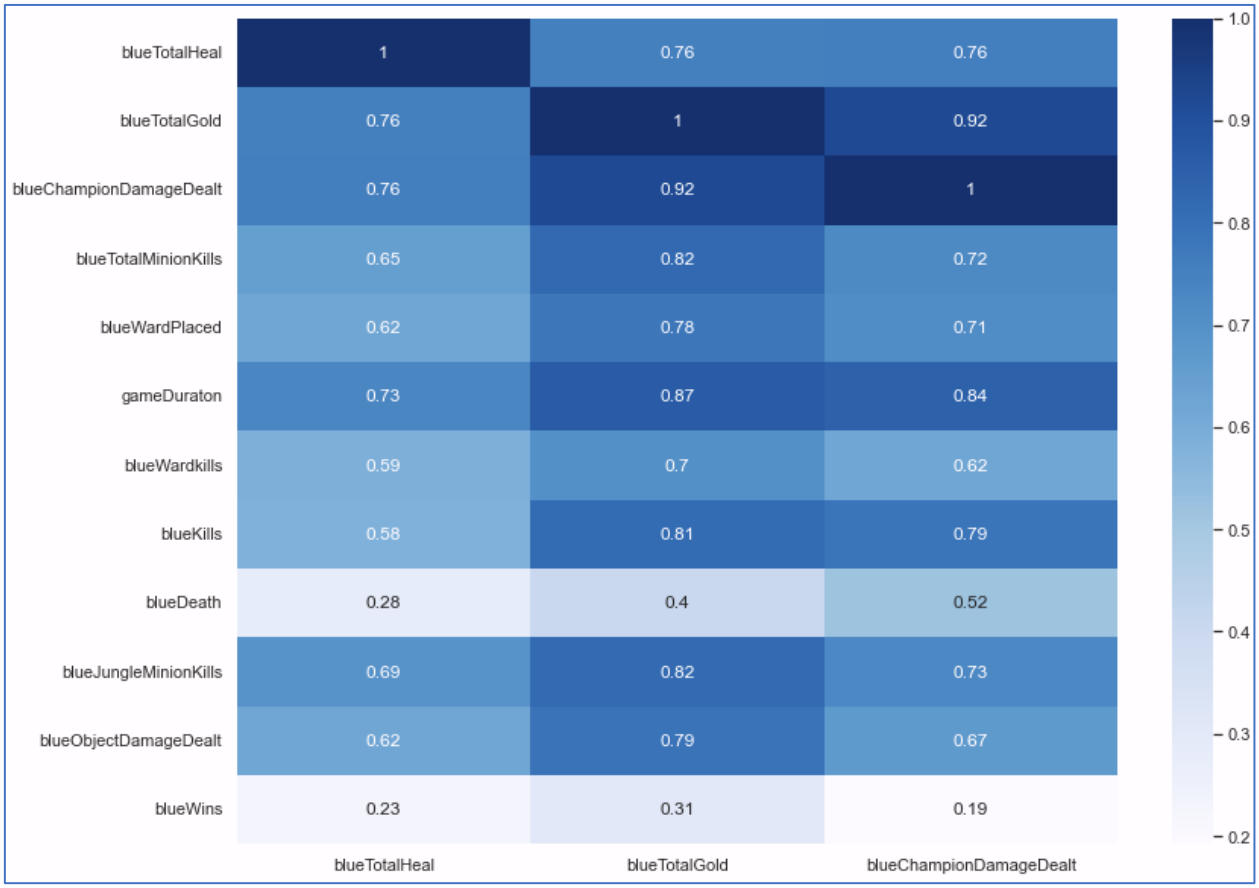
Pic.3. Inverse Transform sampling.

3) Accept-Reject sampling:



Pic.4. Accept-Reject sampling.

3. Estimation of relations between predictors and chosen target variables.



Pic.5. Correlation between predictors and target variables.

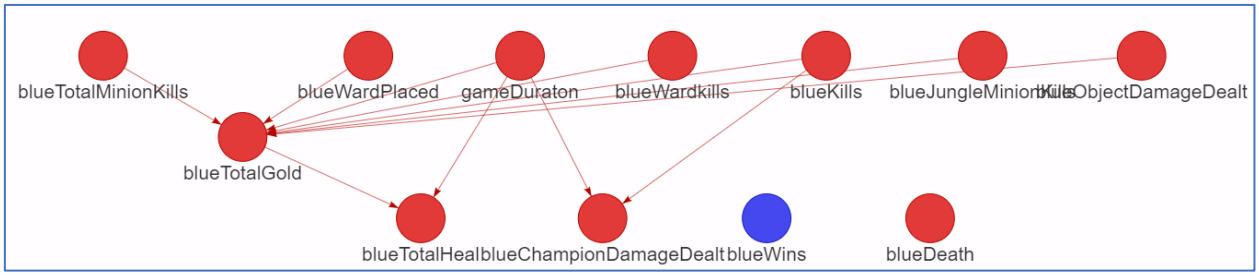
4. Bayesian network

For the Bayesian networks, the BAMT library (introduced to us in classes) was used. By the results of the correlation matrix, some edges were manually created:

```
['blueObjectDamageDealt', 'blueTotalGold'],
['blueJungleMinionKills', 'blueTotalGold'],
['blueTotalMinionKills', 'blueTotalGold'],
['blueWardkills', 'blueTotalGold'],
['blueWardPlaced', 'blueTotalGold'],
['gameDuraton', 'blueTotalGold'],
['gameDuraton', 'blueChampionDamageDealt'],
['blueKills', 'blueTotalGold'],
['blueTotalGold', 'blueTotalHeal'],
['gameDuraton', 'blueTotalHeal'],
['blueKills', 'blueChampionDamageDealt'],
```

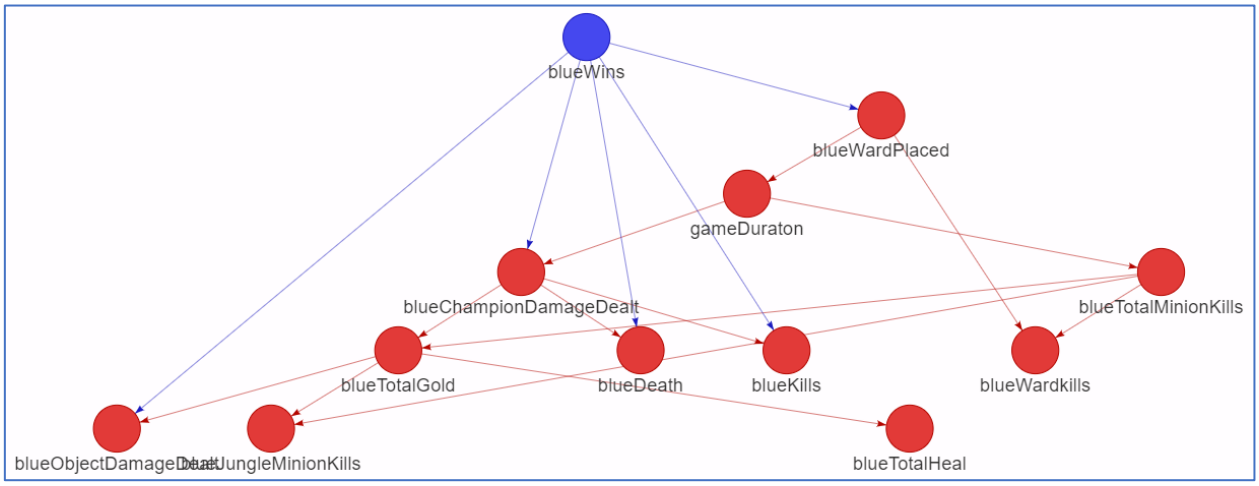
Pic.6. Manually created edges..

1) As a result of manually creating edges, the following graph appeared:



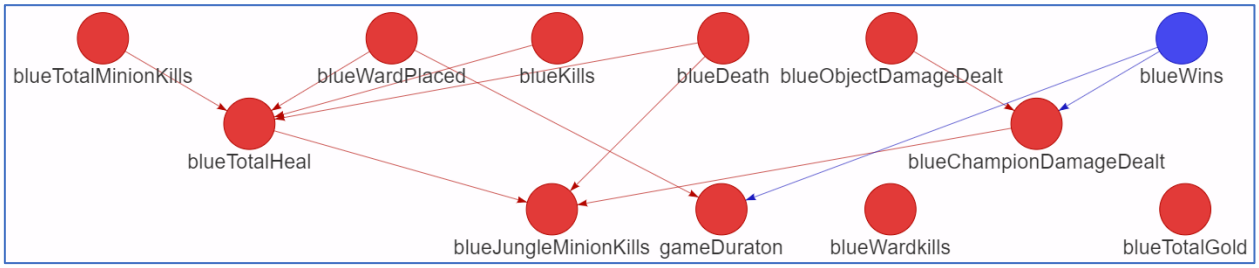
Pic.7. Manually created graph.

2) By using the Hill Climbing algorithm, another graph was structured:



Pic.8. Hill Climbing algorithm graph.

3) Last but not least, the Evolution algorithm creation:

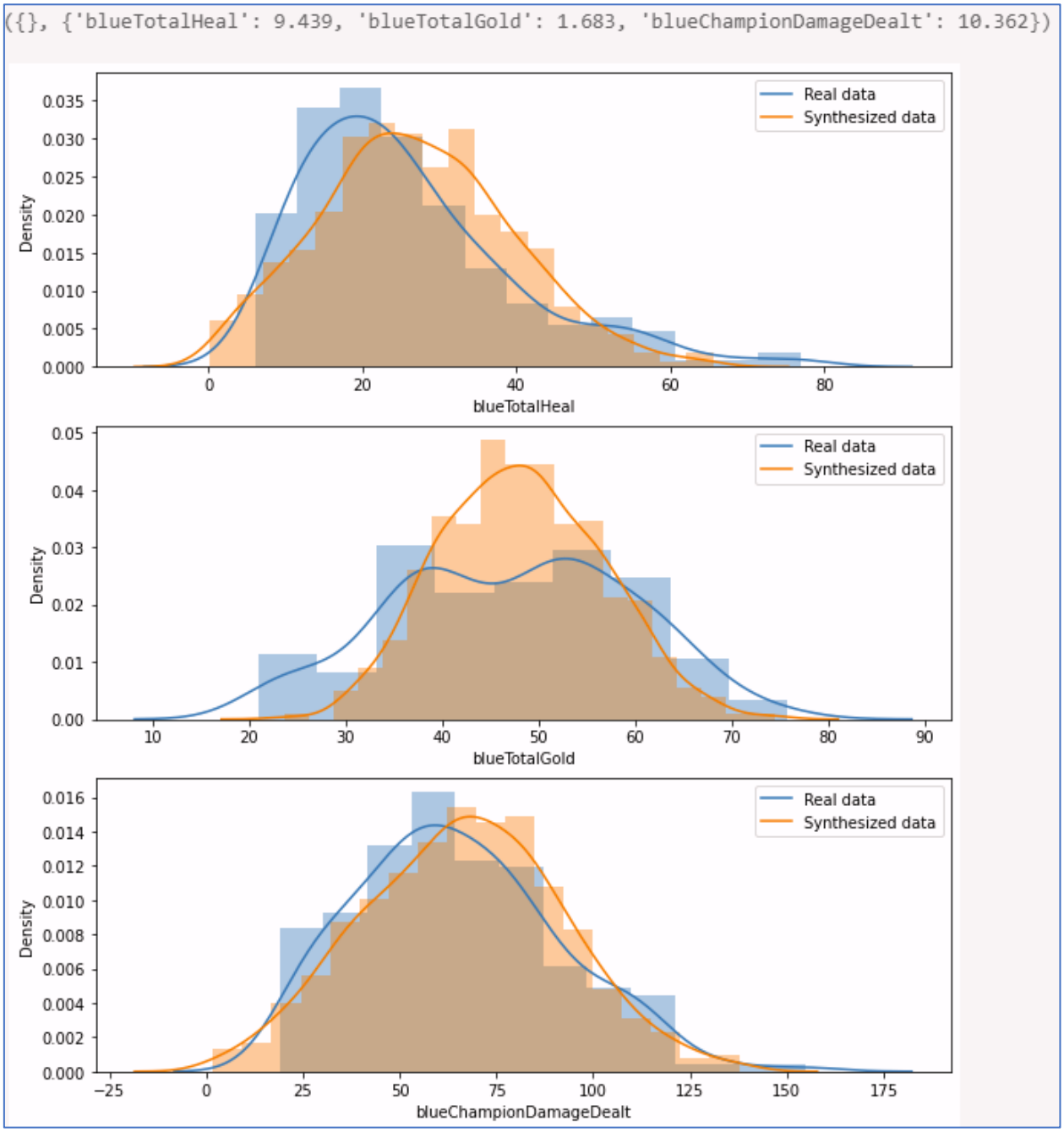


Pic.9. Evolution algorithm graph.

5. Quality analysis.

In order to qualitatively analyze the networks, the data was split to 80% training data and 20% test parts. This way synthesized data was generated and compared to the test data. The difference is described in the mean squared error (MSE) metric.

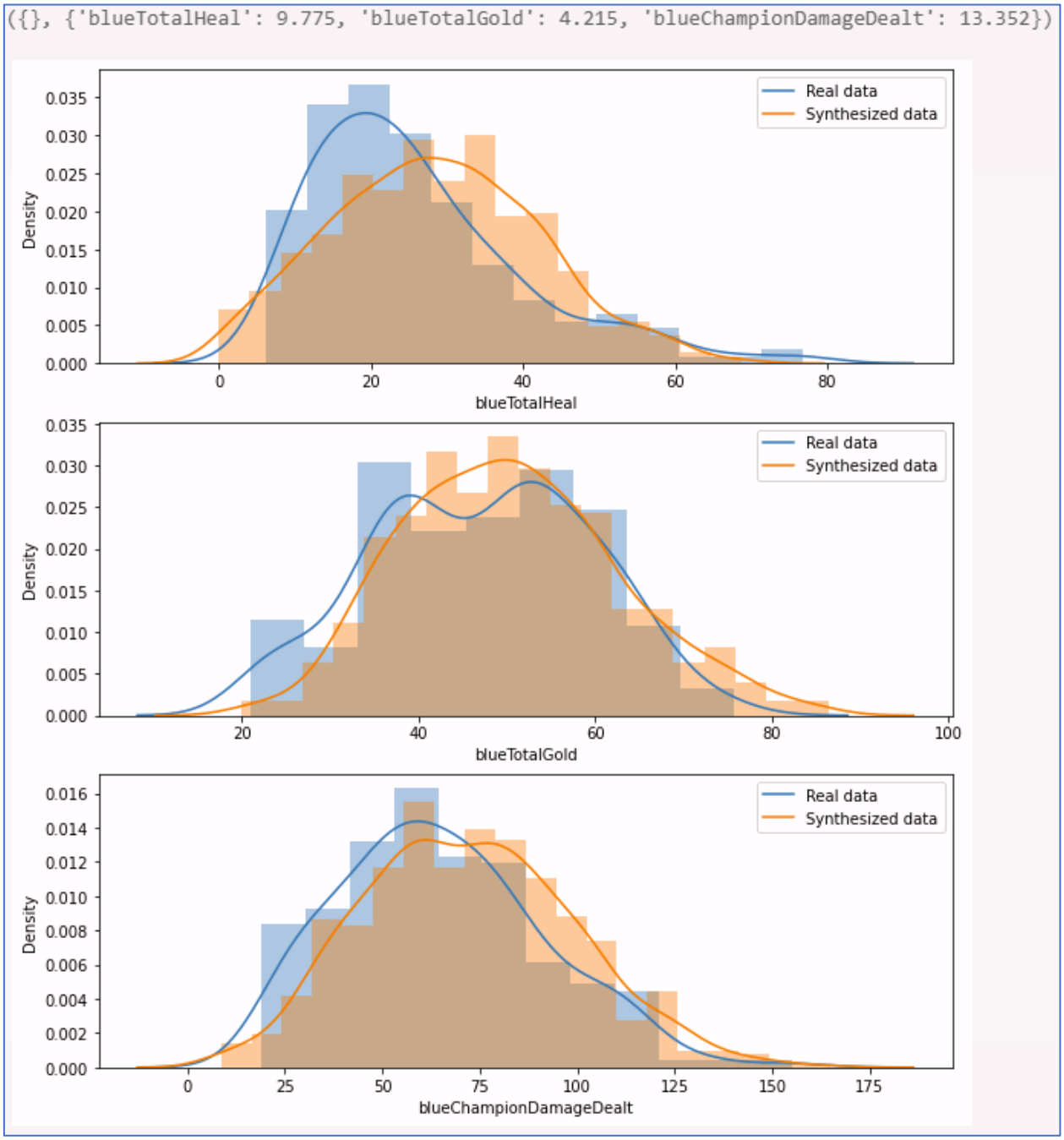
1) Manually created network:



Pic.10. Manually created network.

MSE: blueTotalHeal: 9.439, blueTotalGold: 1.683, blueChampionTotalDamageDealt: 10.362.

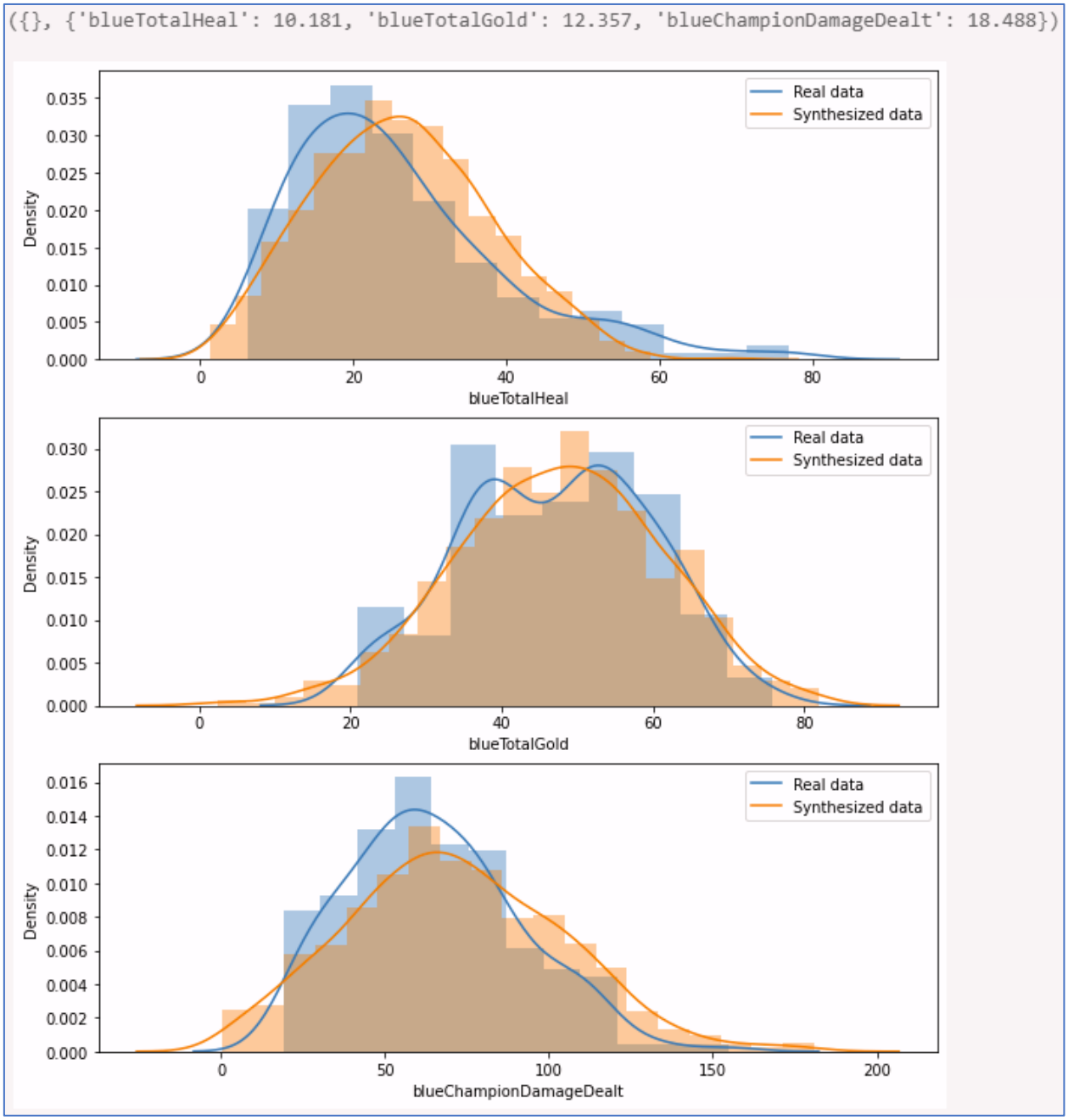
2) Hill Climbing algorithm network:



Pic.11. Hill Climbing algorithm network.

MSE: blueTotalHeal: 9.775, blueTotalGold: 4.215, blueChampionTotalDamageDealt: 13.352.

3) Evolution algorithm network:



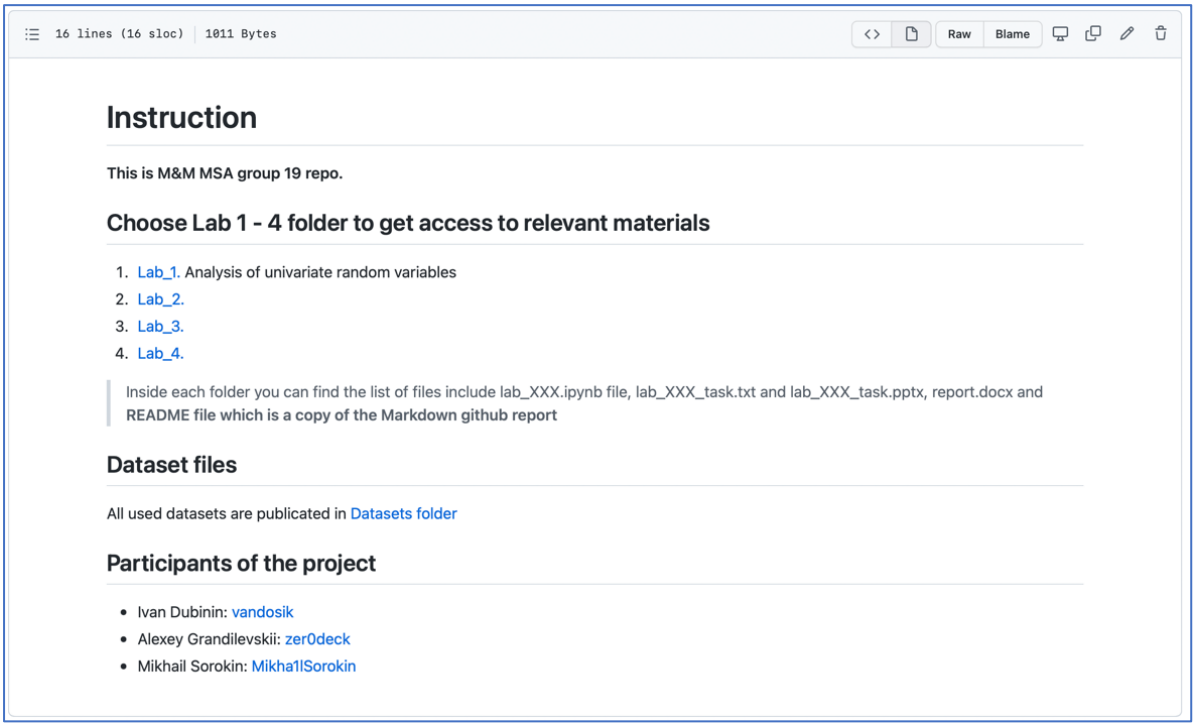
Pic.12. Evolution algorithm network.

MSE: blueTotalHeal: 10.181, blueTotalGold: 12.357, blueChampionTotalDamageDealt: 18.488.

Sourcecode:

- The full repository with all the labs: https://github.com/vandosik/M-M-MSA
- The repo with Datasets and additional used Data info: https://github.com/vandosik/M-M-MSA/tree/master/Datasets
- The Lab 3 ipynb file (steps 1-3): https://github.com/vandosik/M-M-MSA/blob/master/Lab_3/lab_3.ipynb
- The Lab 3 ipynb file (steps 4-6, BAMT used): https://github.com/vandosik/M-M-MSA/blob/master/Lab_3/lab_3_bayes.ipynb

We recommend using the first link because our GitHub project has README file with similar links and instructions which is really easy to use.



Pic.13. Our README.md file.