Technical sciences

УДК 004.85, 004.89

Potip Yuliia
Student of the
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
Потіп Юлія Сергіївна
студентка
Національного технічного університету України
«Київський політехнічний інститут імені Ігоря Сікорського»
Потип Юлия Сергеевна
студентка
Национального технического университета Украины
«Киевский политехнический институт имени Игоря Сикорского»

Kysliak Serhii
Senior Lecturer
National Technical University of Ukraine
«Igor Sikorsky Kyiv Polytechnic Institute»
Кисляк Сергій Володимирович
старший викладач
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Кисляк Сергей Владимирович
старший преподаватель
Национальный технический университет Украины
«Киевский политехнический институт имени Игоря Сикорского»

PROTEIN SEQUENCES CLASSIFICATION BY MACHINE LEARNING METHODS
КЛАСИФІКАЦІЯ БІЛКОВИХ ПОСЛІДОВНОСТЕЙ МЕТОДАМИ МАШИННОГО НАВЧАННЯ
КЛАССИФИКАЦИЯ БЕЛКОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

Summary. *The peculiarity of modern development of computational molecular biology is the exponential accumulation of biological data, which require detailed study and analysis. There is a variety of data mining techniques that can be used to classify biological data, but not all of them provide accurate prediction results and require special processing of biological sequences. The quality and speed of the protein classification result depends on the number of sequences presented in each class, the processing and transformation of these sequences, and the specificity of the machine learning algorithm. Newest sequencing methods are emerging, increasing the number of proteins, which leads to the problem of annotation. Big data is a big expense of computing power, contributing to the latest decisions on the classification of protein sequences. After all, classified protein is a step towards a narrower comparison of sequences and the solution of one of the most difficult tasks of bioinformatics.*

Key words: *k-nearest-neighbor, logistic regression, decision tree, gradient boosting, random forest.*

Анотація. *Особливістю сучасного розвитку обчислювальної молекулярної біології є експоненційне накопичення біологічних даних, що потребують детального вивчення та аналізу. Існує велика кількість різноманітних методів інтелектуального аналізу даних, що можуть бути застосовані для класифікації біологічних даних, але не всі вони дають точний*

результат прогнозу та потребують особливої обробки біологічних послідовностей. Якість та швидкість результату класифікації білків залежать від кількості послідовностей представлених у кожному класі, обробки і трансформації цих послідовностей та від специфіки обраного алгоритму машинного навчання. Враховуючи появу новітніх методів секвенування, зростає кількість білків, що призводить до проблеми анотації. Великі об'єми даних - великі витрати обчислювальних потужностей комп'ютера, що вимагають новітні рішення щодо класифікації білкових послідовностей. Адже, класифікований білок – це крок до більш вузького порівняння послідовностей та рішення однієї з найскладніших задач біоінформатики.

Ключові слова: *метод k-найближчих сусідів, логістична регресія, дерево рішень, градієнтне прискорення, випадкові дерева.*

Аннотация. *Особенностью современного развития вычислительной молекулярной биологии является экспоненциальное накопление биологических данных, которые требуют детального изучения и анализа. Существует большое количество разнообразных методов интеллектуального анализа данных, которые могут быть применены для классификации биологических данных, но не все они дают точный результат прогноза и требуют особенной обработки биологических последовательностей. Качество и скорость результата классификации белков зависит от количества последовательностей, представленных в каждом классе, обработки и трансформации этих последовательностей и от специфики избранных алгоритмов машинного обучения. Учитывая появление новых методов секвенирования, растет количество белков, что приводит к проблеме аннотации. Большие объемы данных - большие затраты вычислительных*

мощностей компьютера, требующие новейшие решения по классификации белковых последовательностей. Ведь, классифицированный белок - это шаг к более узкому сравнению последовательностей и решение одной из самых сложных задач биоинформатики.

Ключевые слова: *алгоритм k-ближайших соседей, логистическая регрессия, дерево решений, градиентное ускорение, случайные деревья.*

Introduction. Machine learning is gaining popularity in modern science. With the increasing amount of biological data new methods of analyzing them are emerging. With the advent of next-generation sequencing methods, the number of protein sequences is increasing at a high rate. The main unsolved problem of modern bioinformatics is the lag of the number of annotated proteins in comparison with the unannounced protein sequences (Fig.1). At the beginning of 2020, the Swiss-Prot database, that containing validated protein information, retains 561911 manually annotated sequences. The computer-annotated protein sequence database TrEMBL contains 177754527 sequences. There are various methods of annotating protein sequences [1; 2; 3], the main ones being alignment algorithms [4; 5; 6], which do not allow solving the main problem of bioinformatics.

Classification of protein sequences is a complex task that involves the analysis, processing and transformation of biological data, using statistical and analytical tools [7]. As demonstrated by studies [8; 9; 10], various machine learning algorithms can be used to classify protein sequences, which allow to achieve more than 93% prediction accuracy. Machine learning models are able to accelerate the process of annotating biological sequences by identifying a protein class of unknown sequence, resulting in a narrower range of proteins for further comparison. The point of the work is to create an optimal binary classification model for three groups of proteins: oxidoreductase, transferase, and hydrolase, using five machine learning algorithms:

k-nearest neighbor method, Logistic Regression, Decision Tree, Gradient Boosting, and Random Forest.

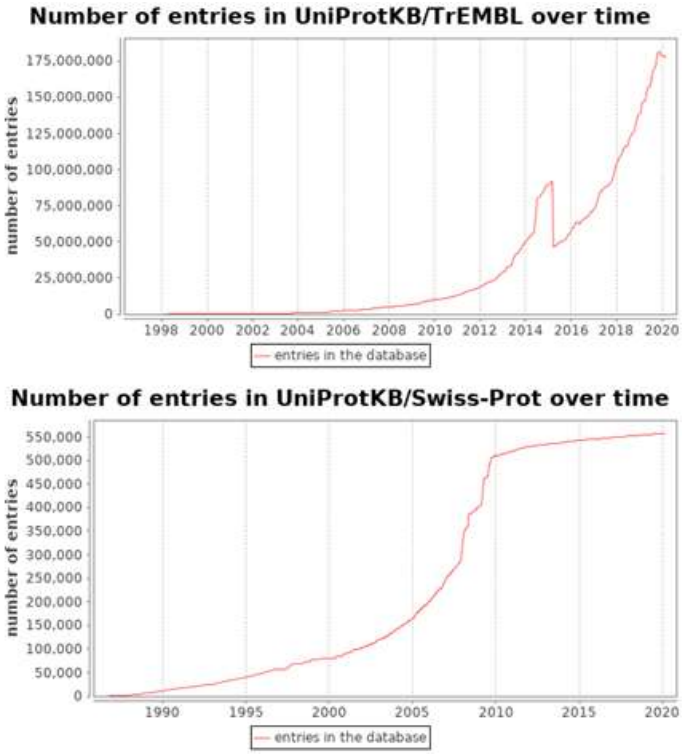

Fig. 1.Number of annotated protein sequences in Swiss-Prot and TrEMBL databases

Materials and Methods. The study used machine learning methods in Python programming language, using a tool for interactive development and visualization of projects in the field of data science - Jupyter Notebook.

The k-nearest neighbor method (KNN) is one of the simplest and at the same time efficient algorithms [11]. By calculating the distances to each amino acid sequence, the k-nearest neighbors are chosen, to which the distances are shortest and they are allocated to a separate class.

Logistic Regression (LR) is an algorithm that can be used for binary classification. The result of Logistic Regression is an estimate of dependent variable in the range from 0 to 1. The prediction of a protein class occurs by setting a threshold that indicates the separation between the two classes [12; 13].

The Decision Tree is an algorithm with a tree structure which main idea is the recursive selection of attributes - amino acid sequences [14]. At the beginning of the algorithm, the root is the aggregate dataset, the branch is the rule according to which the decision was made, and the leaf is the result - the corresponding protein class.

The Gradient Boosting method is a technology that creates a forecast model in the form of an ensemble of weak models represented as decision trees [15]. By updating forecasts so that the amount of balances is minimal and predicted values are close to actual ones, the technology achieves the best results.

The Random Forest method is an ensemble of a large number of decision trees [16]. Each individual tree generates a class prediction. As a result, the class with the highest number of votes becomes the model's prediction.

In general, each algorithm has its advantages and disadvantages, which arise depending on the task and data. It is important to note that a large number of scientists prefer the support vector machine algorithm (SVM) [17; 18; 19; 20]. The SVM method has high precision, even with a small amount of data, but it requires extremely large computing resources. Whereas with large amounts of data, other algorithms demonstrate learning speed and prediction accuracy [21].

It is advisable to visualize the result of the algorithm using the ROC-curve, and the quality is estimated as the area under this curve AUC (Area Under ROC Curve) [22; 23; 24].

There are various indicators that are used to evaluate a classifier model, such as: "accuracy", "precision", "recall", and "f1 weighted" [25; 26]. These metrics estimate the accuracy of the model, but are calculated differently. "Accuracy" is calculated using the indicator function, "precision" and "recall" take into account the ratio of the number of responses, "f1 weighted"-metric is calculated according to the values of "precision" and "recall".

A database of protein sequences was used in computational experiments, classified according to Enzyme Commission numbers approved by the International Union of Biochemistry and Molecular Biology, it was obtained from the Research Collaboratory for Structural Bioinformatics (RCSB) of Protein Data Bank(PDB) [27; 28]. The three largest groups of proteins were selected: oxidoreductase, transferase and hydrolase. Five models were created for each group. The database of the three groups of proteins contains 117081 sequences. In the oxidoreductase class are presented 34321 sequences, in the transferase class are presented 36424 sequences, in the hydrolase class 46336 sequences are presented (Fig.2).

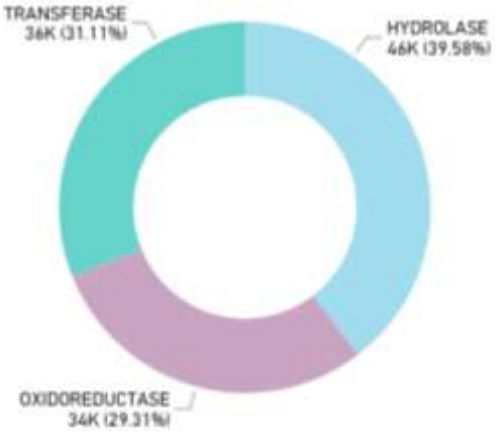

Fig. 2. The ratio of the number of sequences in each group

The following libraries were used to process and visualize the data: "Pandas" (http://pandas.pydata.org/) [29], "NumPy" (https://numpy.org/) [30], "Matplotlib" (https: //matplotlib.org/) [31]; to transform data and create models "Scikit-learn" (http://scikit-learn.org/) [32] in Python programming languages.

Results. Five machine learning models were developed for each protein group: oxidoreductase, transferase, hydrolase, and model accuracy metrics were obtained (Tables 1-3) and graphs of algorithms in the form of a ROC-curve can be seen in Fig. 3-11.

Table 1

Algorithm accuracy metrics for the hydrolase classification models

algorithm name	fit_time	test_accuracy	test_f1_weighted	test_precision_macro	test_recall_macro	test_roc_auc
LR	68.09939	0.98399	0.98397	0.98394	0.9826	0.99830
RandomForest	82.91798	0.97926	0.97923	0.97951	0.9771	0.99699
KNN	0.75713	0.91446	0.91535	0.91027	0.9277	0.98094
DecisionTree	557.63519	0.97658	0.97657	0.97556	0.9755	0.97694
GradientBoosting	354.70491	0.74514	0.70897	0.83938	0.6799	0.85859

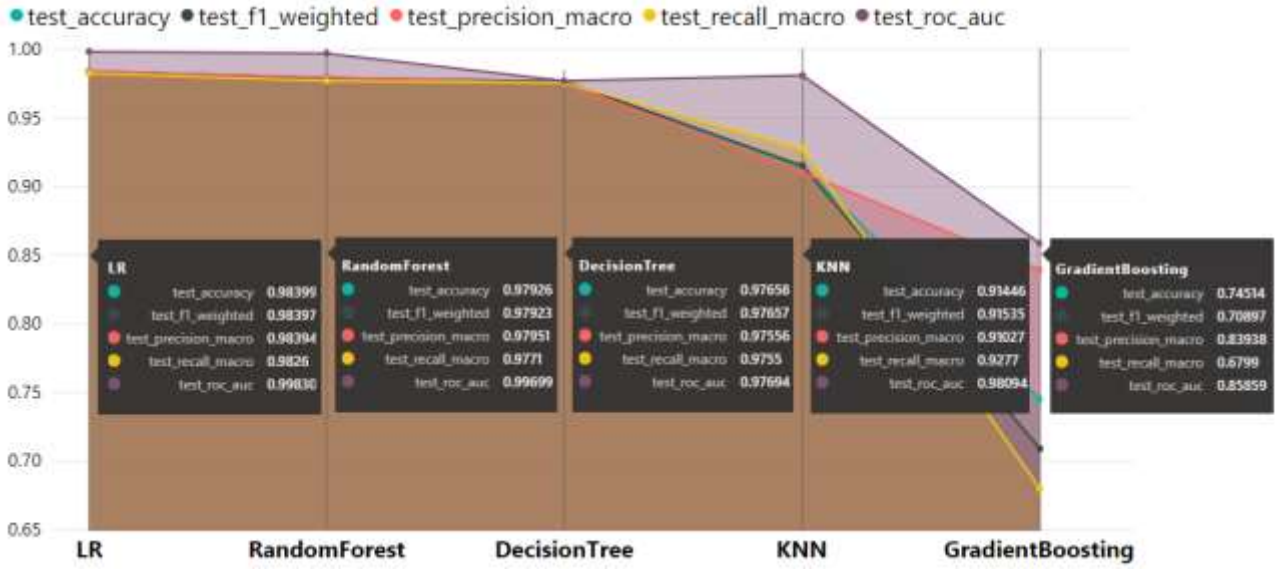

Fig. 3. Ratio of "accuracy", "f1 weighted", "precision", "recall", "roc-auc" metrics for hydrolase classification models

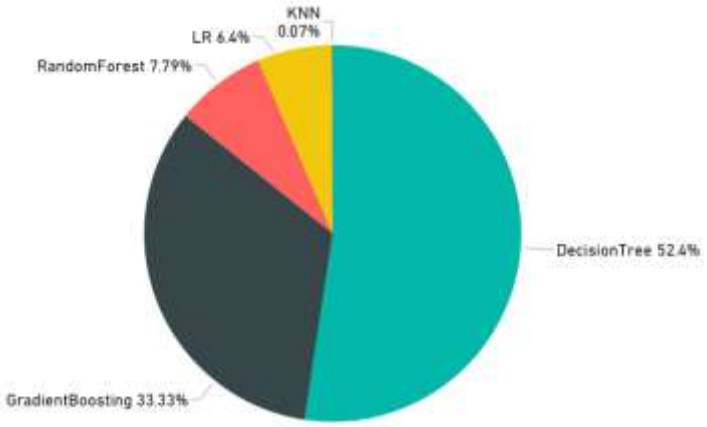

Fig. 4. Ratio of the "fit time" metric for hydrolase classification models

Table 2

Algorithm accuracy metrics for the transferase classification models

algorithm name	fit_time	test_accuracy	test_f1_weighted	test_precision_macro	test_recall_macro	test_roc_auc
LR	71.66131	0.98586	0.98583	0.98596	0.98105	0.99820
RandomForest	96.58021	0.97971	0.97963	0.98035	0.97227	0.99565
KNN	0.76972	0.93536	0.93326	0.95399	0.89827	0.97263
DecisionTree	516.12035	0.96980	0.96982	0.96402	0.96577	0.96717
GradientBoosting	354.04959	0.77736	0.73382	0.85925	0.64644	0.84345

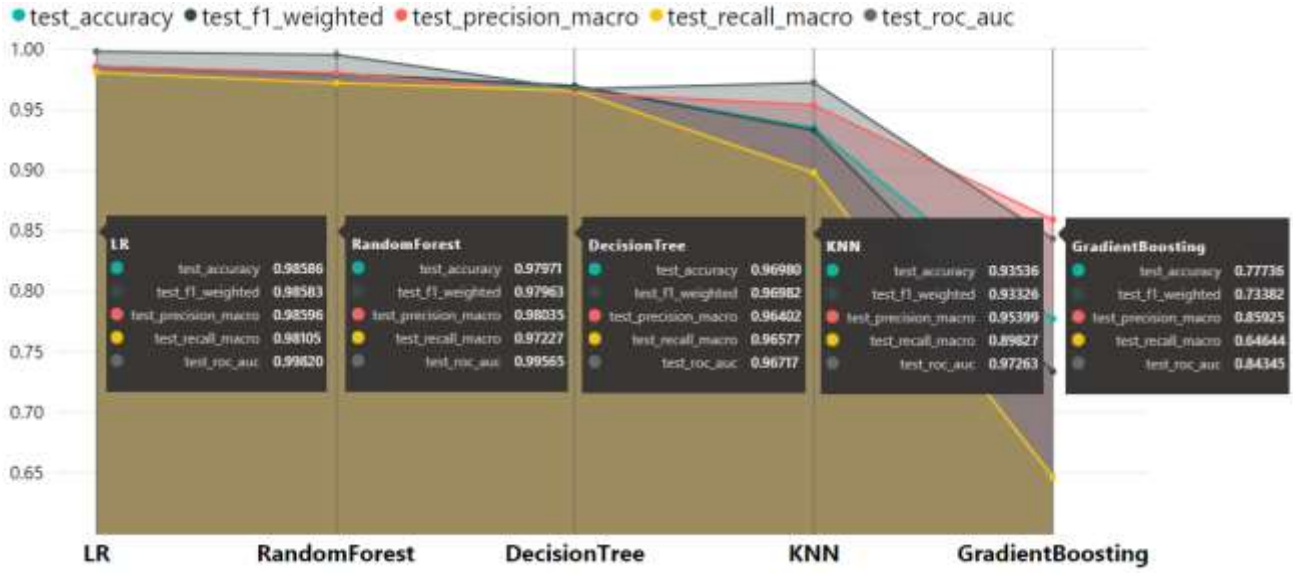

Fig. 5. Ratio of "accuracy", "f1 weighted", "precision", "recall", "roc-auc" metrics for transferase classification models

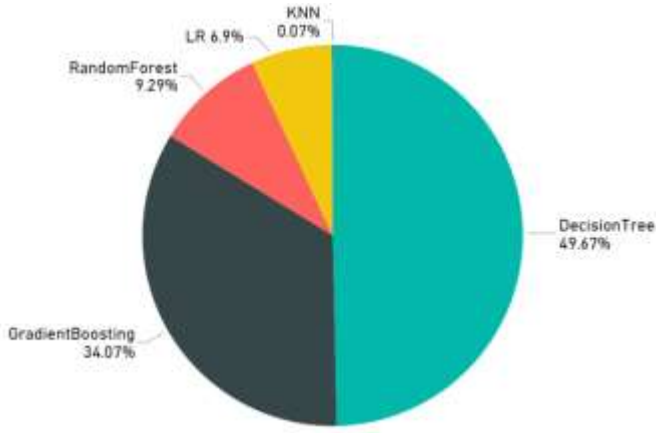

Fig. 6. Ratio of the "fit time" metric for transferase classification models

Table 3

Algorithm accuracy metrics for the oxidoreductase classification models

algorithm name	fit_time	test_accuracy	test_f1_weighted	test_precision_macro	test_recall_macro	test_roc_auc
LR	54.43252	0.99128	0.99126	0.9910	0.98784	0.99911
RandomForest	60.05371	0.98862	0.98858	0.9899	0.98260	0.99719
DecisionTree	489.25117	0.98492	0.98495	0.9805	0.98319	0.98408
KNN	0.79592	0.96043	0.95964	0.9707	0.93443	0.98255
GradientBoosting	386.10834	0.80160	0.76364	0.8839	0.66182	0.86355

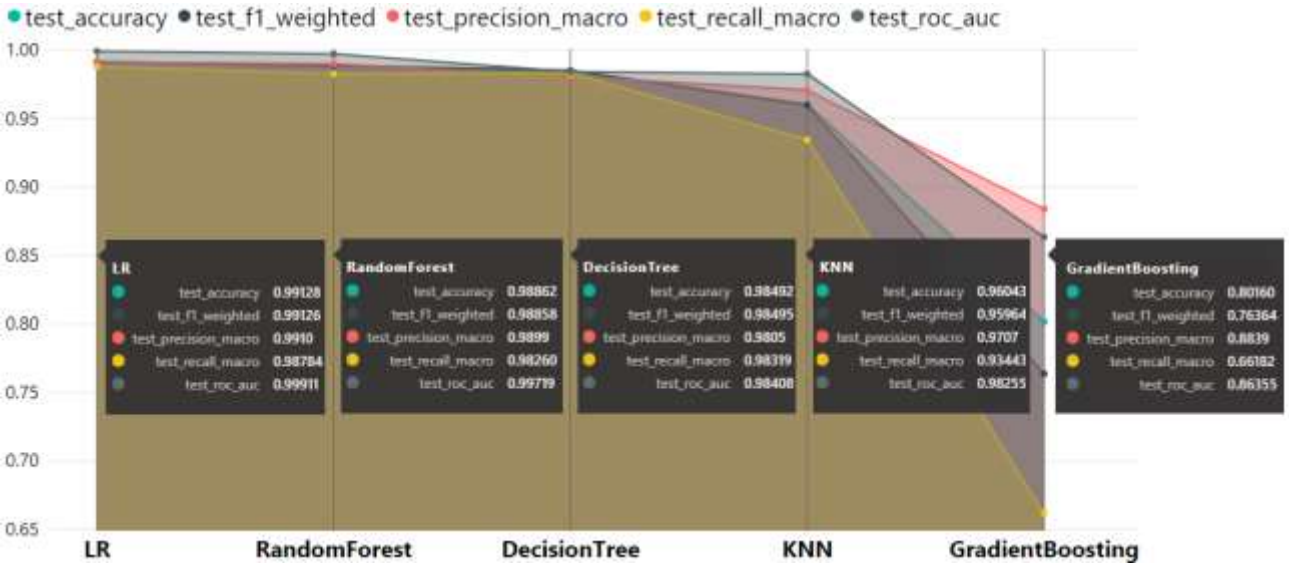

Fig. 7. Ratio of "accuracy", "f1 weighted", "precision", "recall", "roc-auc" metrics for oxidoreductase classification models

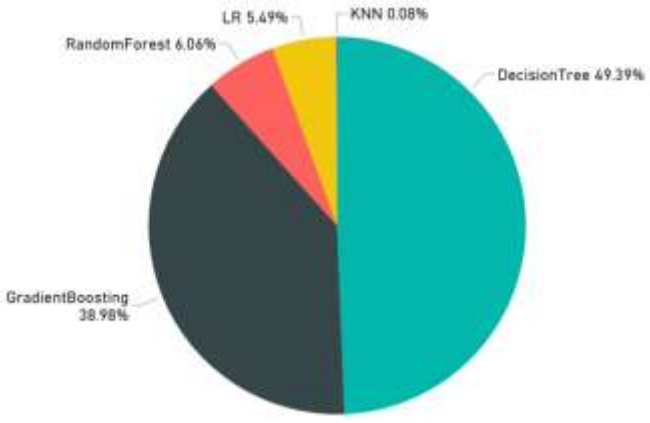

Fig. 8. Ratio of the "fit time" metric for oxidoreductase classification models

Note that in (Fig. 9-11) the "x" axis are displayed false positive decisions (FPR – False Positive Rate), and on the "y" axis are true positive decisions (TPR - True Positive Rate).

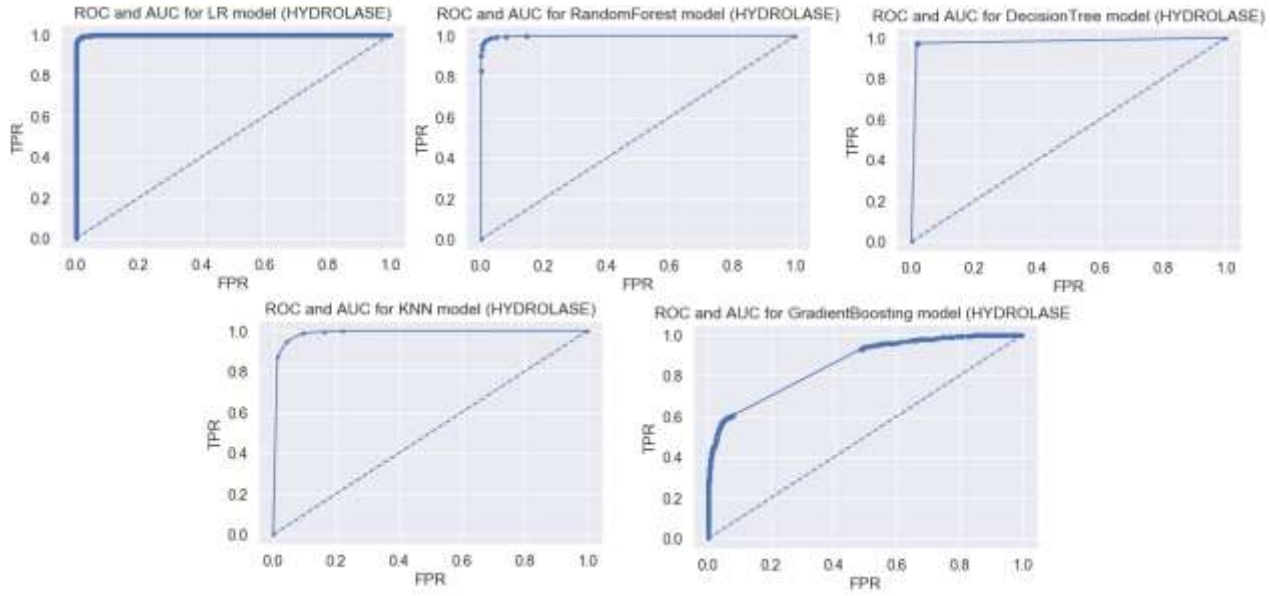

Fig. 9. Graphs of the ROC-curve of the algorithms for hydrolase classification models

Figure 9 shows that for the hydrolase classification the best result is demonstrated by the Logistic Regression and the Random Forest algorithm. The quantitative interpretation of the ROC-curve is the area under this curve (AUC). AUC has the highest values for these algorithms. The worst result is demonstrated by the Gradient Boosting algorithm, for which the area under the curve has the lowest value.

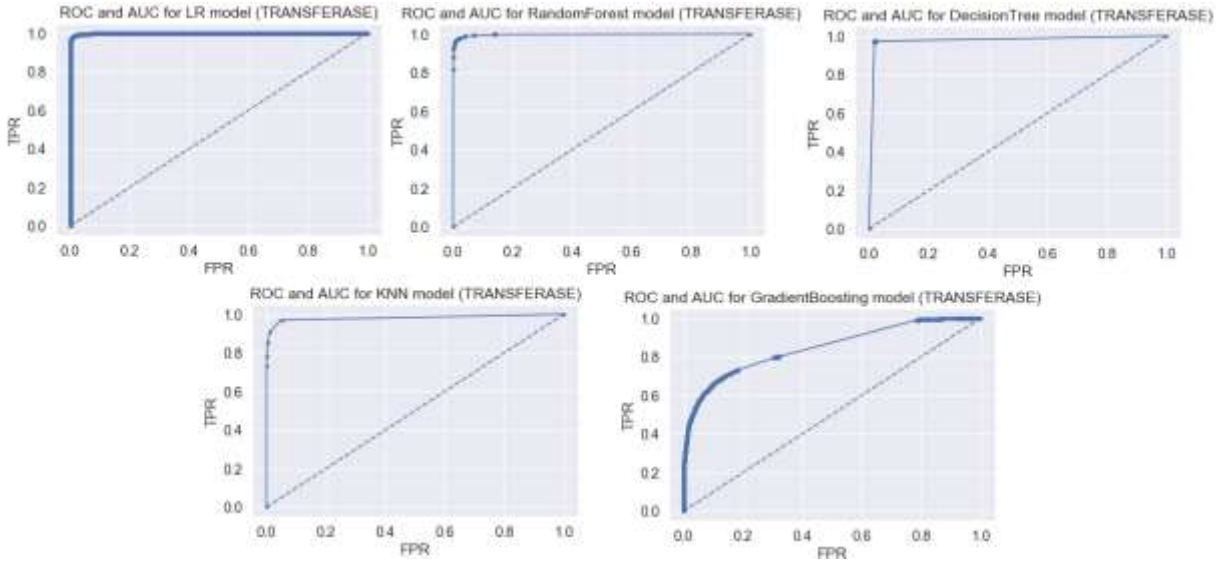

Fig. 10. Graphs of the ROC-curve of the algorithms for transferase classification models

Figure 10 shows that for transferase classification the best result is demonstrated by the Logistic Regression and the Random Forest algorithm, for which the area under the curve (AUC) has the highest values. The worst result is demonstrated by the Gradient Boosting algorithm, for which the area under the curve has the lowest value.

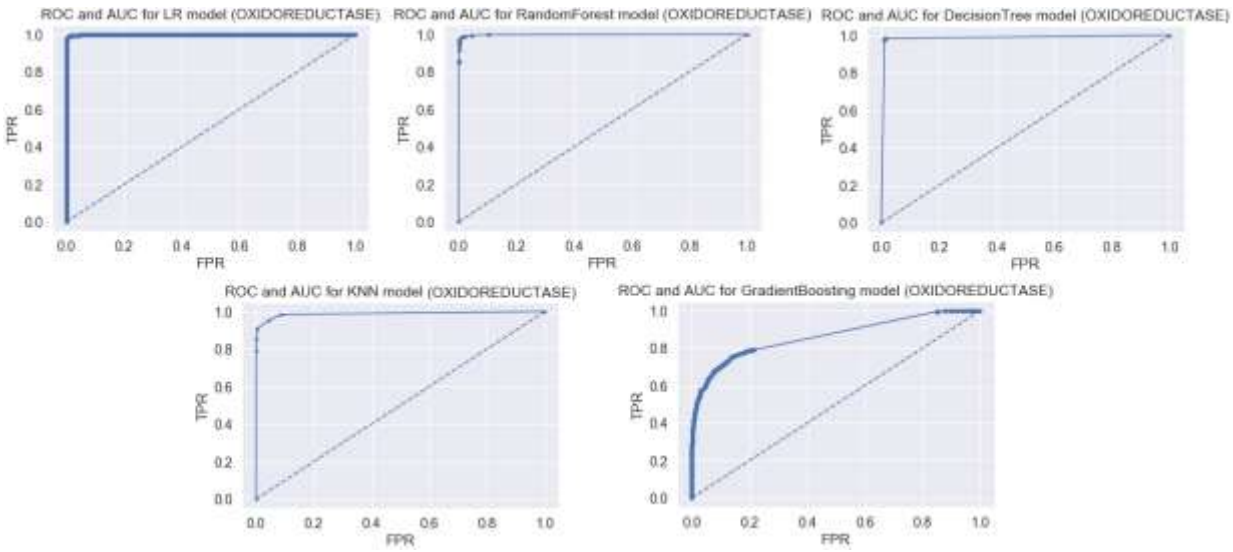

Fig. 11. Graphs of the ROC-curve of the algorithms for oxidoreductase classification models

Figure 11 shows that for the oxidoreductase classification also the best result is demonstrated by the Logistic Regression and the Random Forest algorithm, the worst result is demonstrated by Gradient Boosting algorithm.

Discussion. According to the results in Table 1 and Figures 1,2,7 for the hydrolase classification, it can be seen that according to the "fit time" metric, the k-nearest neighbors algorithm (KNN) has the lowest value. This algorithm demonstrates the maximum speed of operation. Other metrics for accuracy of the algorithm, such as: "accuracy", "precision", "recall", "f1 weighted", "roc-auc", are calculated as follows [32]:

$$\textbf{accuracy} = \frac{1}{n_{samples}} \sum_{i=1}^{n_{samples}-1} 1(y`_i = y_i),$$

where $y`_i$ is predicted protein class; $y_i$ is corresponding true protein class $1(y`_i = y_i)$ is indicator function;

$$\textbf{precision} = \frac{TP}{TP + FP},$$

where TP is a number of true positive values; FP – is a number of false positives values;

$$\textbf{recall} = \frac{TP}{TP + FN},$$

where TP is a number of true positive values; FN is a number of false negative values;

$$\textbf{f1 weighted} = 2 * (\frac{precision * recall}{precision + recall}),$$

where precision – value of metric "precision"; recall – value of metric "recall";

roc-auc or values AUC =

$$\int_{x=0}^{1} TPR(FPR^{-1}(x))dx = \int_{+\infty}^{-\infty} TPR(T)FPR`(T)dT =$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(T`>T) f_1(T`) f_0(T)dT`dT = P(X_1 > X_0)$$

where $TPR = \frac{TP}{TP+FN}$; $FPR = \frac{FP}{FP+TN}$, T is variable threshold; X_1 is rating for a positive copy; X_0 is rating for a negative copy; f_1, f_0 are probability densities;

In contrast to the fast running k-nearest-neighbor algorithm, it can be seen that Decision Tree and Gradient Boosting algorithms are fitting more than 6 times longer than all algorithms on the test data. However, these algorithms do not show the best accuracy values, according to other metrics "accuracy", "precision", "recall", "f1 weighted" "roc-auc".

Logistic Regression and Random Forest algorithms do not have large fitting time and have enough high result of accuracy. Note that the Logistic Regression algorithm operates a little faster and according to the metrics "accuracy", "precision", "recall", "f1 weighted" has a prediction accuracy higher by 0.01 and according to "roc-auc" - by 0.002 (Table 1).

Similar results of the accuracy of the algorithms can be seen for the transferase and oxidoreductase classification models (Table 2.3, Figure 3-6, 8, 9). It should be noted the short fitting time of the k-nearest neighbor algorithm and long fitting time of the Decision Tree and Gradient Boosting algorithms.

The Logistic Regression algorithm shows the best result of prediction accuracy and the optimal fitting time of data for classification of individual groups of proteins: hydrolase, transferase and oxidoreductase in comparison with other algorithms. Comparing the Logistic Regression algorithm for the classification of hydrolase, transferase and oxidoreductase, we can see that the fitting time of the algorithm for the hydrolase classification model is "68.099", for transferase - "71.661", for

oxidoreductase - "54.432", while the average of all accuracy metrics reach "0.986" for hydrolase, "0.988" for transferase and "0.992" for oxidoreductase. The oxidoreductase classification model using the Logistic Regression algorithm is the most accurate and optimal for fitting time.

Conclusions. Analyzing k-nearest neighbors, Logistic Regression, Random Forest, Decision Tree and Gradient Boosting machine learning algorithms for the problem of hydrolase, transferase and oxidoreductase classification, we can make the following conclusions: the k-nearest-neighbor algorithm has the lowest fitting time; the Decision Tree algorithm has the highest fitting time; the Gradient Boosting algorithm has large fitting time and the worst predictive accuracy; Random Forest algorithm has little fitting time and medium-high prediction accuracy; the Logistic Regression algorithm has small fitting time and the best accuracy for individual binary classification models of hydrolase, transferase, and oxidoreductase.

References

1. Mahlich Y, Steinegger M, Rost B, Bromberg Y. HFSP: high speed homology-driven function annotation of proteins. Bioinformatics. 2018;34(13):i304-i312.
2. Armean I, Lilley K, Trotter M, Pilkington N, Holden S. Co-complex protein membership evaluation using Maximum Entropy on GO ontology and InterPro annotation. Bioinformatics. 2018; 34(11):1884-1892.
3. Makrodimitris S, van Ham R, Reinders M. Improving protein function prediction using protein sequence and GO-term similarities. Bioinformatics. 2018; 35(7):1116–1124.
4. Dijkstra M, van der Ploeg A, Feenstra K, Fokkink W, Abeln S, Heringa J. Tailor-made multiple sequence alignments using the PRALINE 2 alignment toolkit. Bioinformatics. 2019; 35(24):5315-5317.

5. Zhang C, Zheng W, Mortuza S, Li Y, Zhang Y. DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. Bioinformatics. 2019; 36(7):2105-2112.
6. Makigaki S, Ishida T. Sequence alignment using machine learning for accurate template-based protein structure prediction. Bioinformatics. 2019; 36(1):104-111.
7. Tan J, Lv H, Wang F, Dao F, Chen W, Ding H. A Survey for Predicting Enzyme Family Classes Using Machine Learning Methods. Current Drug Targets. 2019; 20(5):540-550.
8. De Ferrari L, Mitchell J. From sequence to enzyme mechanism using multi-label machine learning. BMC Bioinformatics. 2014; 15(1).
9. Wang Y, Jing R, Hua Y, Fu Y, Dai X, Huang L et al. Classification of multi-family enzymes by multi-label machine learning and sequence-based descriptors. Analytical Methods. 2014; 6(17):6832.
10. Lin J, Chen H, Li S, Liu Y, Li X, Yu B. Accurate prediction of potential druggable proteins based on genetic algorithm and Bagging-SVM ensemble classifier. Artificial Intelligence in Medicine. 2019; 98:35-47.
11. Beretta L, Santaniello A. Nearest neighbor imputation algorithms: a critical evaluation. BMC Medical Informatics and Decision Making. 2016; 16(S3).
12. Niinimäki T, Heikkilä M, Honkela A, Kaski S. Representation transfer for differentially private drug sensitivity prediction. Bioinformatics. 2019; 35(14):i218-i224.
13. Vangaveti S, Vreven T, Zhang Y, Weng Z. Integrating ab initio and template-based algorithms for protein–protein complex structure prediction. Bioinformatics. 2019; 36:751–757.

14. Wang P, Tu Y, Tseng Y. PgpRules: a decision tree based prediction server for P-glycoprotein substrates and inhibitors. Bioinformatics. 2019; 35(21):4535-4535.
15. Zhang Y, Yu S, Xie R, Li J, Leier A, Marquez-Lago T et al. PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. Bioinformatics. 2019; 36(3):704–712.
16. Fabris F, Doherty A, Palmer D, de Magalhães J, Freitas A. A new approach for interpreting Random Forest models and its application to the biology of ageing. Bioinformatics. 2018; 34(14):2449-2456.
17. Cai C. SVM-Prot: web-based support vector machine software for functional classification of a protein from its primary sequence. Nucleic Acids Research. 2003; 31(13):3692-3697.
18. Cao Z, Pan X, Yang Y, Huang Y, Shen H. The lncLocator: a subcellular localization predictor for long non-coding RNAs based on a stacked ensemble classifier. Bioinformatics. 2018; 34(13):2185-2194.
19. Wei L, Luan S, Nagai L, Su R, Zou Q. Exploring sequence-based features for the improved prediction of DNA N4-methylcytosine sites in multiple species. Bioinformatics. 2018; 35(8):1326-1333.
20. Iqbal S, Hoque M. PBRpredict-Suite: a suite of models to predict peptide-recognition domain residues from protein sequence. Bioinformatics. 2018; 34(19):3289-3299.
21. Hardware Acceleration of SVM classifier using Zynq SoC FPGA. International Journal of Innovative Technology and Exploring Engineering. 2019; 8(12):2280-2288.
22. Pagel K, Pejaver V, Lin G, Nam H, Mort M, Cooper D et al. When loss-of-function is loss of function: assessing mutational signatures and impact of loss-of-function genetic variants. Bioinformatics. 2017; 33(14):i389-i398.

23. Saito T, Rehmsmeier M. Precrec: fast and accurate precision–recall and ROC curve calculations in R. Bioinformatics. 2016; 33(1):145-147.
24. Bian Y, He C, Hou J, Cheng J, Qiu J. PairedFB: a full hierarchical Bayesian model for paired RNA-seq data with heterogeneous treatment effects. Bioinformatics. 2018; 35(5):787-797.
25. Saha S, Johnson J, Pal S, Weinstock G, Rajasekaran S. MSC: a metagenomic sequence classification algorithm. Bioinformatics. 2019; 35(17):2932-2940.
26. Mineeva O, Rojas-Carulla M, Ley R, Schölkopf B, Youngblut N. DeepMAsED: evaluating the quality of metagenomic assemblies. Bioinformatics. 2020.
27. Matsuta Y, Ito M, Tohsato Y. ECOH: An Enzyme Commission number predictor using mutual information and a support vector machine. Bioinformatics. 2012; 29(3):365-372.
28. Quinn G, Bi C, Christie C, Pang K, Prli A, Nakane T et al. RCSB PDB Mobile: iOS and Android mobile apps to provide data access and visualization to the RCSB Protein Data Bank. Bioinformatics. 2014; 31(1):126-127.
29. Cario C, Witte J. Orchid: a novel management, annotation and machine learning framework for analyzing cancer mutations. Bioinformatics. 2017; 34(6):936-942.
30. McKinney W. Pandas: a foundational python library for data analysis and statistics. Python for High Performance and Scientific Computing [Internet]. 2020. URL: https://www.researchgate.net/publication/265194455_pandas_a_Foundational_Python_Library_for_Data_Analysis_and_Statistics
31. Hunter J. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. 2007; 9(3):90-95.

32. Virtanen P, Gommers R, Oliphant T, Haberland M, Reddy T, Cournapeau D et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods. 2020; 17(3):261-272.