Asfendiyarov Kazakh National Medical University

Тема исследования: Практическое применение Автоматизированной системы научных исследований в медицине, здравоохранении и смежных областях

Проект: Анализ факторов риска сердечно сосудистых заболеваний и прогноз исходов лечения при помощи методов Машинного Обучения

Раздел II. Моделирование и Анализ данных

Автор исследования: Dr. Alexander Wagner (Berlin)

Berlin-Almaty
2025

Содержание

Предисловие

Данная работа посвящена проблеме проведения научного исследования и создания научного отчета в медицине и здравоохранении при помощи Автоматизированной Системы Научных Исследований. Краткая характеристика системы приведена в данном разделе. Описание научного исследования, выполненое по установленным международным нормам, правилам и рекомендациям, приводится в последующих разделах данного документа.

Моделирование и Анализ данных

Для проведения моделирование и анализа данных нами использованы 18 моделей Машинного обучения. Ниже приведены выходные результаты, полученные в результате работы каждой модели.

Результатами являются:

- Таблица классификации
- Confusion Matrix
- ROC Curve
- Score plot

Модель: Linear Regression

Linear Regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression. Reference Wikipedia.

Таблица №15. Таблица классификации

Classes+Metrics	precision	recall	f1-score	support
class 0	0.822	0.817	0.82	164.0
class 1	0.853	0.857	0.855	203.0
accuracy	0.839	0.839	0.839	0.839
macro avg	0.838	0.837	0.837	367.0
weighted avg	0.839	0.839	0.839	367.0

График №46. Confusion Matrix

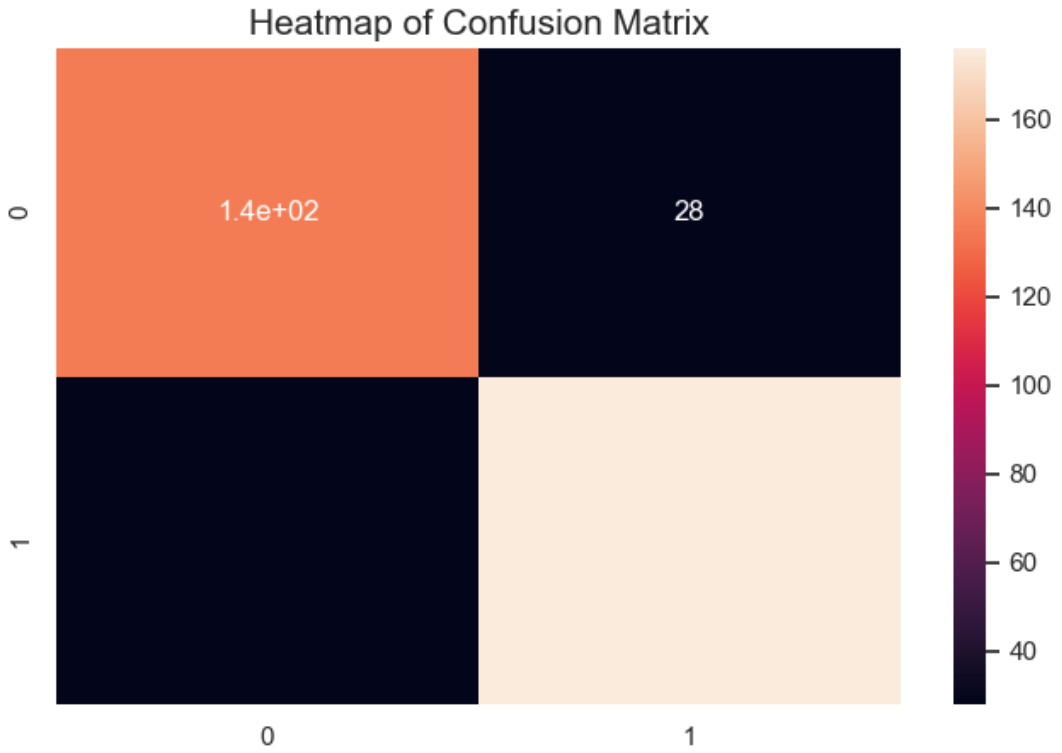


График №47. ROC Curve

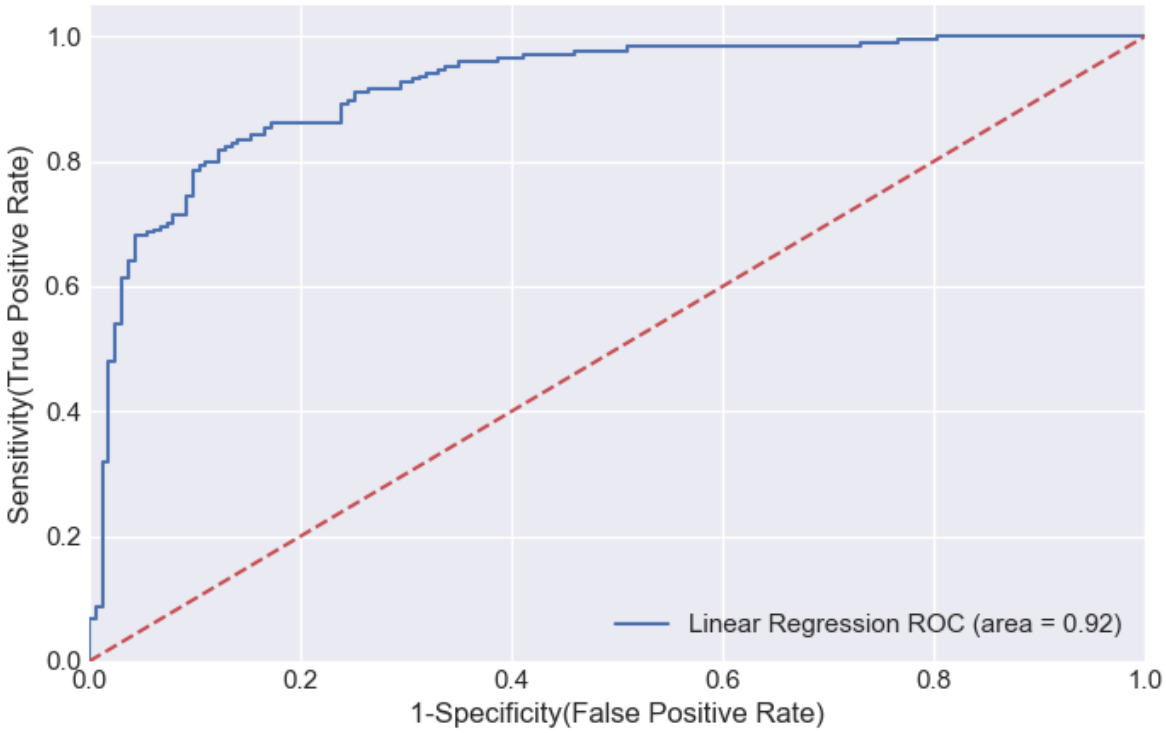
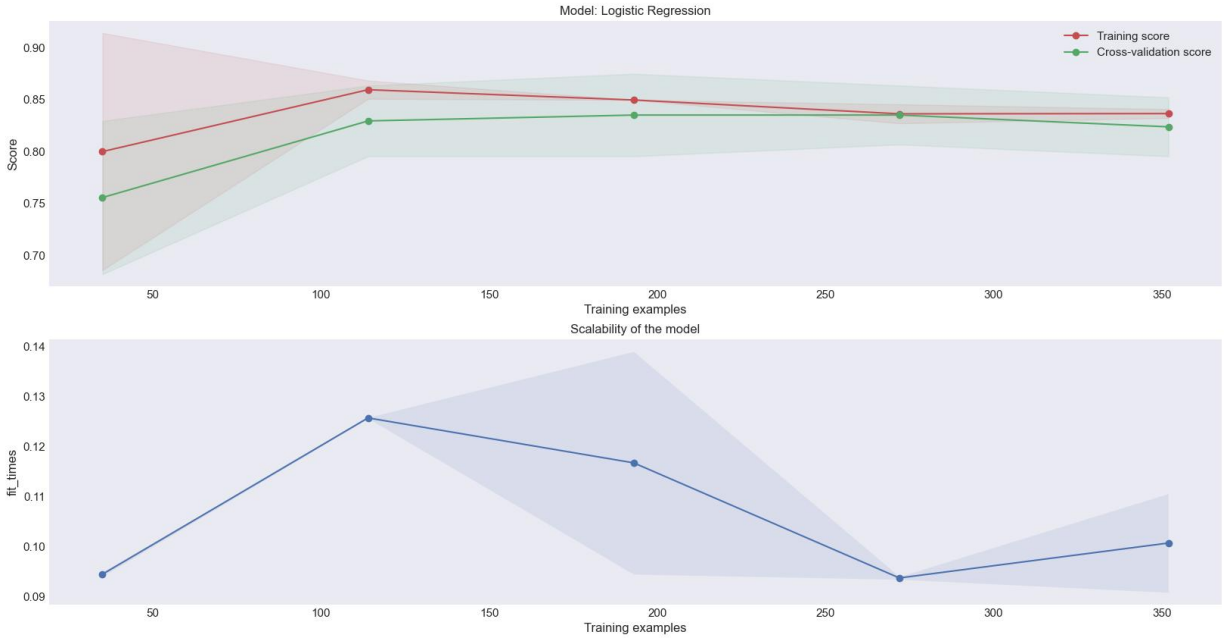


График №48. Score plot



Модель: Logistic Regression

Logistic Regression is a useful model to run early in the workflow. Logistic regression measures the relationship between the categorical dependent variable (feature) and one or more independent variables (features) by estimating probabilities using a logistic function, which is the cumulative logistic distribution. Reference Wikipedia.

Таблица №16. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.822	0.817	0.82	164.0
class 1	0.853	0.857	0.855	203.0
accuracy	0.839	0.839	0.839	0.839
macro avg	0.838	0.837	0.837	367.0
weighted avg	0.839	0.839	0.839	367.0

График №49. Confusion Matrix

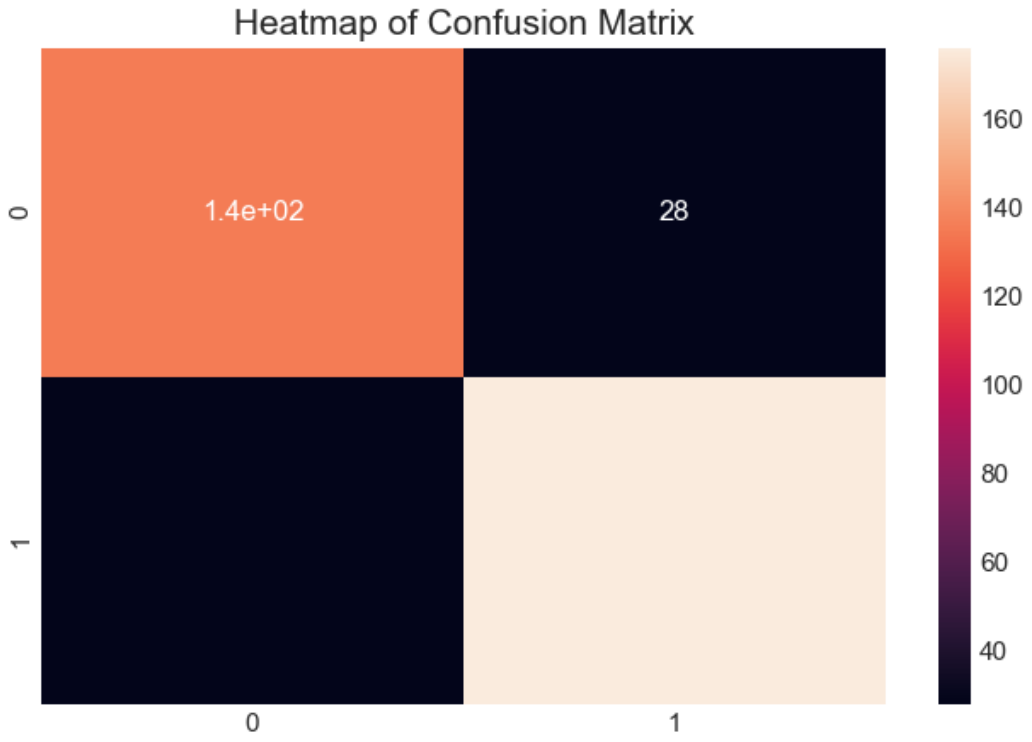


График №50. ROC Curve

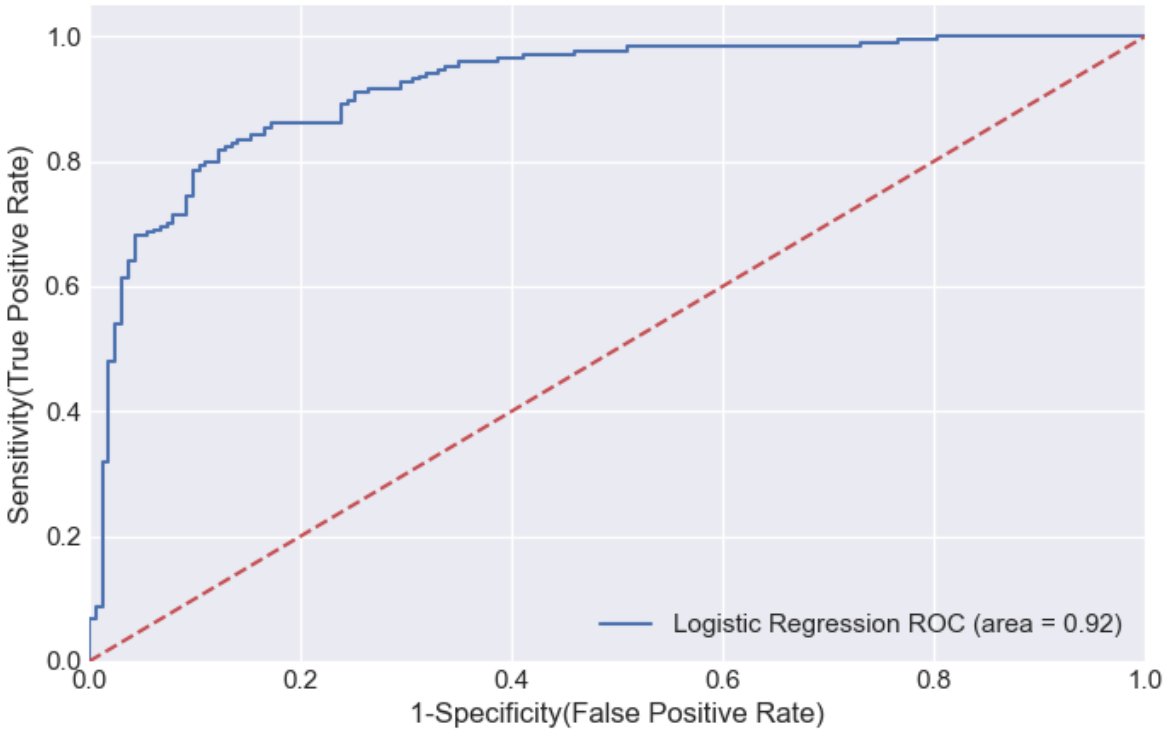
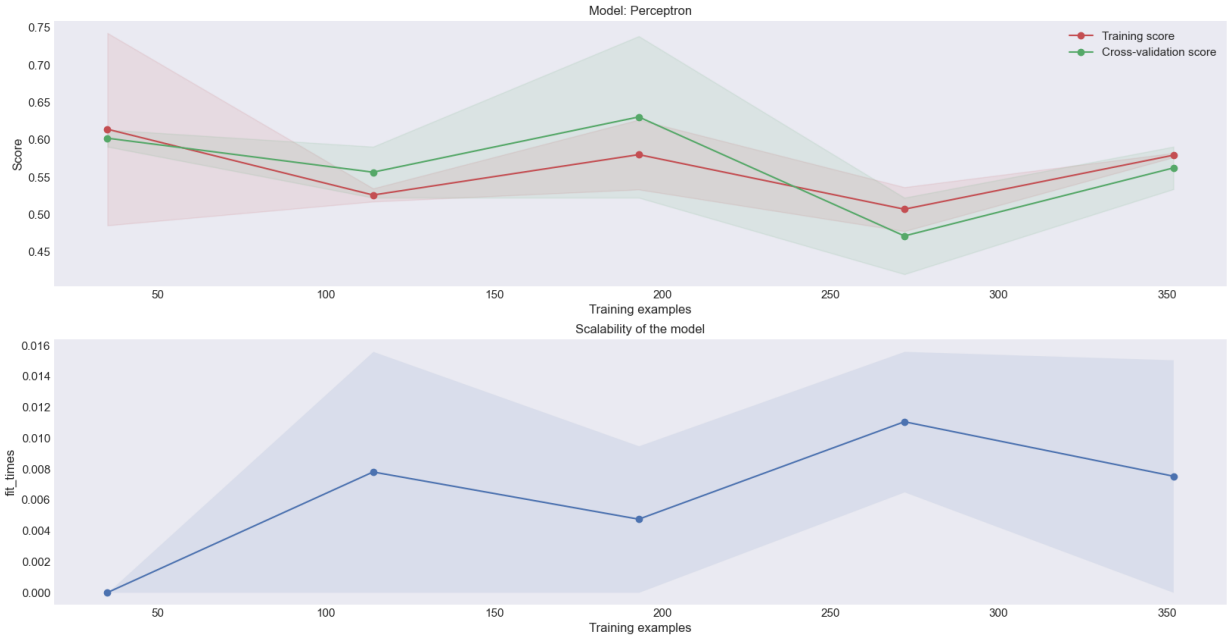


График №51. Score plot



Модель: Perceptron

Thanks to https://www.kaggle.com/startupsci/titanic-data-science-solutions The Perceptron is an algorithm for supervised learning of binary classifiers (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not). It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector. The algorithm allows for online learning, in that it processes elements in the training set one at a time. Reference Wikipedia.

Таблица №17. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.2	0.018	0.034	164.0
class 1	0.543	0.941	0.688	203.0
accuracy	0.529	0.529	0.529	0.529
macro avg	0.371	0.48	0.361	367.0
weighted avg	0.39	0.529	0.396	367.0

График №52. Confusion Matrix

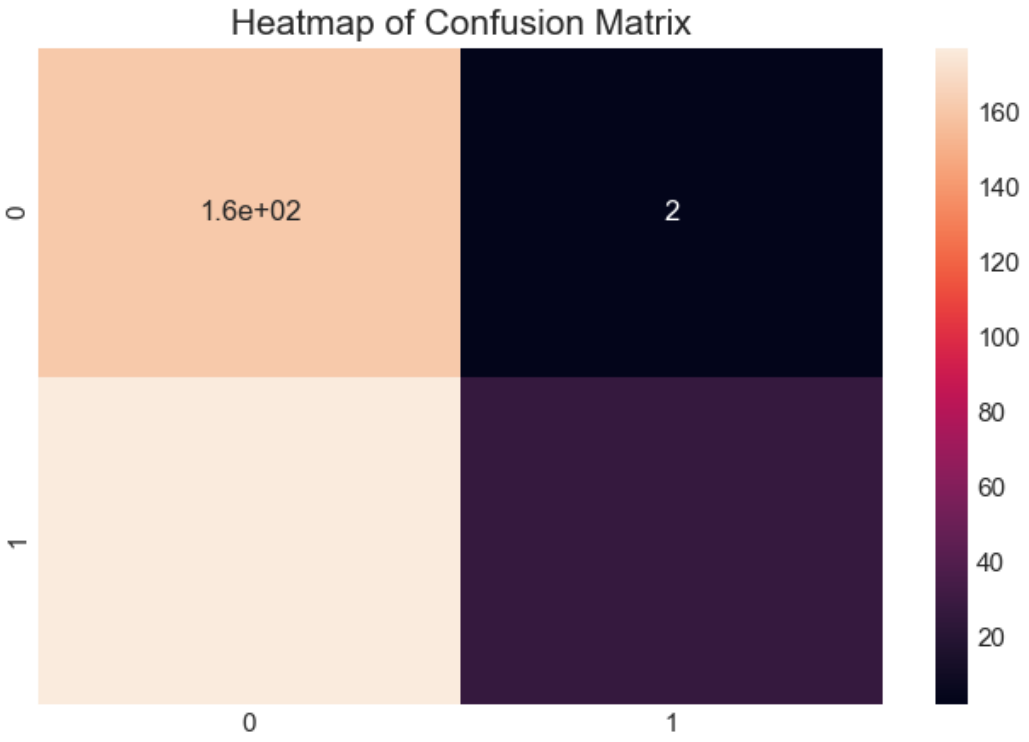


График №53. ROC Curve

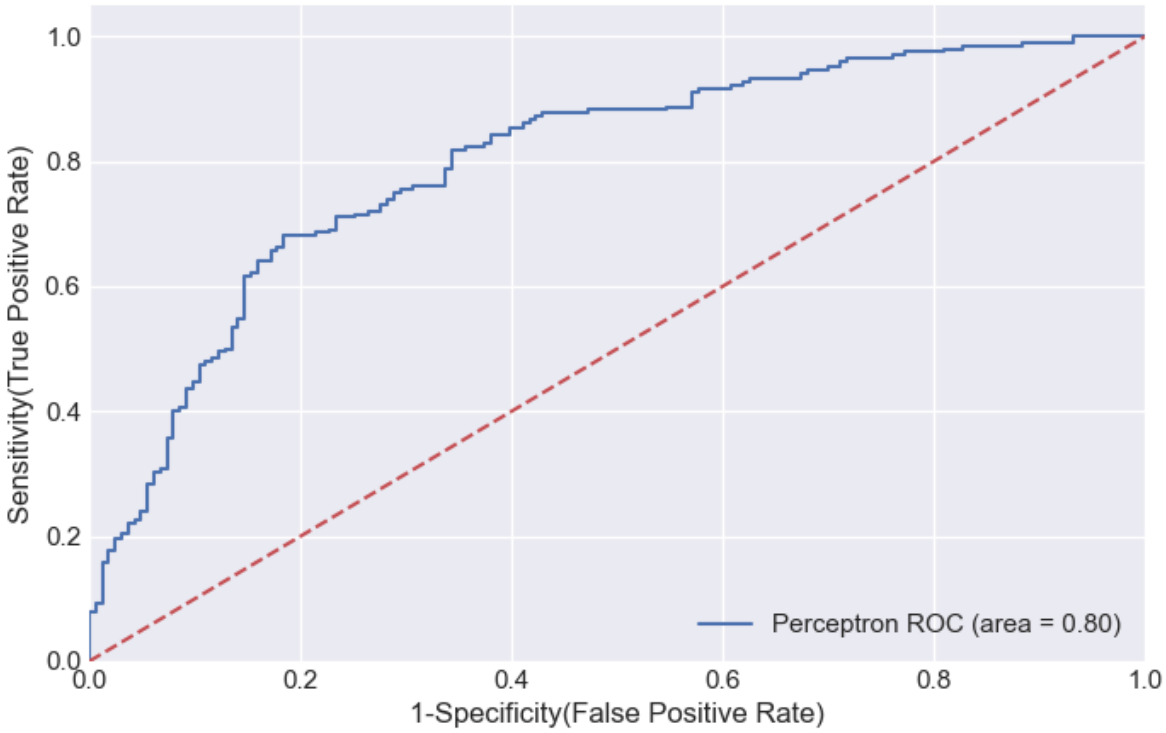
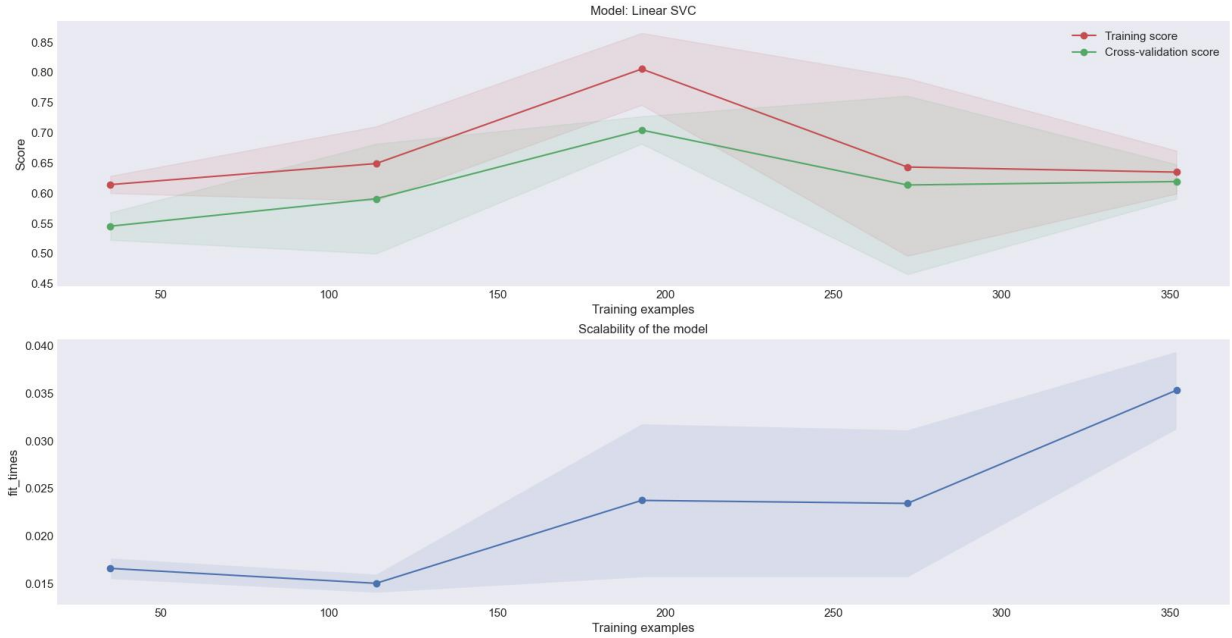


График №54. Score plot



Модель: Linear SVC

Linear SVC is a similar to SVM method. Its also builds on kernel functions but is appropriate for unsupervised learning. Reference Wikipedia.

Таблица №18. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.822	0.817	0.82	164.0
class 1	0.853	0.857	0.855	203.0
accuracy	0.839	0.839	0.839	0.839
macro avg	0.838	0.837	0.837	367.0
weighted avg	0.839	0.839	0.839	367.0

График №55. Confusion Matrix

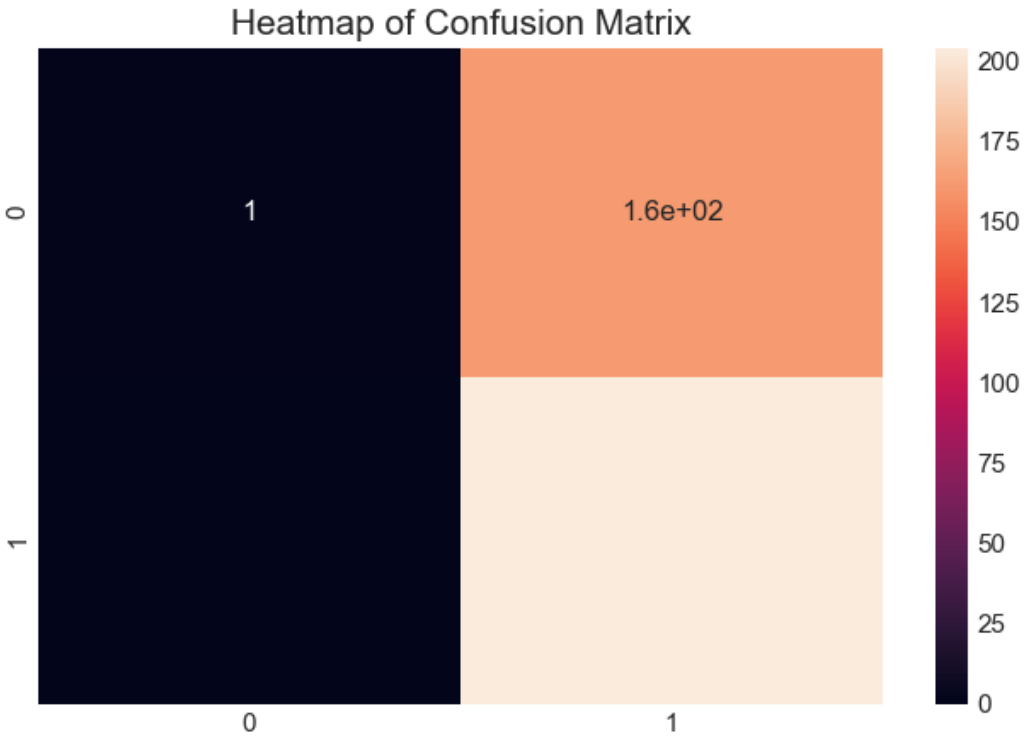


График №56. ROC Curve

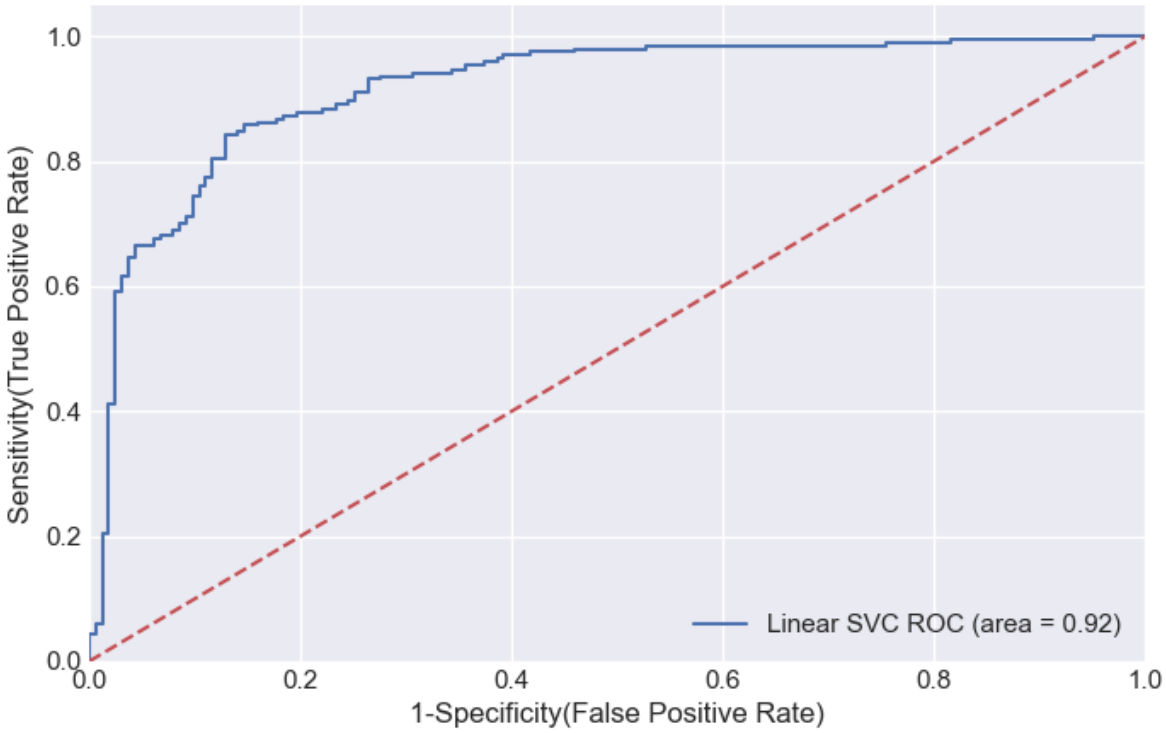
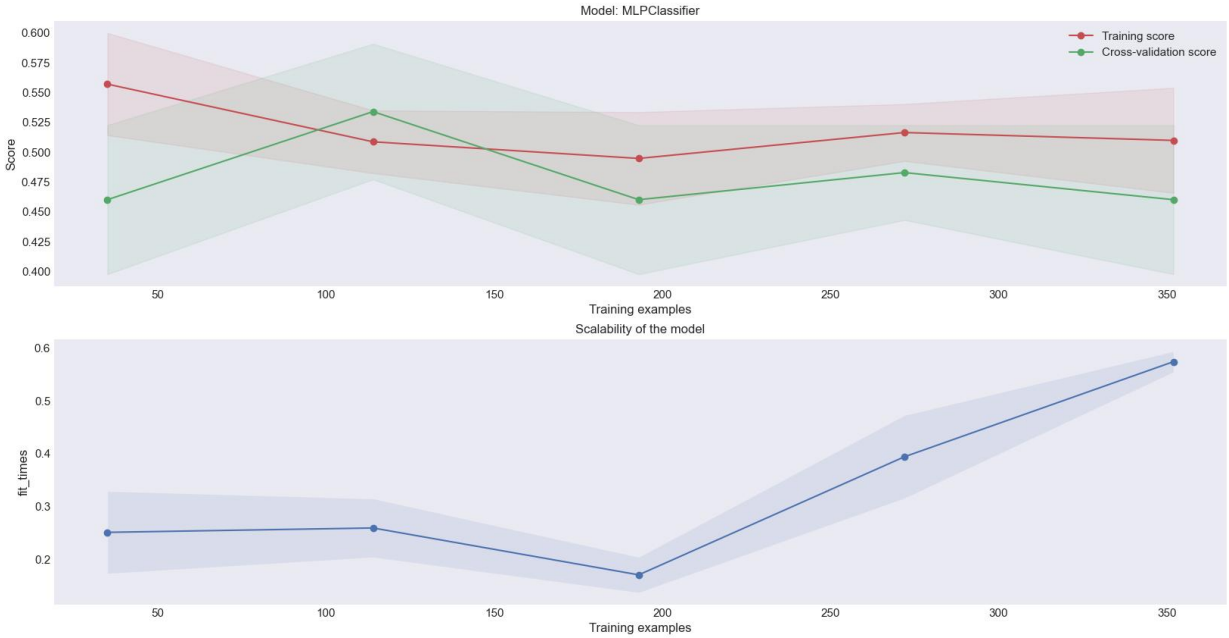


График №57. Score plot



Модель: MLPClassifier

The MLPClassifier optimizes the squared-loss using LBFGS or stochastic gradient descent by the Multi-layer Perceptron regressor. Reference Sklearn documentation.
Thanks to: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html#sklearn.neural_network.MLPRegressor https://stackoverflow.com/questions/44803596/scikit-learn-mlpregressor-performance-cap

Linear SVC is a similar to SVM method. Its also builds on kernel functions but is appropriate for unsupervised learning. Reference Wikipedia.

Таблица №19. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.0	0.0	0.0	164.0
class 1	0.552	0.995	0.71	203.0
accuracy	0.55	0.55	0.55	0.55
macro avg	0.276	0.498	0.355	367.0
weighted avg	0.305	0.55	0.393	367.0

График №58. Confusion Matrix

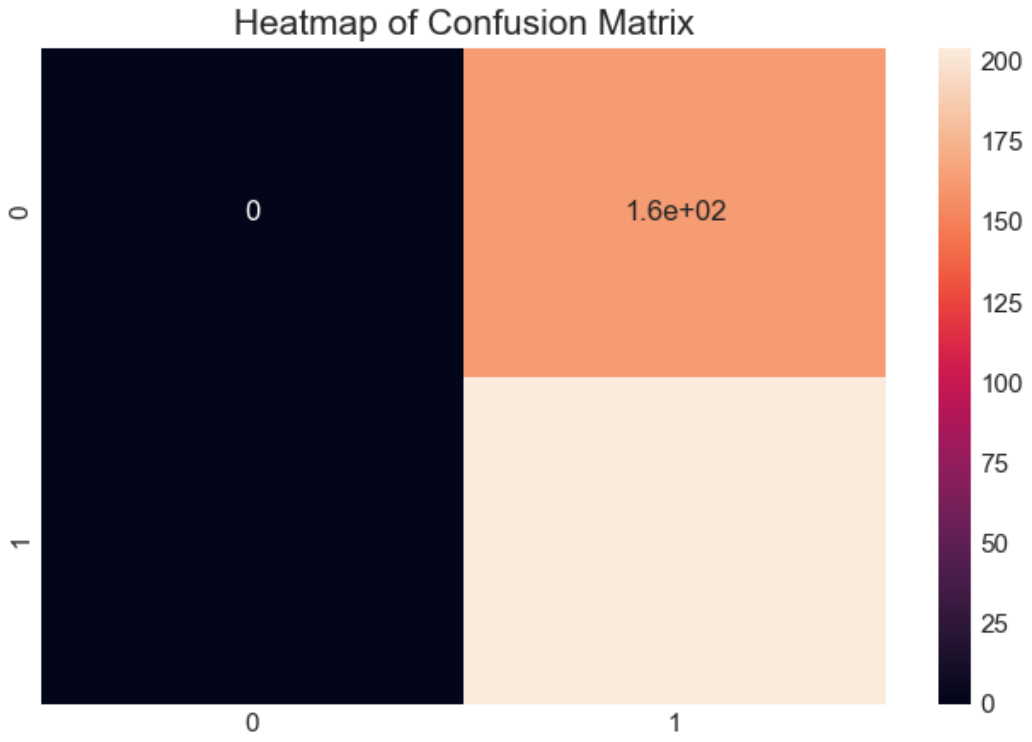


График №59. ROC Curve

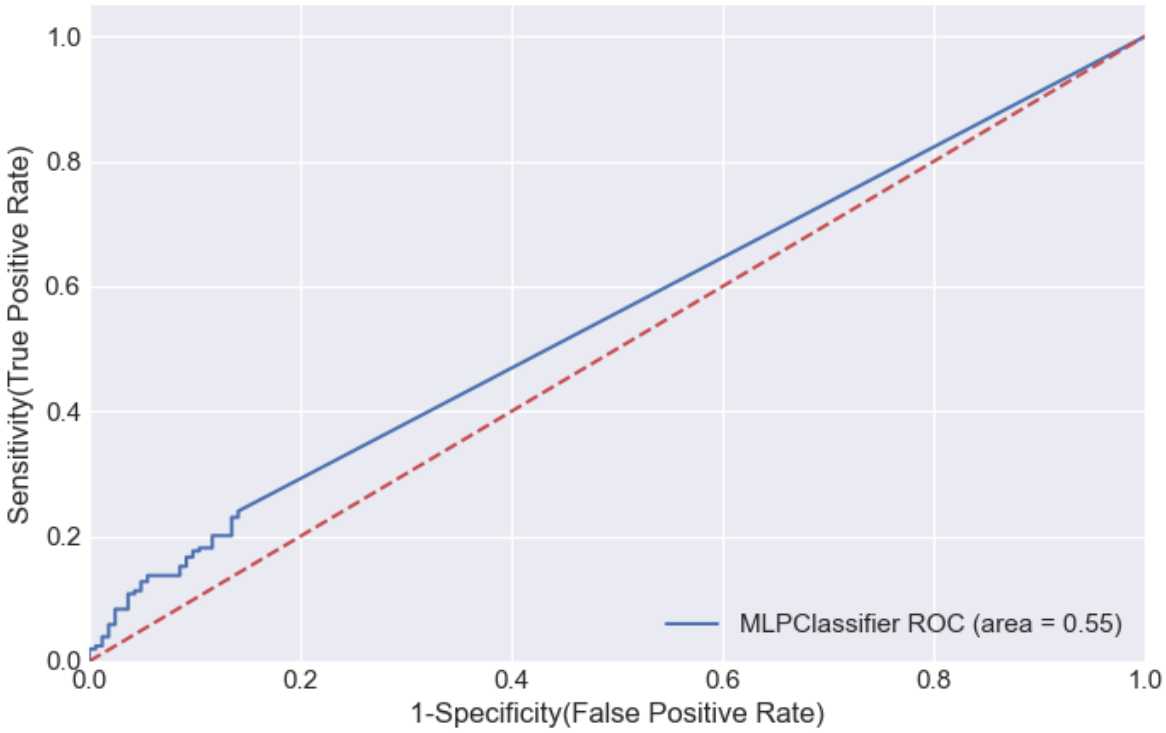
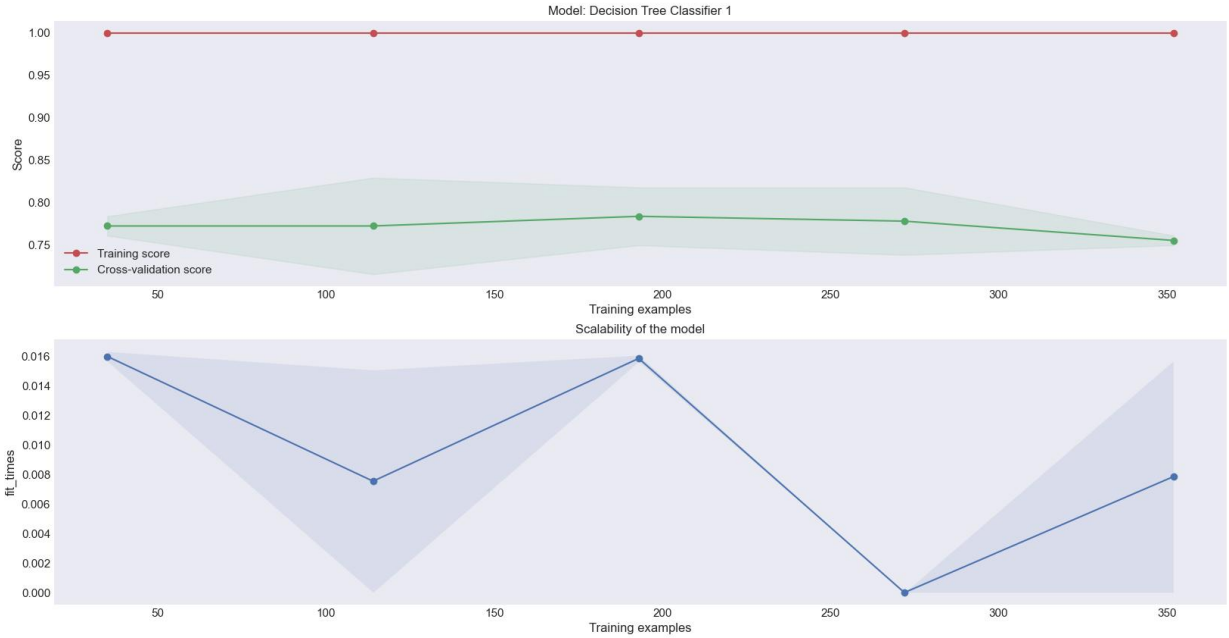


График №60. Score plot



Модель: Decision Tree Classifier 1

This model uses a Decision Tree as a predictive model which maps features (tree branches) to conclusions about the target value (tree leaves). Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Reference Wikipedia.

Таблица №20. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.756	0.774	0.765	164.0
class 1	0.814	0.798	0.806	203.0
accuracy	0.787	0.787	0.787	0.787
macro avg	0.785	0.786	0.786	367.0
weighted avg	0.788	0.787	0.788	367.0

График №61. Confusion Matrix

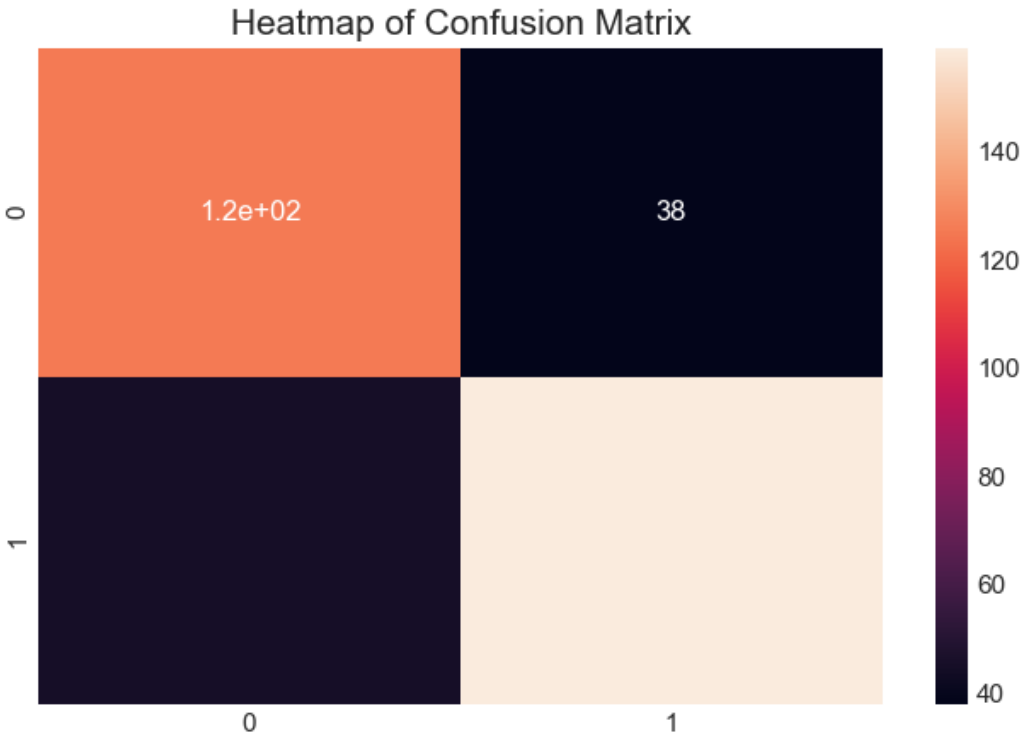


График №62. ROC Curve

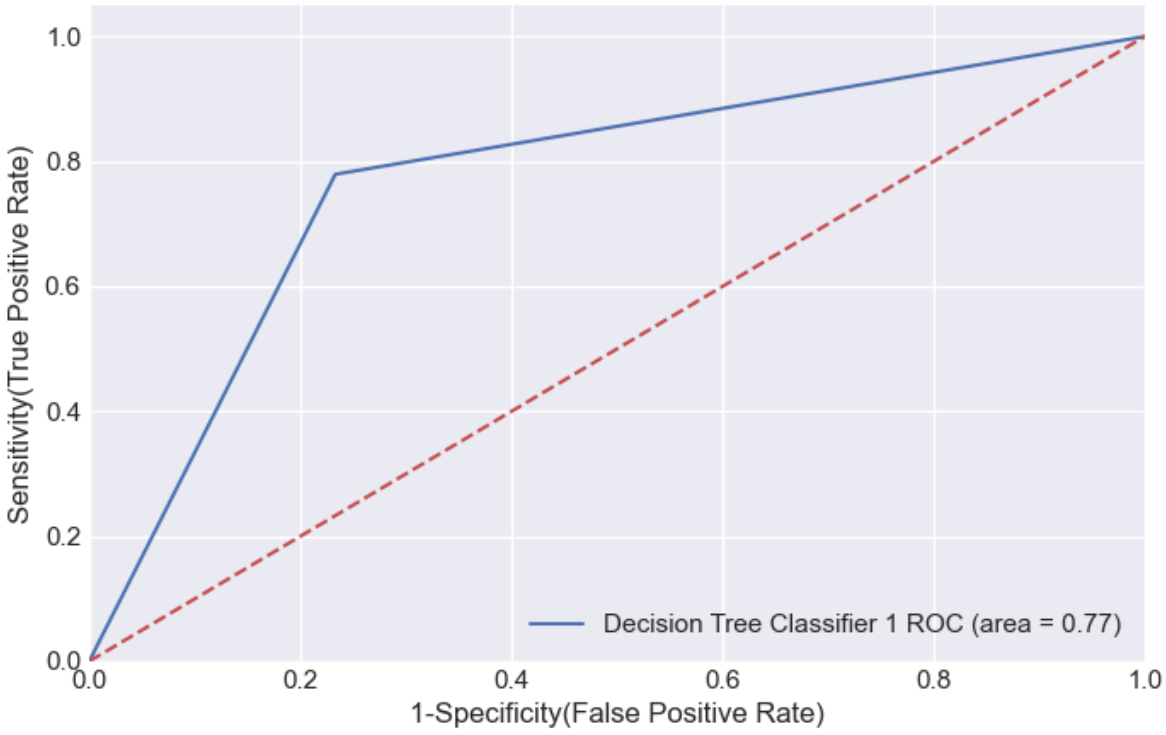
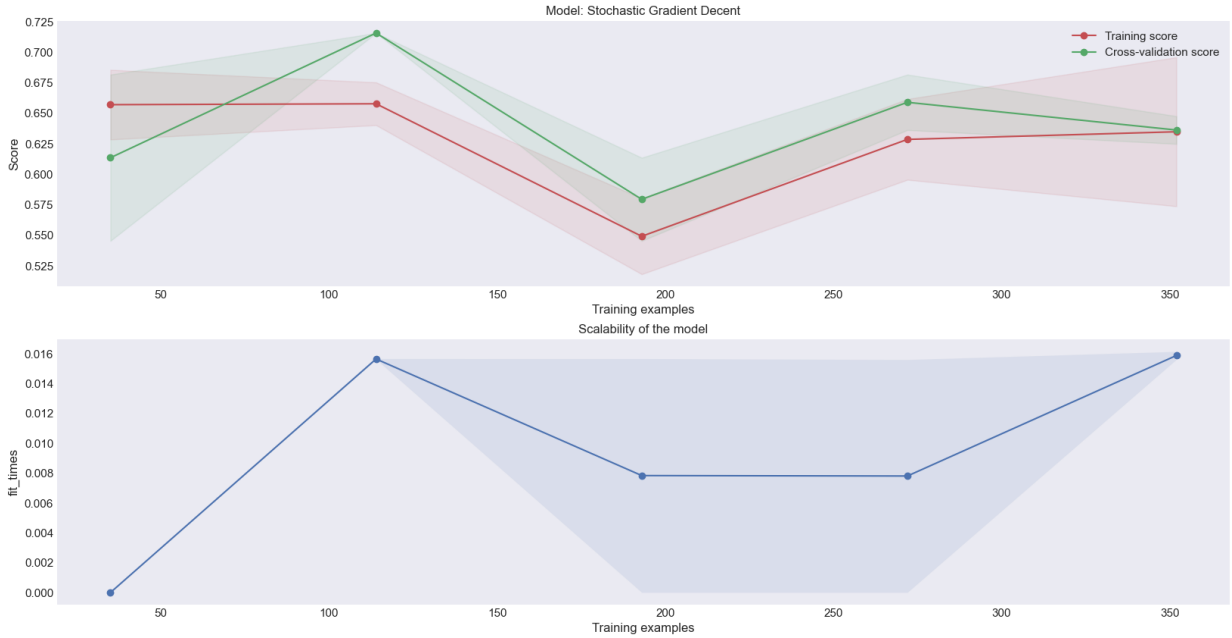


График №63. Score plot



Модель: Stochastic Gradient Decent

Stochastic gradient descent (often abbreviated SGD) is an iterative method for optimizing an objective function with suitable smoothness properties (e.g. differentiable or subdifferentiable). It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in big data applications this reduces the computational burden, achieving faster iterations in trade for a slightly lower convergence rate. Reference Wikipedia.

Таблица №21. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.63	0.841	0.721	164.0
class 1	0.824	0.601	0.695	203.0
accuracy	0.708	0.708	0.708	0.708
macro avg	0.727	0.721	0.708	367.0
weighted avg	0.738	0.708	0.707	367.0

График №64. Confusion Matrix

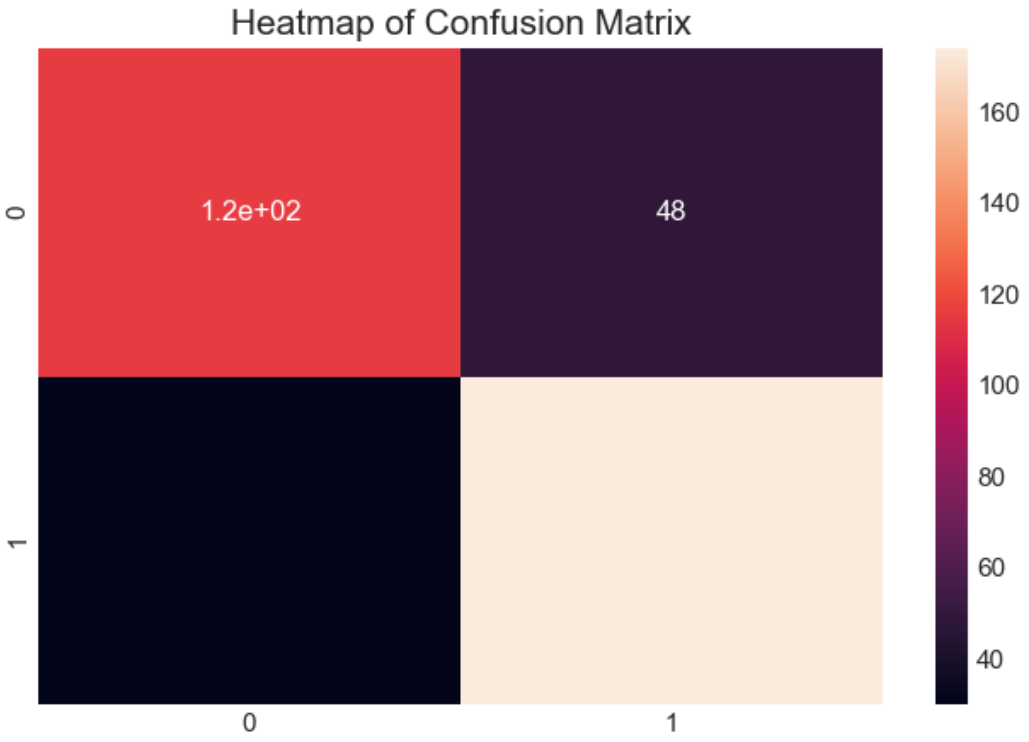


График №65. ROC Curve

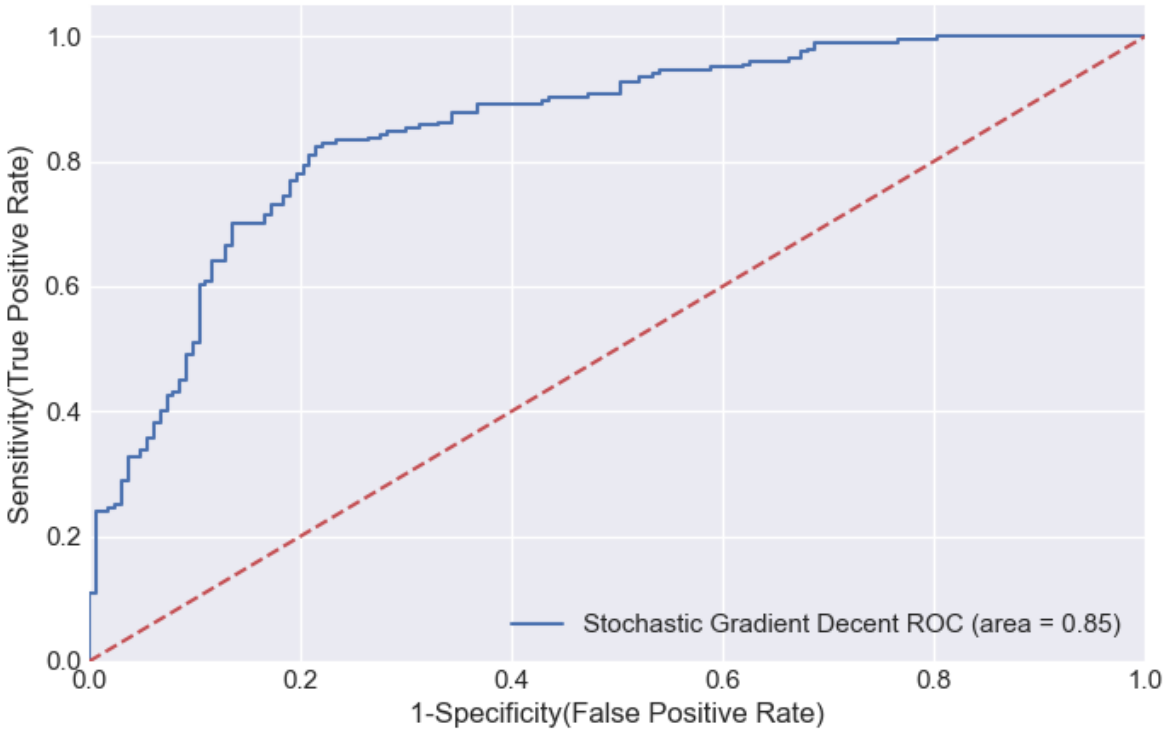
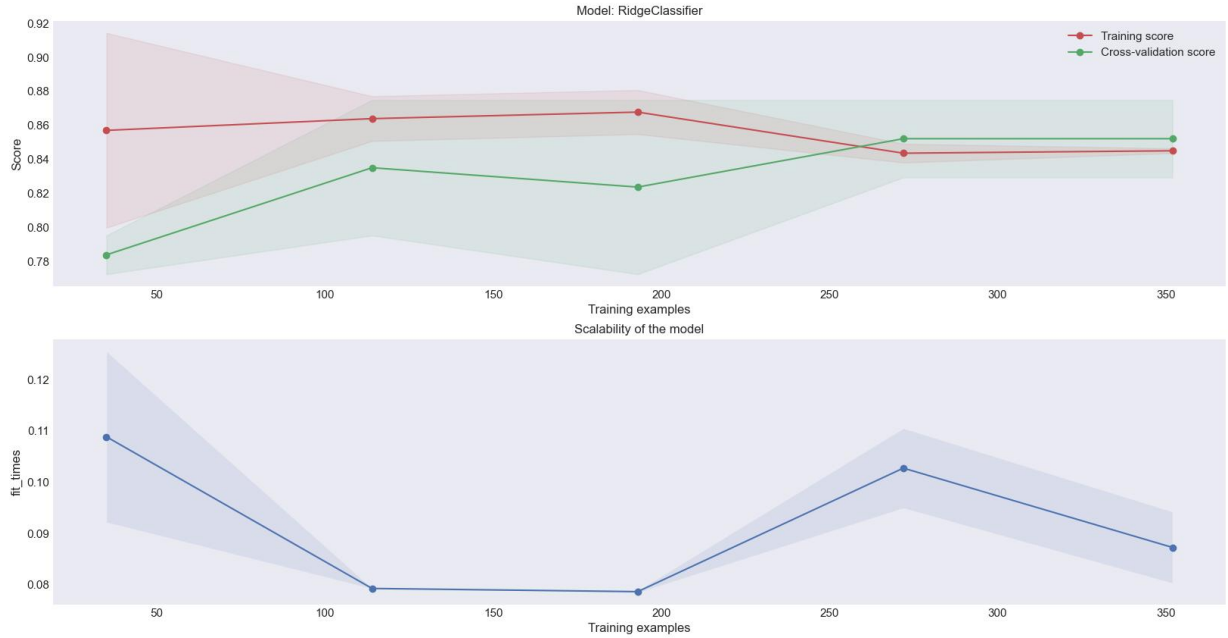


График №66. Score plot



Модель: RidgeClassifier

Tikhonov Regularization, colloquially known as Ridge Classifier, is the most commonly used regression algorithm to approximate an answer for an equation with no unique solution. This type of problem is very common in machine learning tasks, where the "best" solution must be chosen using limited data. If a unique solution exists, algorithm will return the optimal value. However, if multiple solutions exist, it may choose any of them. Reference Brilliant.org.

Таблица №22. Таблица классификации

class 0	0.819	0.829	0.824	164.0
class 1	0.861	0.852	0.856	203.0
accuracy	0.842	0.842	0.842	0.842
macro avg	0.84	0.841	0.84	367.0
weighted avg	0.842	0.842	0.842	367.0

График №67. Confusion Matrix

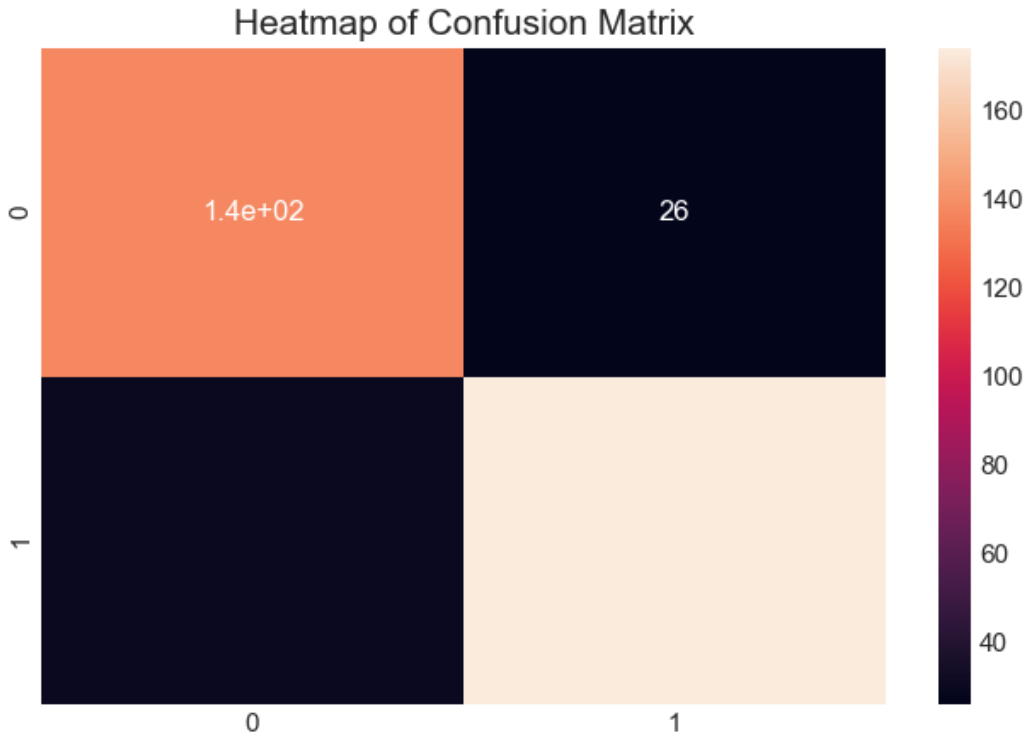


График №68. ROC Curve

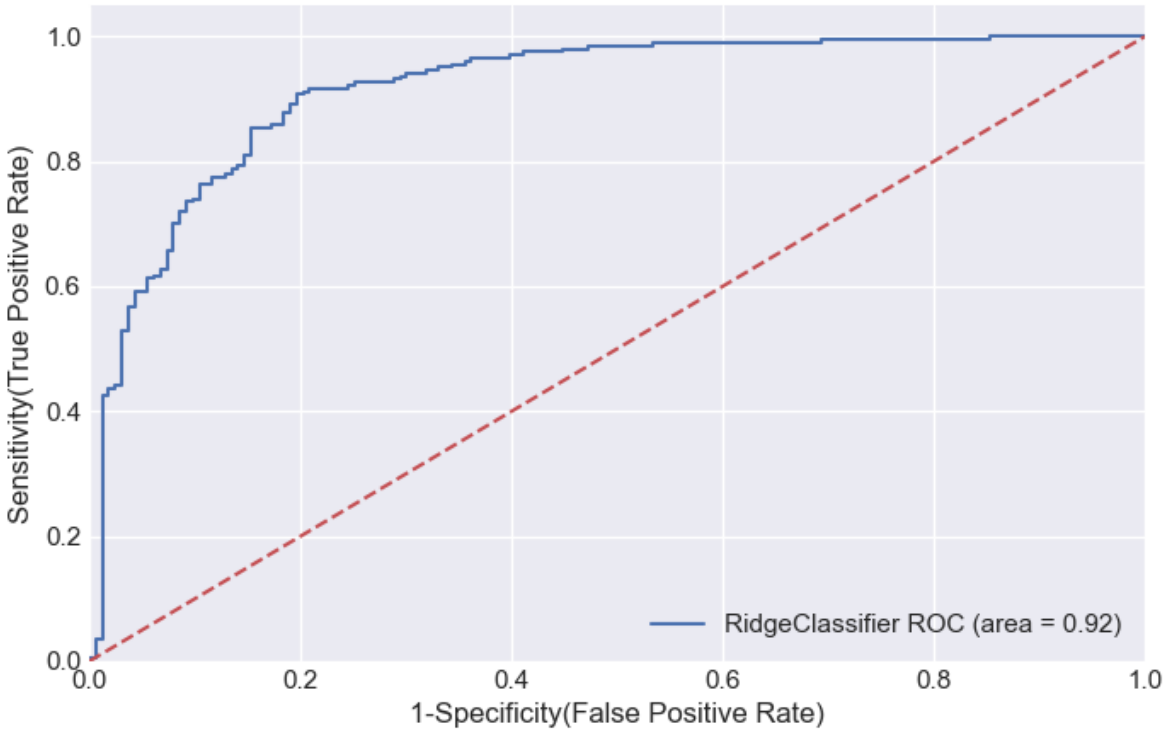
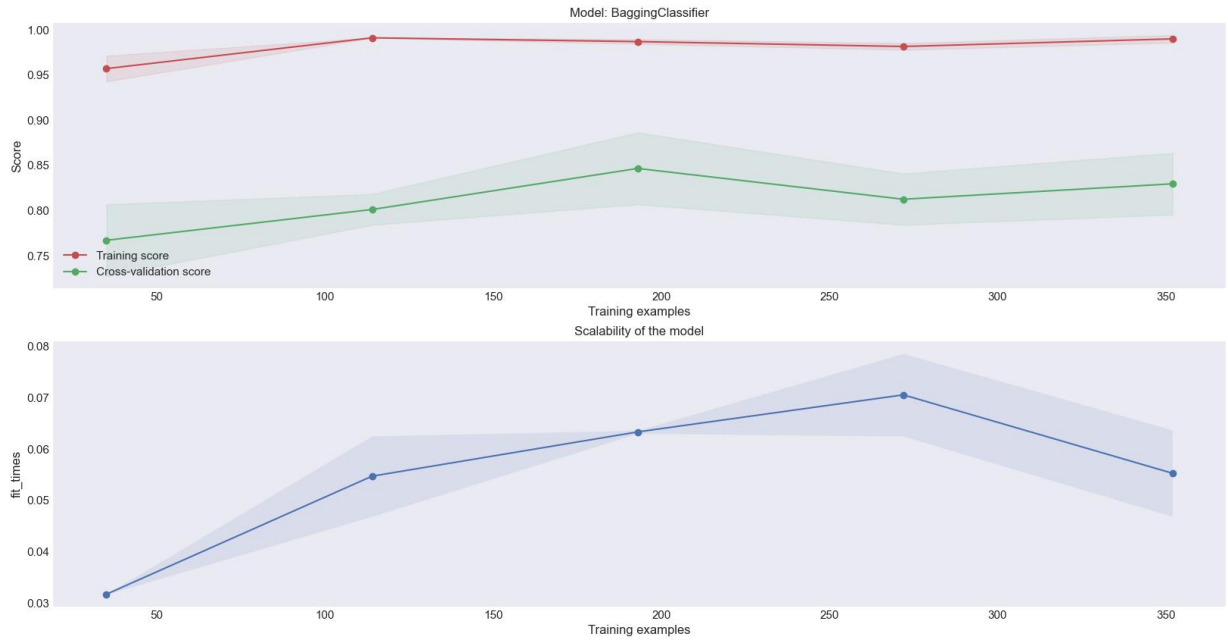


График №69. Score plot



Модель: BaggingClassifier

Bootstrap aggregating, also called Bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach. Bagging leads to "improvements for unstable procedures", which include, for example, artificial neural networks, classification and regression trees, and subset selection in linear regression. On the other hand, it can mildly degrade the performance of stable methods such as K-nearest neighbors. Reference Wikipedia.

Таблица №23. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.799	0.799	0.799	164.0
class 1	0.837	0.837	0.837	203.0
accuracy	0.82	0.82	0.82	0.82
macro avg	0.818	0.818	0.818	367.0
weighted avg	0.82	0.82	0.82	367.0

График №70. Confusion Matrix

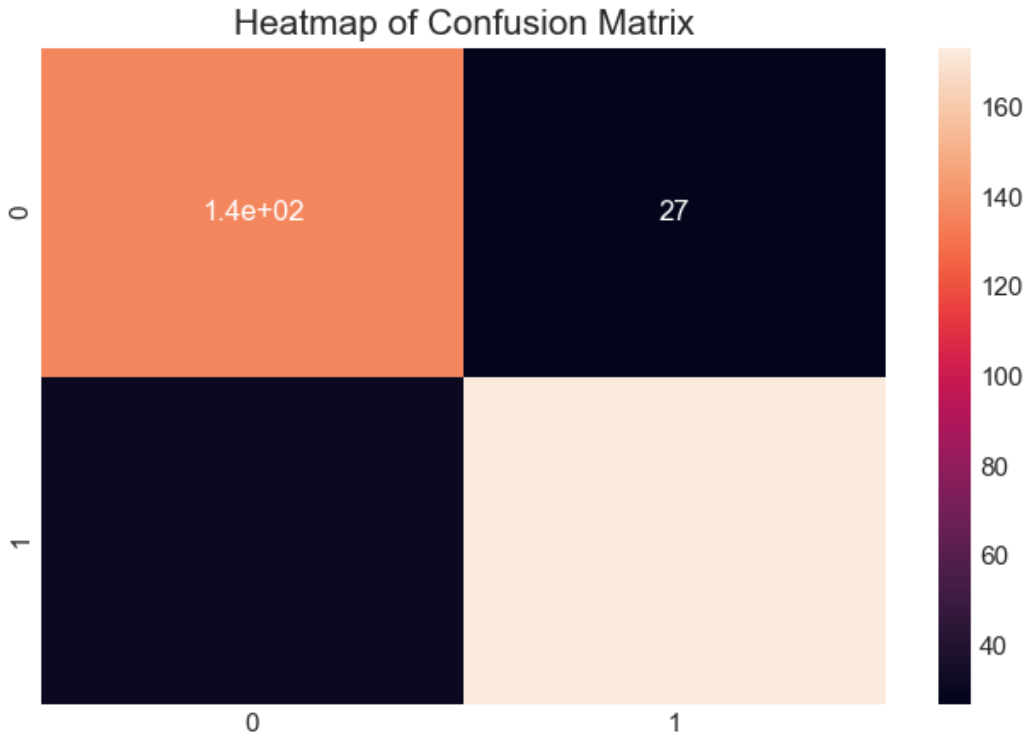


График №71. ROC Curve

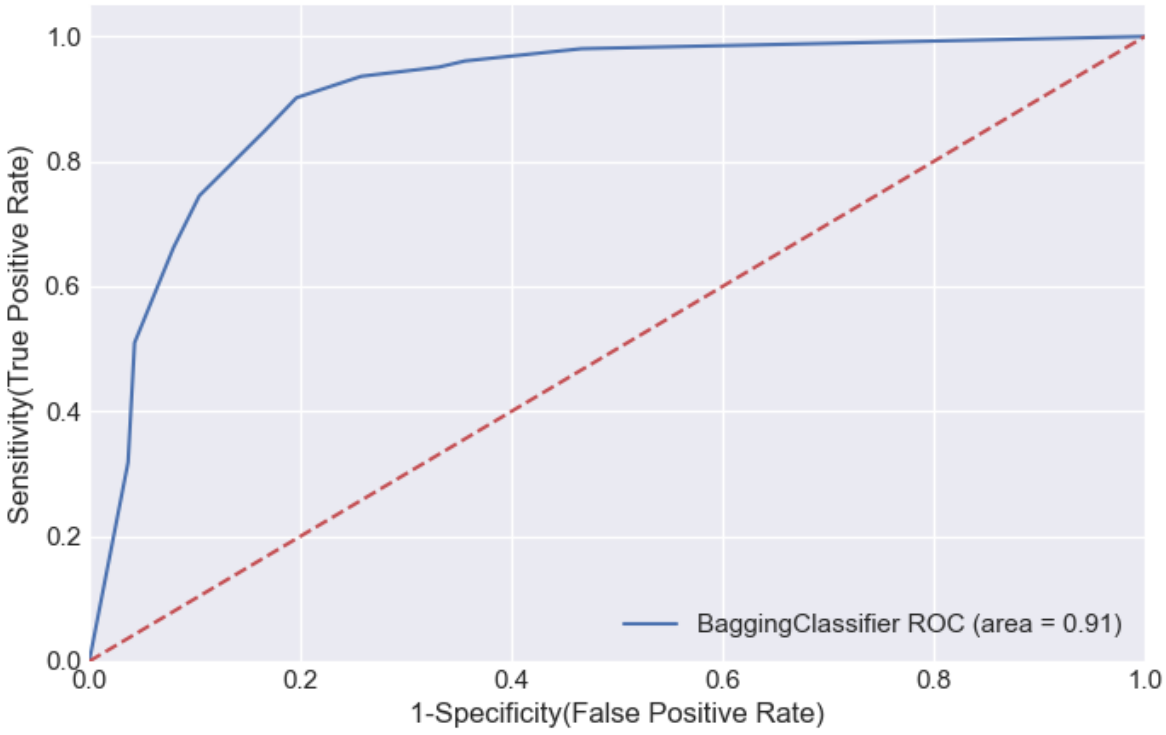
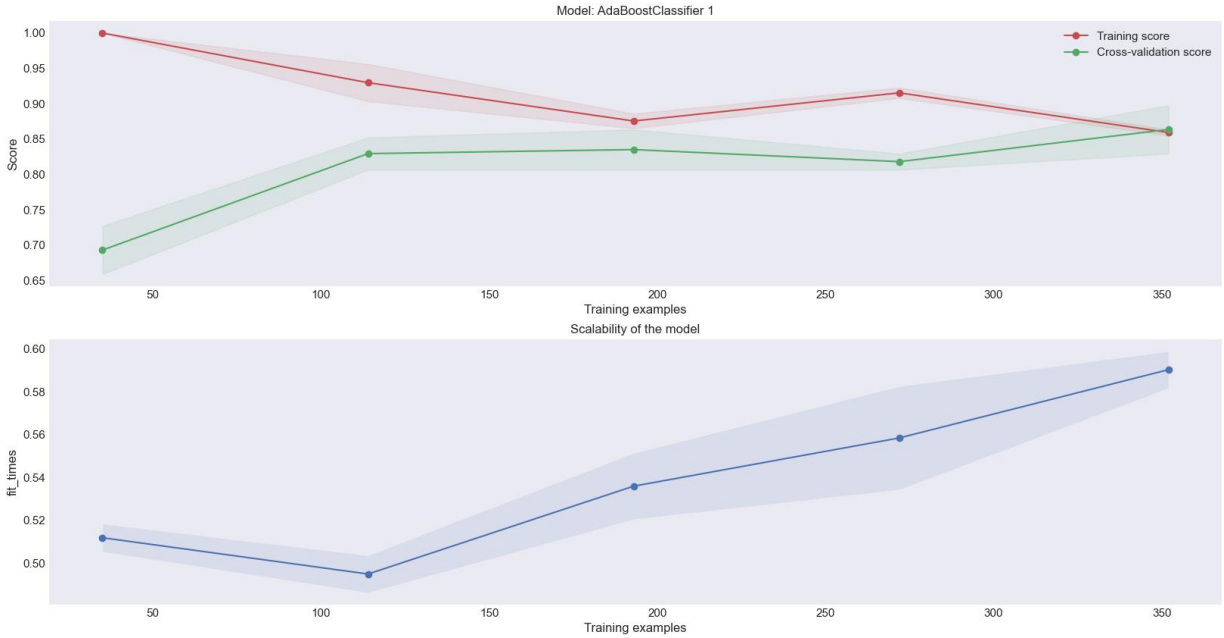


График №72. Score plot



Модель: AdaBoostClassifier 1

The core principle of AdaBoost ("Adaptive Boosting") is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying N weights to each of the training samples. Initially, those weights are all set to 1/N, so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence. Reference sklearn documentation.

Таблица №24. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.829	0.829	0.829	164.0
class 1	0.862	0.862	0.862	203.0
accuracy	0.847	0.847	0.847	0.847
macro avg	0.846	0.846	0.846	367.0
weighted avg	0.847	0.847	0.847	367.0

График №73. Confusion Matrix

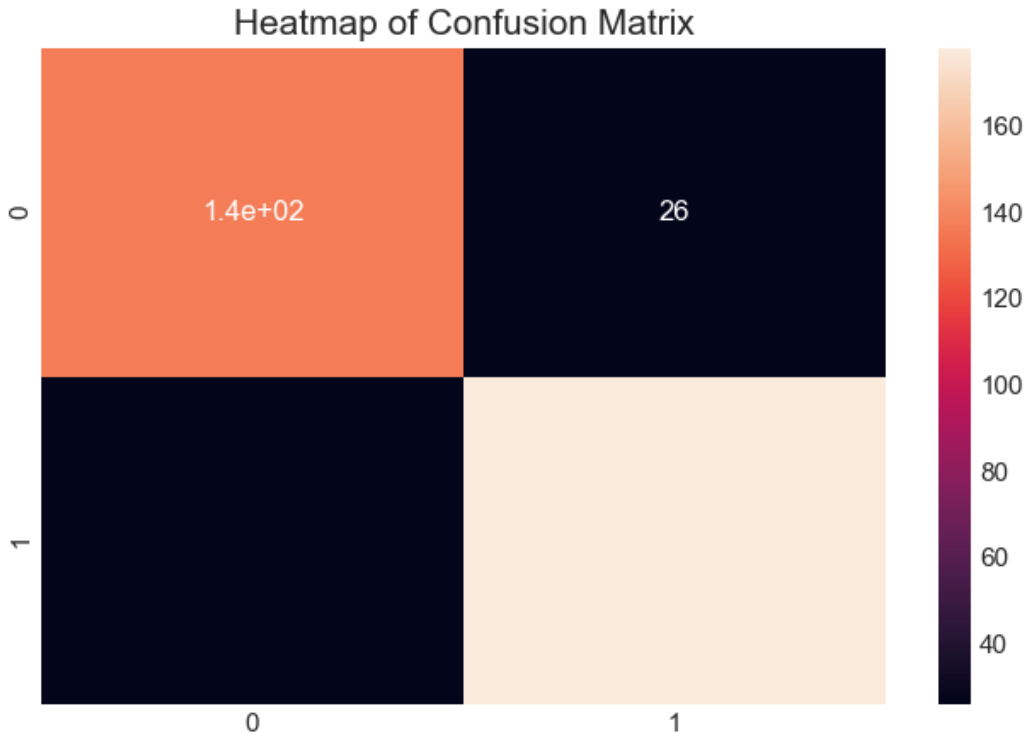


График №74. ROC Curve

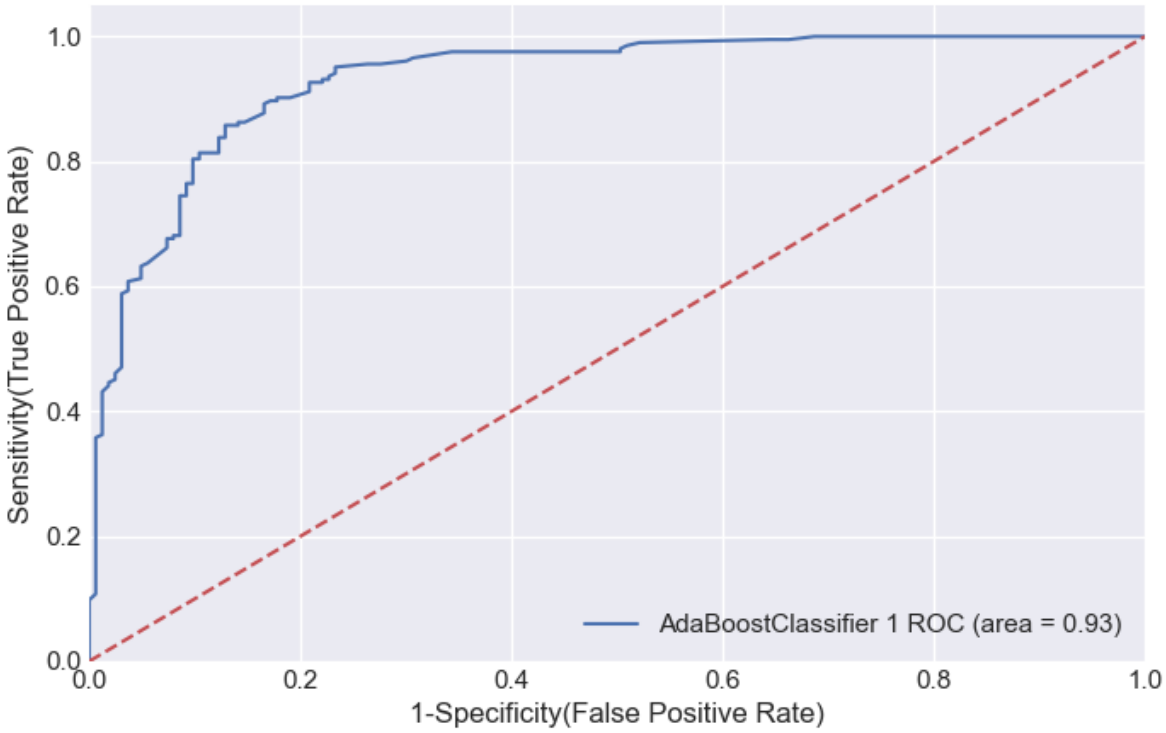
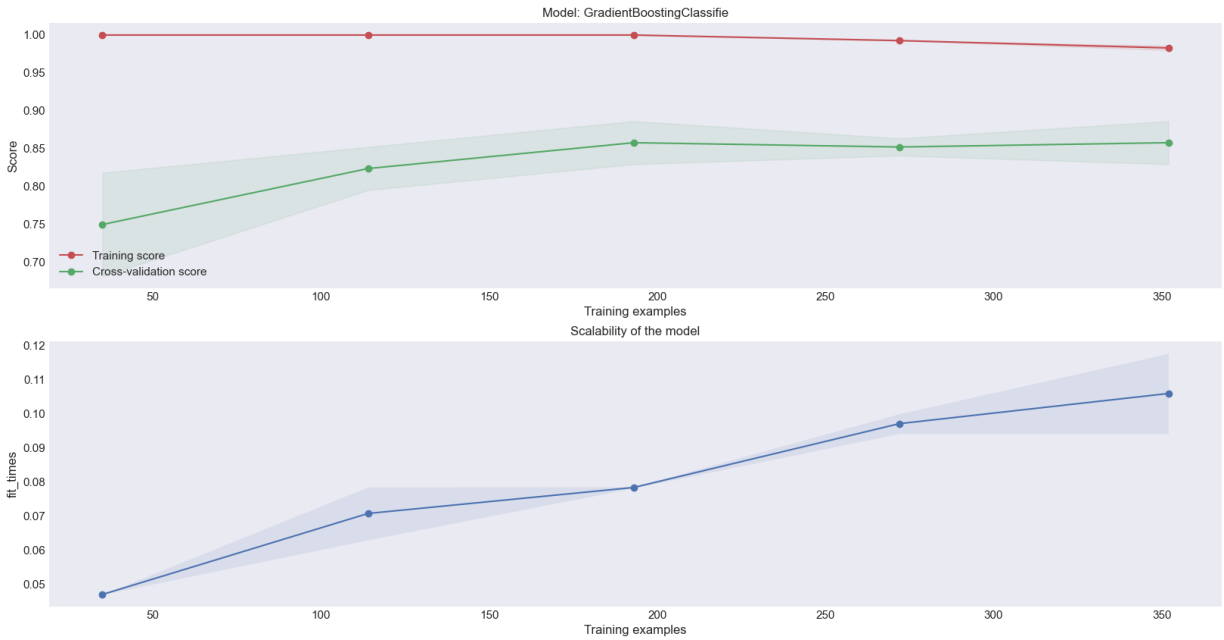


График №75. Score plot



Модель: GradientBoostingClassifie

Thanks to https://www.kaggle.com/kabure/titanic-eda-model-pipeline-keras-nn
Gradient Boosting builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage n_classes_ regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function. Binary classification is a special case where only a single regression tree is induced. The features are always randomly permuted at each split. Therefore, the best found split may vary, even with the same training data and max_features=n_features, if the improvement of the criterion is identical for several splits enumerated during the search of the best split. To obtain a deterministic behaviour during fitting, random_state has to be fixed. Reference sklearn documentation.

Таблица №25. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.856	0.835	0.846	164.0
class 1	0.87	0.887	0.878	203.0
accuracy	0.864	0.864	0.864	0.864
macro avg	0.863	0.861	0.862	367.0
weighted avg	0.864	0.864	0.864	367.0

График №76. Confusion Matrix

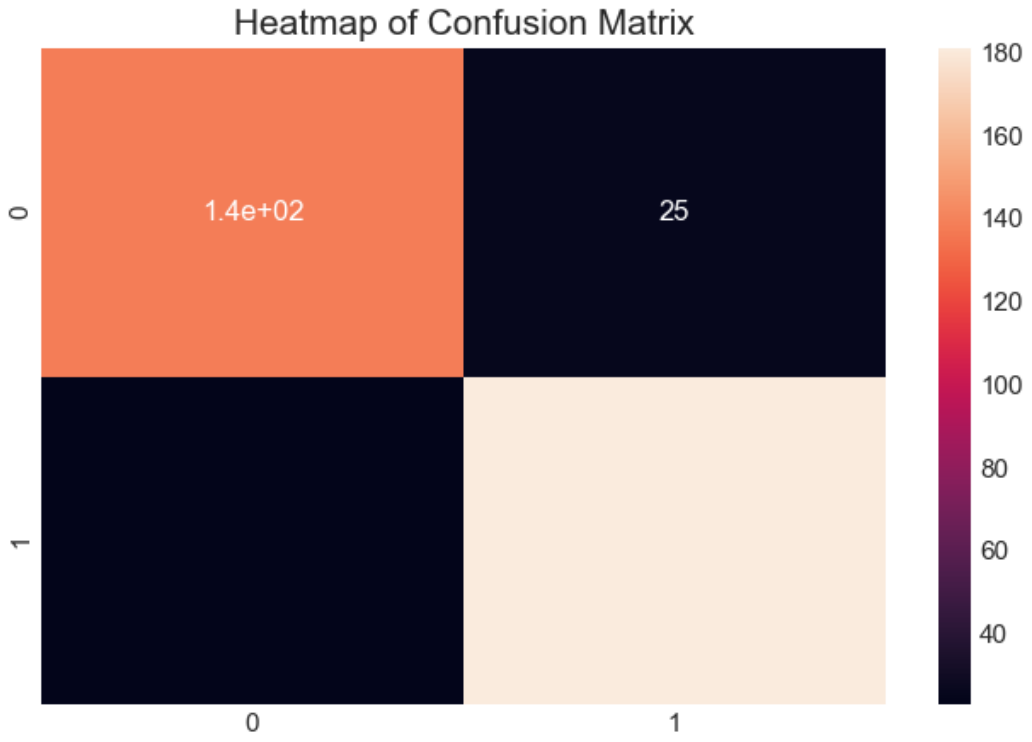


График №77. ROC Curve

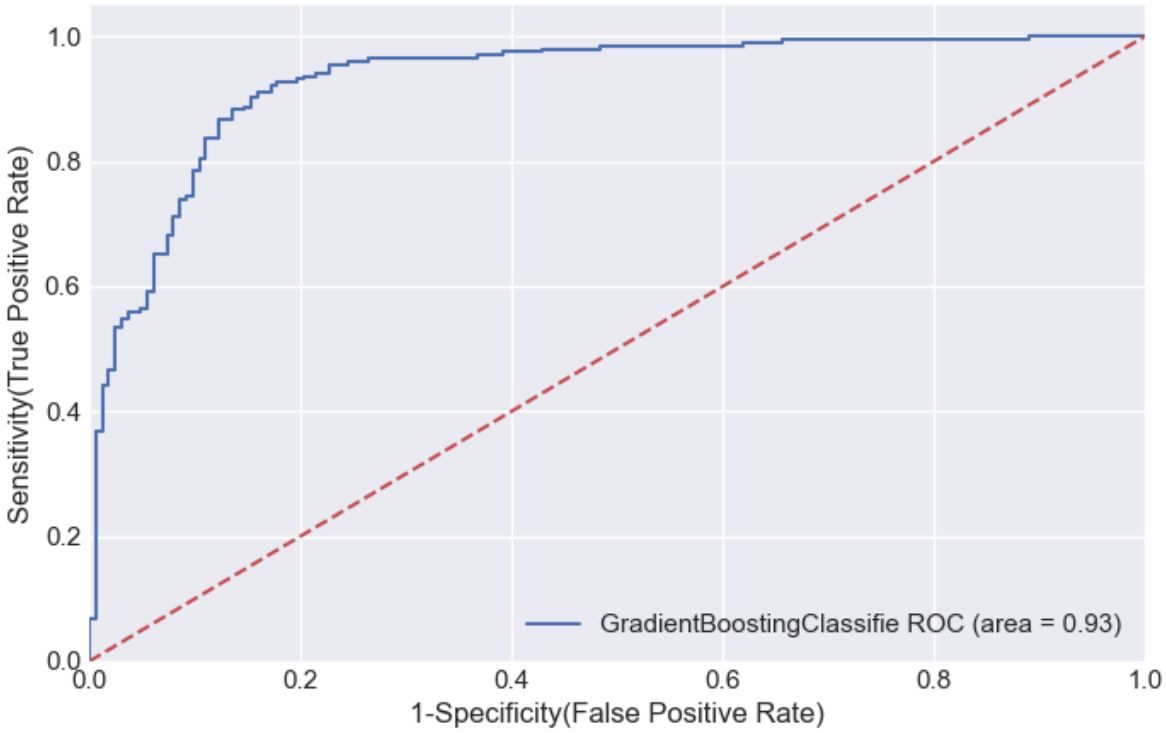
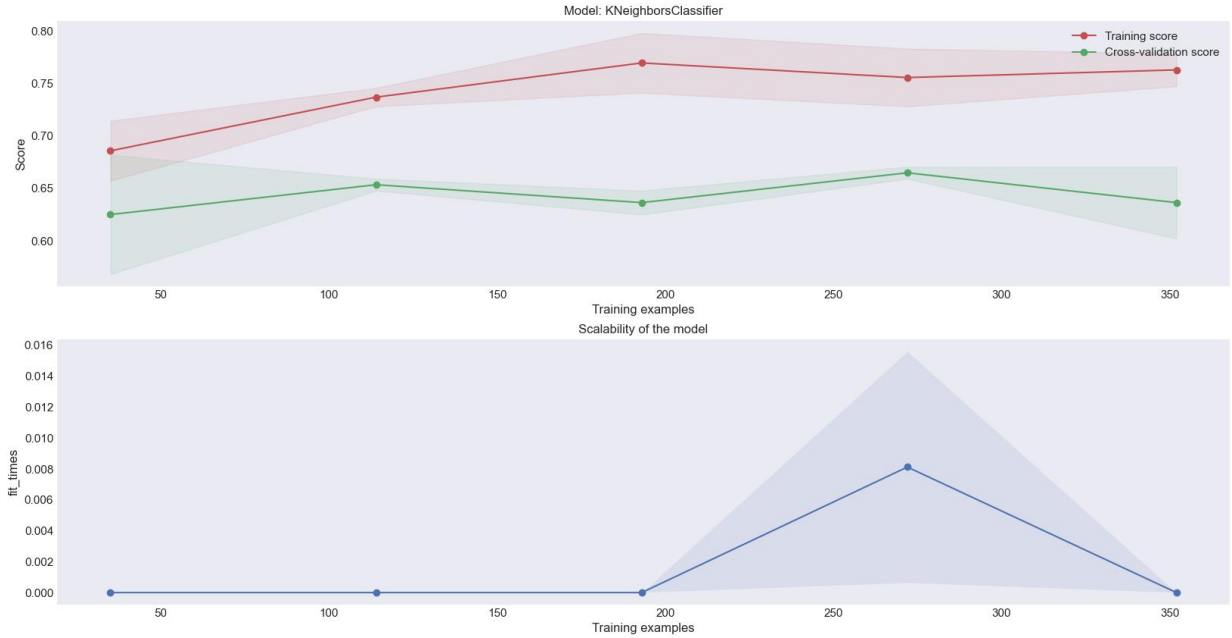


График №78. Score plot



Модель: KNeighborsClassifier

Thanks to https://www.kaggle.com/startupsci/titanic-data-science-solutions
In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for classification and regression. A sample is classified by a majority vote of its neighbors, with the sample being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). Reference Wikipedia.

Таблица №26. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.662	0.61	0.635	164.0
class 1	0.704	0.749	0.726	203.0
accuracy	0.687	0.687	0.687	0.687
macro avg	0.683	0.679	0.68	367.0
weighted avg	0.685	0.687	0.685	367.0
© Dr. Alexander Wagner. Все права охраняются законом				

График №79. Confusion Matrix

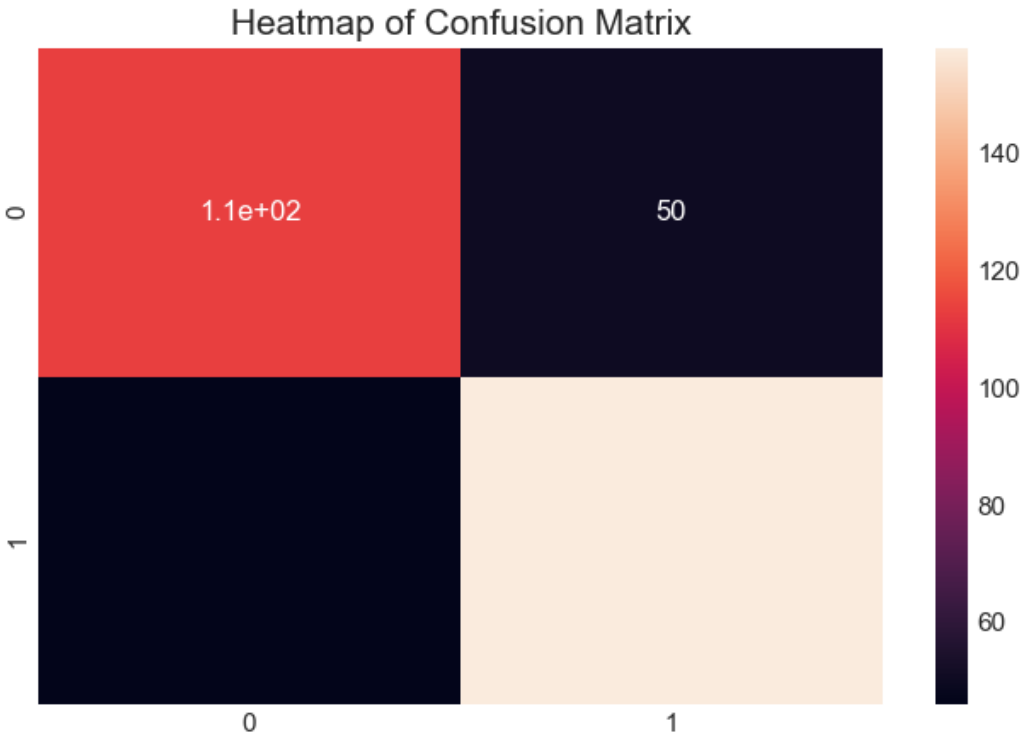


График №80. ROC Curve

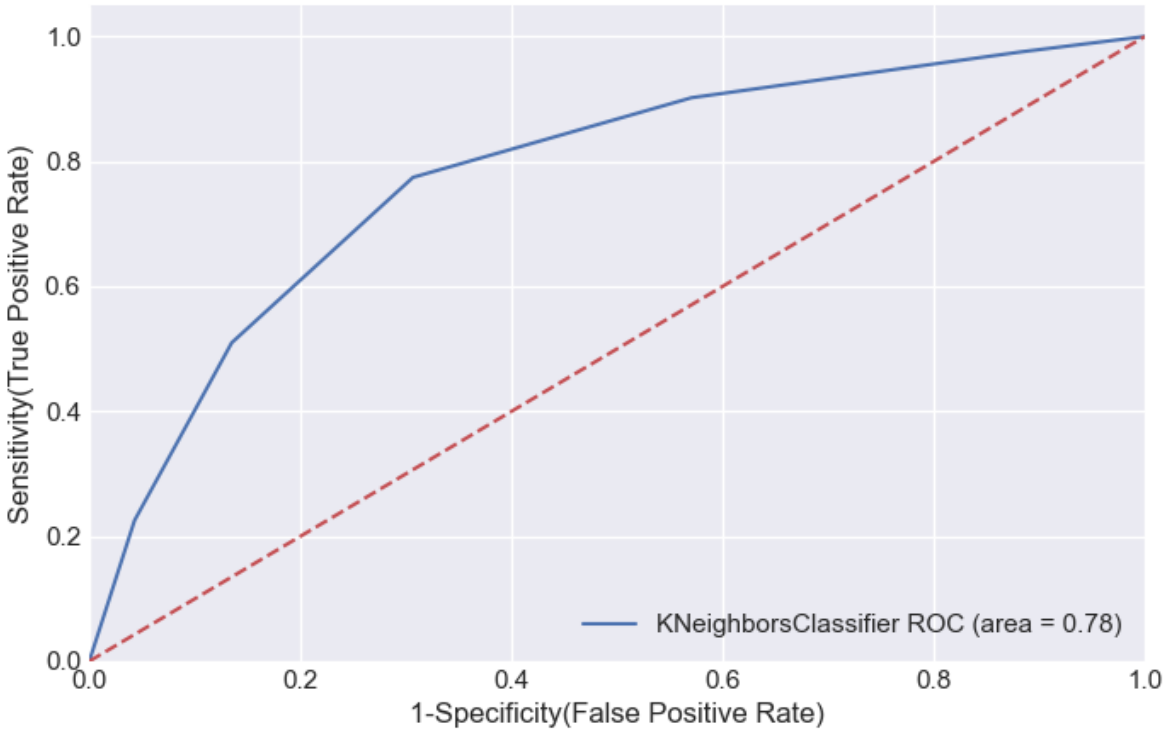
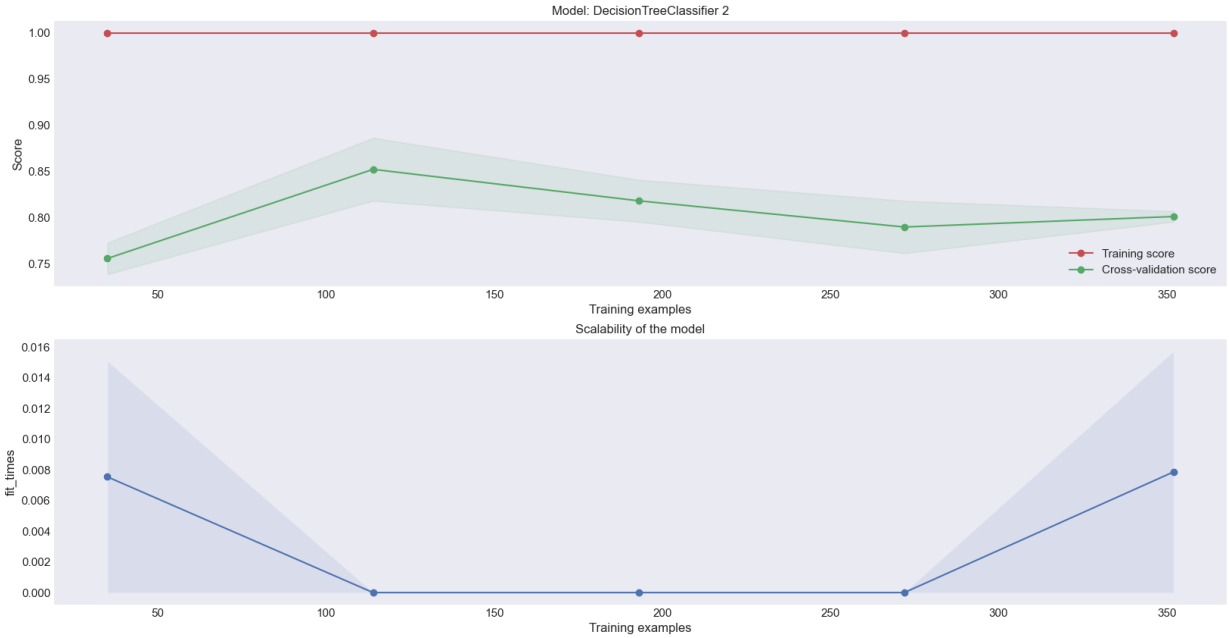


График №81. Score plot



Модель: DecisionTreeClassifier 2

This model uses a Decision Tree as a predictive model which maps features (tree branches) to conclusions about the target value (tree leaves). Tree models where the target variable can take a finite set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called regression trees. Reference Wikipedia.

Таблица №27. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.776	0.78	0.778	164.0
class 1	0.822	0.818	0.82	203.0
accuracy	0.801	0.801	0.801	0.801
macro avg	0.799	0.799	0.799	367.0
weighted avg	0.801	0.801	0.801	367.0

График №82. Confusion Matrix

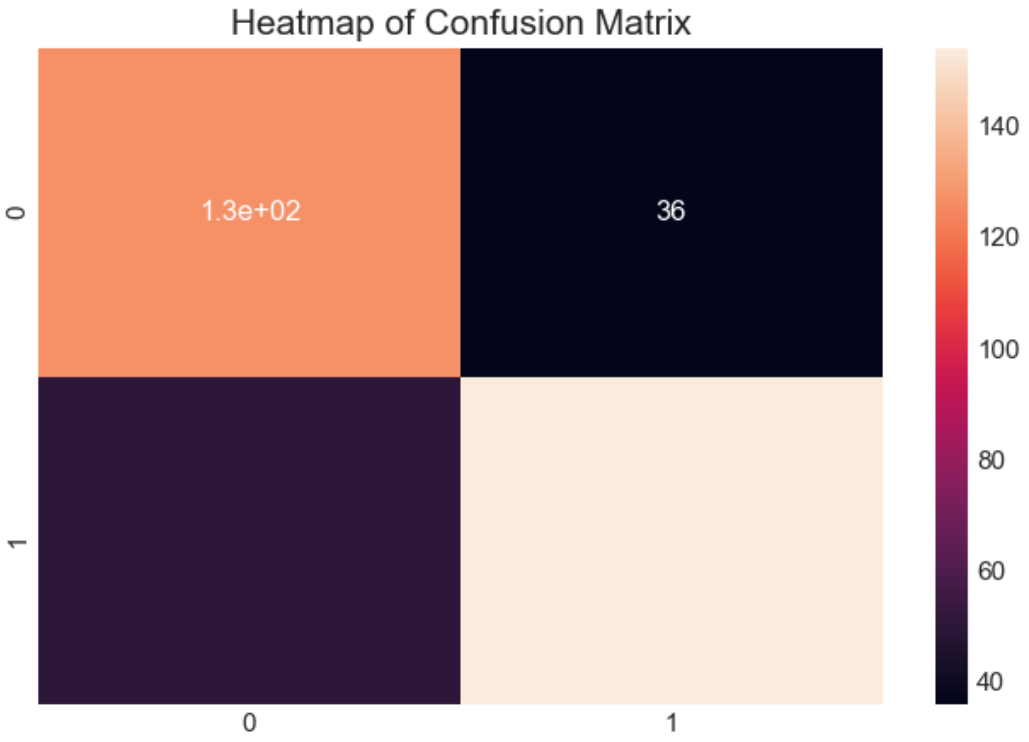


График №83. ROC Curve

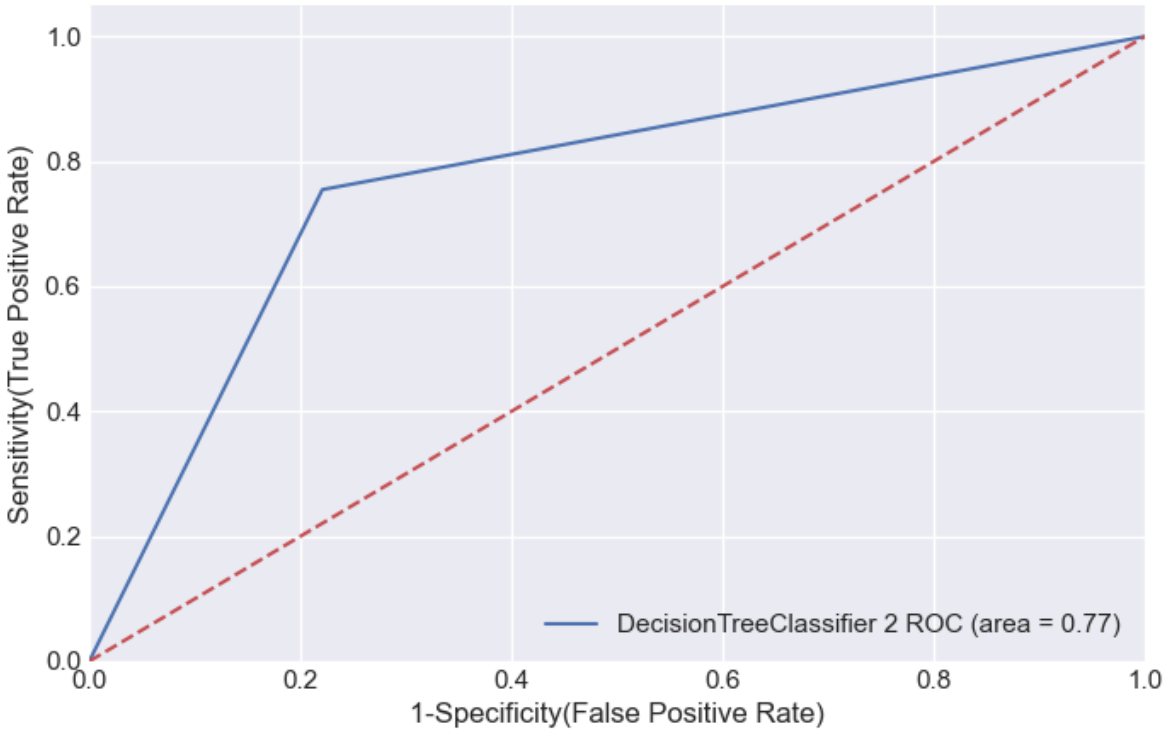
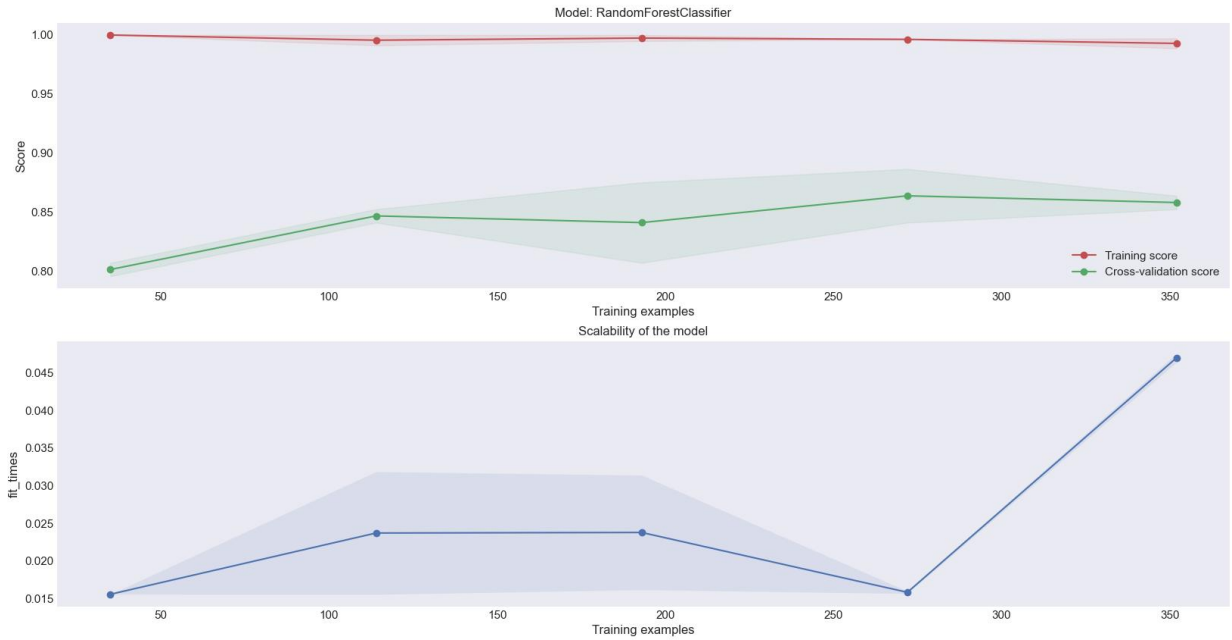


График №84. Score plot



Модель: RandomForestClassifier

Random Forest is one of the most popular model. Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees (n_estimators= [100, 300]) at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Reference Wikipedia.

Таблица №28. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.825	0.835	0.83	164.0
class 1	0.866	0.857	0.861	203.0
accuracy	0.847	0.847	0.847	0.847
macro avg	0.845	0.846	0.846	367.0
weighted avg	0.848	0.847	0.847	367.0

График №85. Confusion Matrix

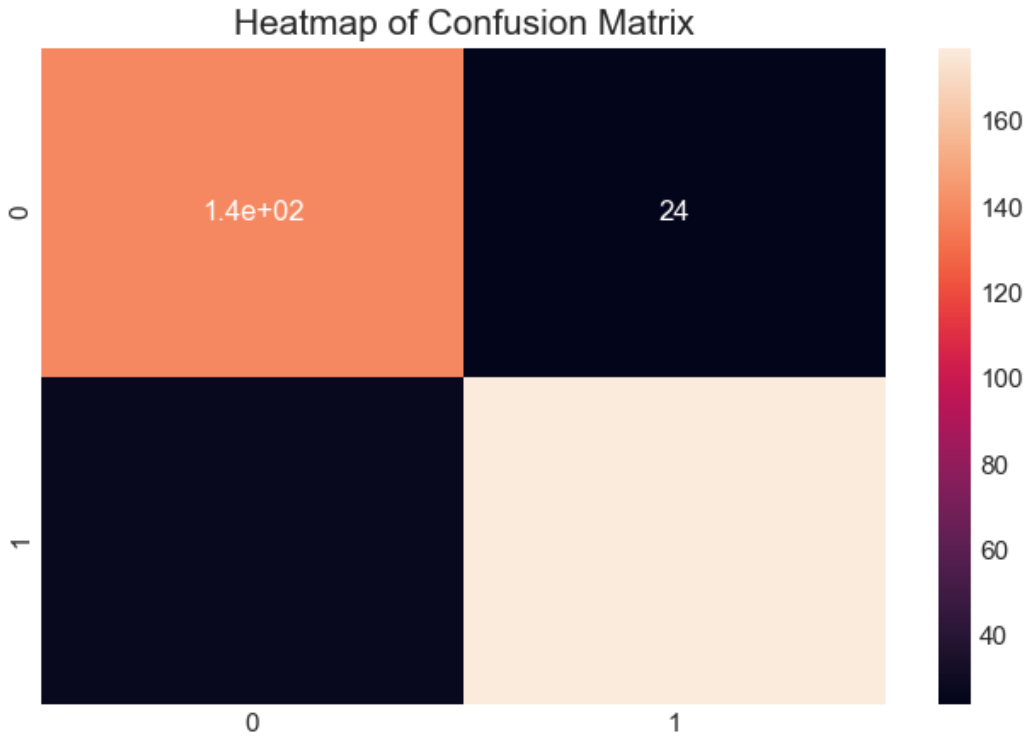


График №86. ROC Curve

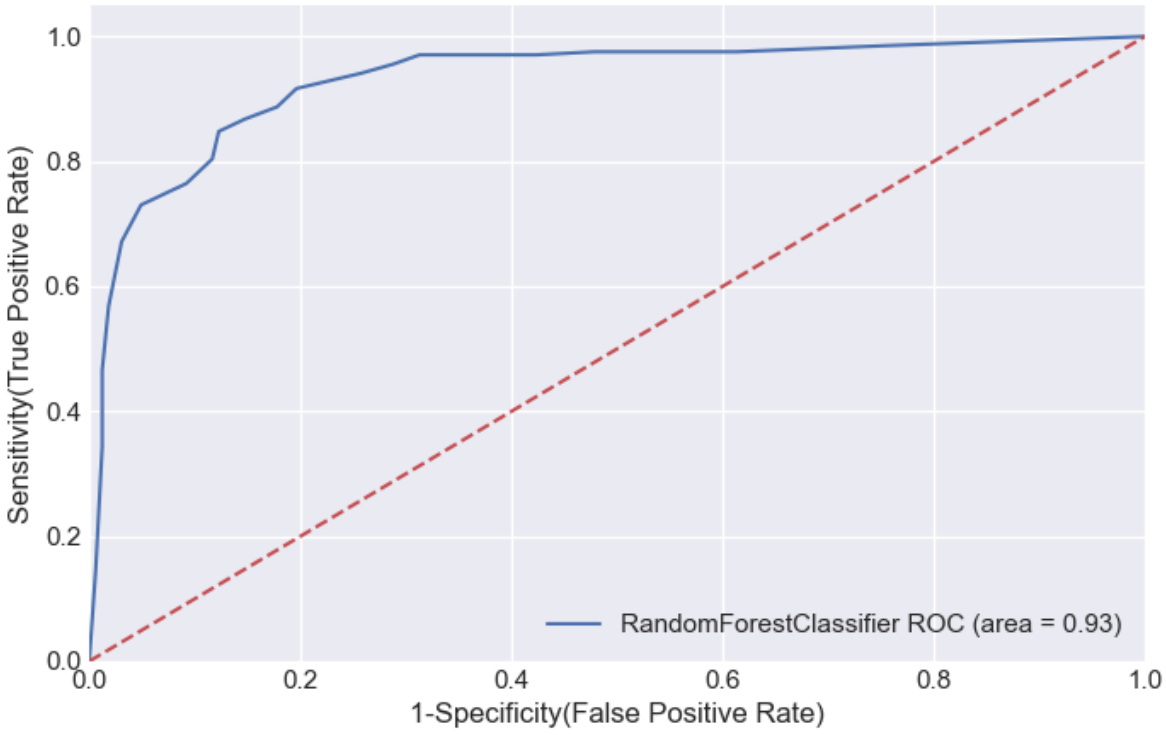
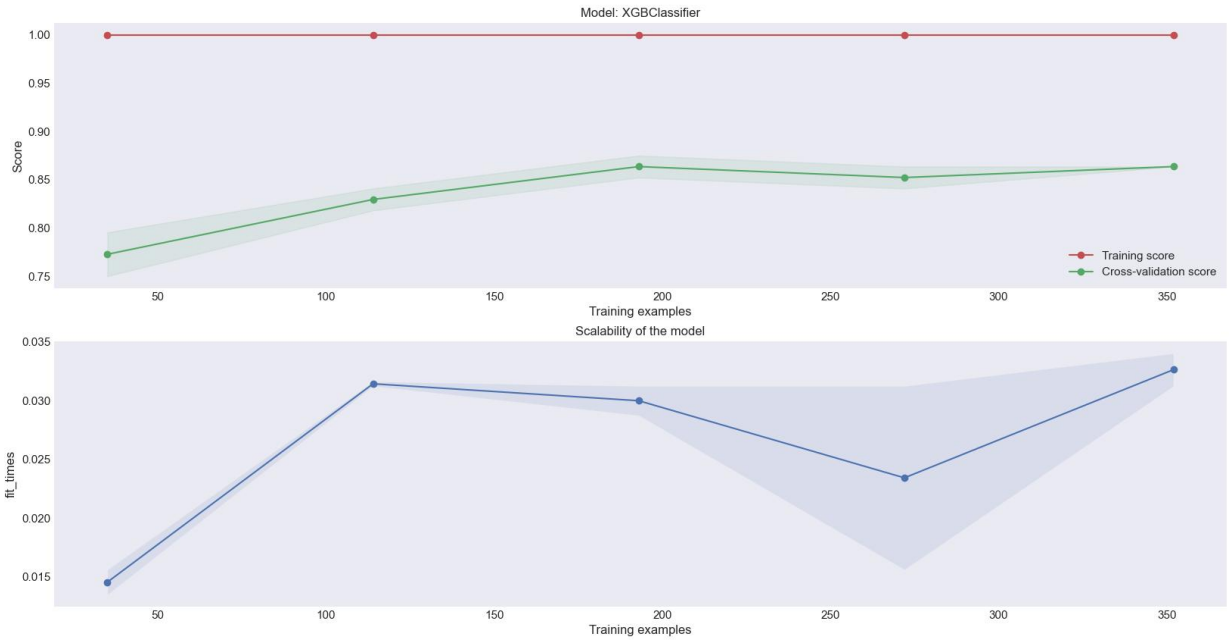


График №87. Score plot



Модель: XGBClassifier

XGBoost is an ensemble tree method that apply the principle of boosting weak learners (CARTs generally) using the gradient descent architecture. XGBoost improves upon the base Gradient Boosting Machines (GBM) framework through systems optimization and algorithmic enhancements. Reference Towards Data Science.

Таблица №29. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.834	0.829	0.832	164.0
class 1	0.863	0.867	0.865	203.0
accuracy	0.85	0.85	0.85	0.85
macro avg	0.849	0.848	0.848	367.0
weighted avg	0.85	0.85	0.85	367.0

График №88. Confusion Matrix

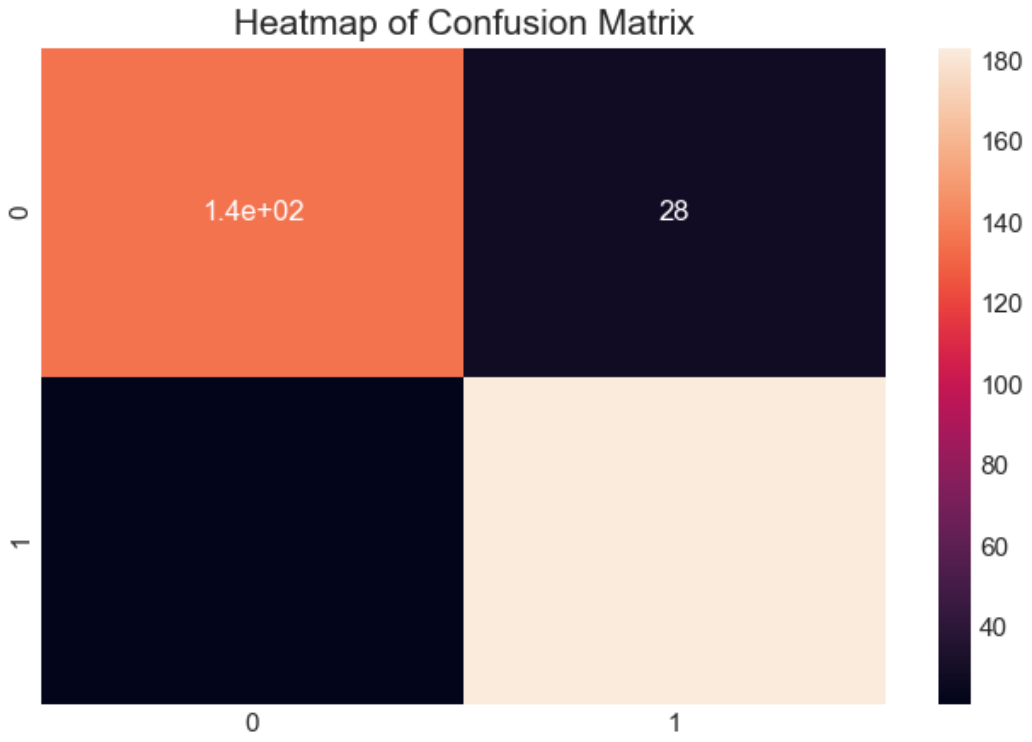


График №89. ROC Curve

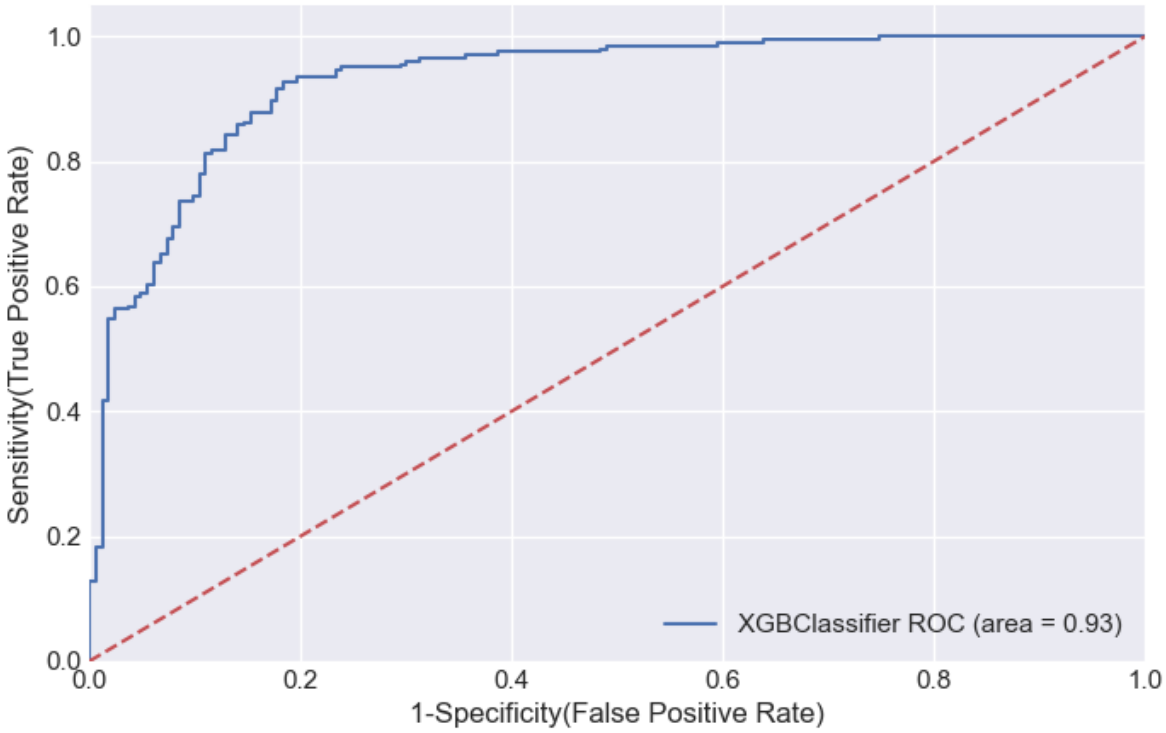
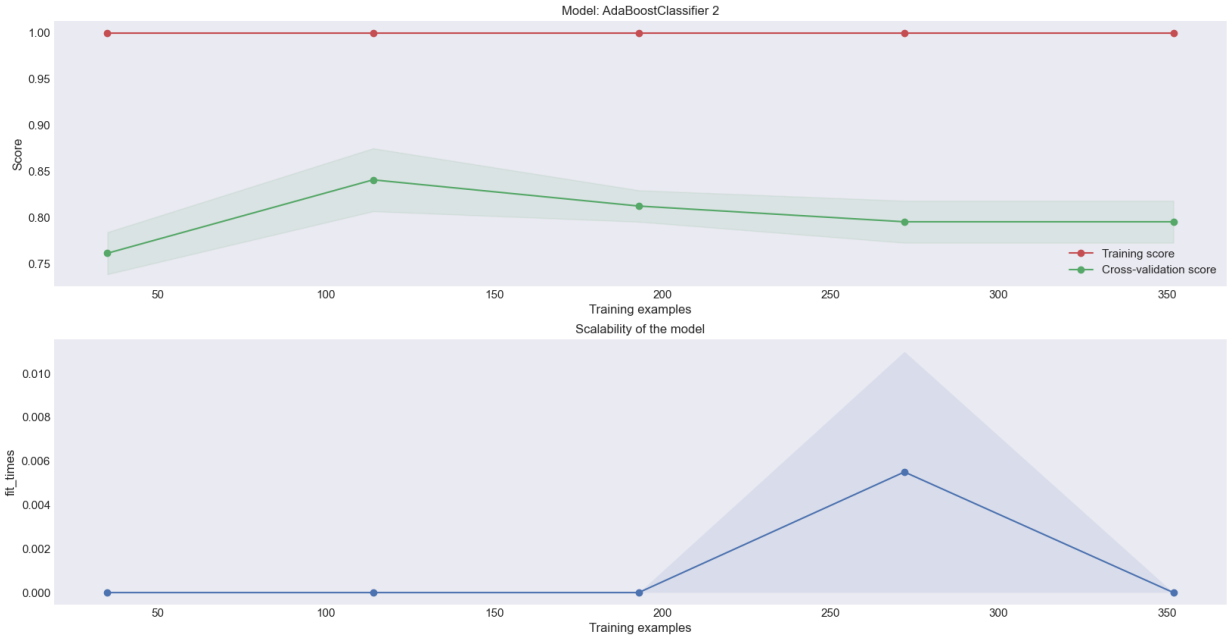


График №90. Score plot



Модель: AdaBoostClassifier 2

The core principle of AdaBoost ("Adaptive Boosting") is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying N weights to each of the training samples. Initially, those weights are all set to 1/N, so that the first step simply trains a weak learner on the original data. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data. At a given step, those training examples that were incorrectly predicted by the boosted model induced at the previous step have their weights increased, whereas the weights are decreased for those that were predicted correctly. As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence. Reference sklearn documentation.

Таблица №30. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.794	0.823	0.808	164.0
class 1	0.853	0.828	0.84	203.0
accuracy	0.826	0.826	0.826	0.826
macro avg	0.823	0.825	0.824	367.0
weighted avg	0.827	0.826	0.826	367.0

График №91. Confusion Matrix

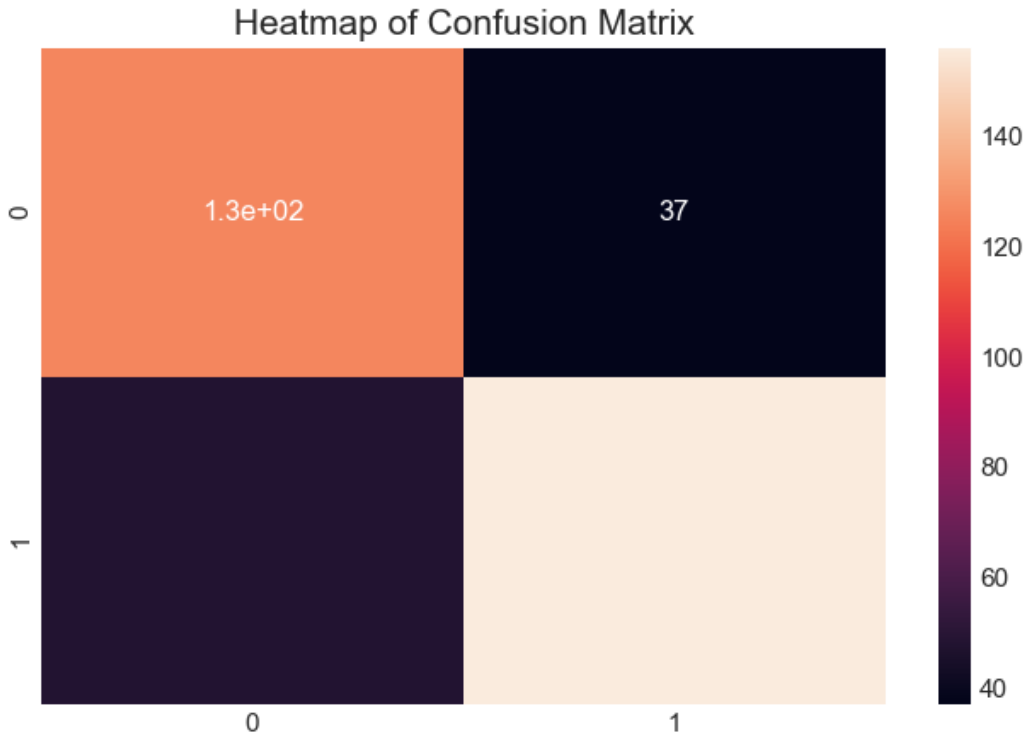


График №92. ROC Curve

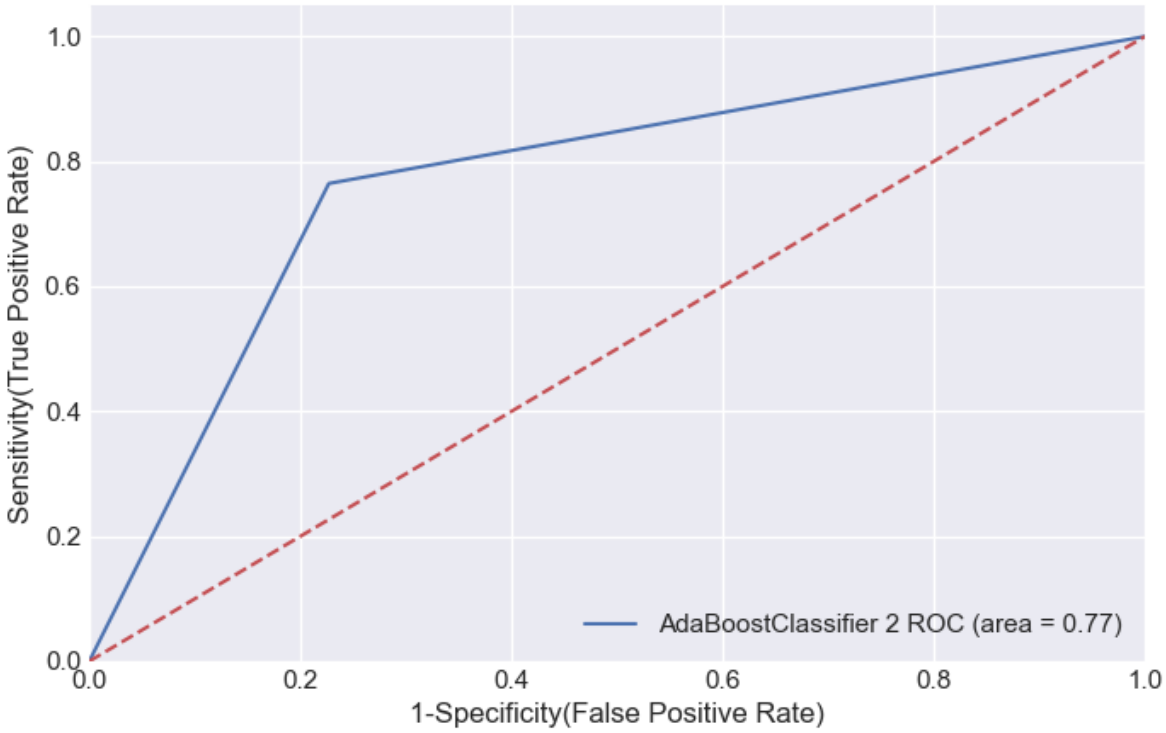
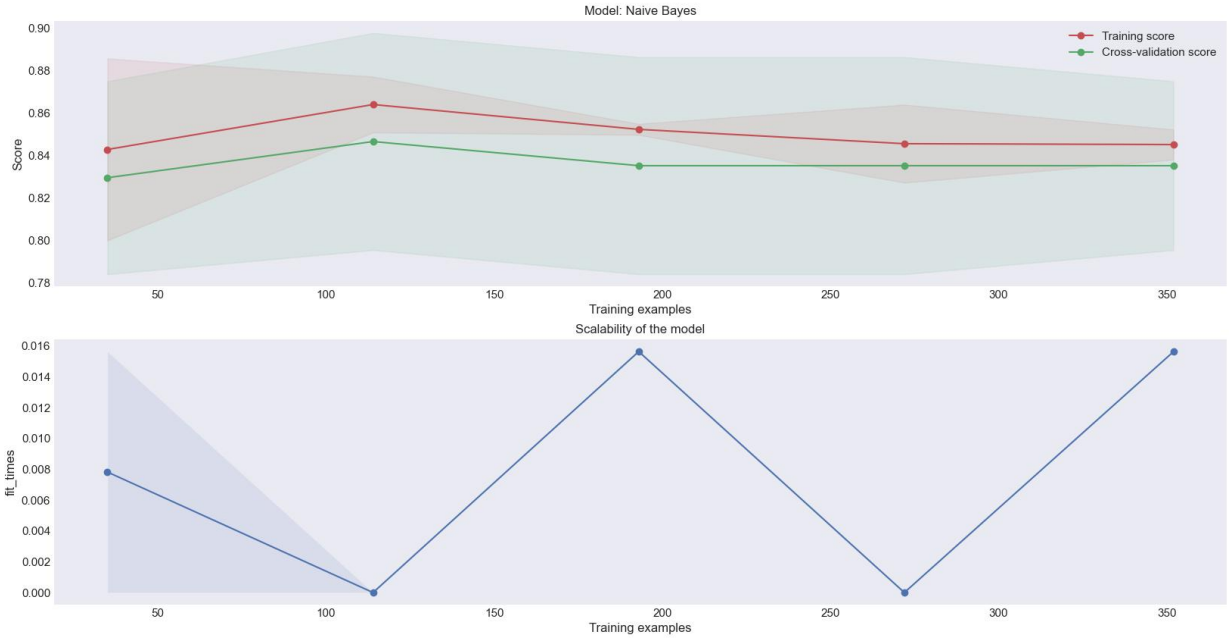


График №93. Score plot



Модель: Naive Bayes

Thanks to https://www.kaggle.com/startupsci/titanic-data-science-solutions
In machine learning, Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features) in a learning problem. Reference Wikipedia.

Таблица №31. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.811	0.835	0.823	164.0
class 1	0.864	0.842	0.853	203.0
accuracy	0.839	0.839	0.839	0.839
macro avg	0.837	0.839	0.838	367.0
weighted avg	0.84	0.839	0.839	367.0
© Dr. Alexander Wagner. Все права охраняются законом				

График №94. Confusion Matrix

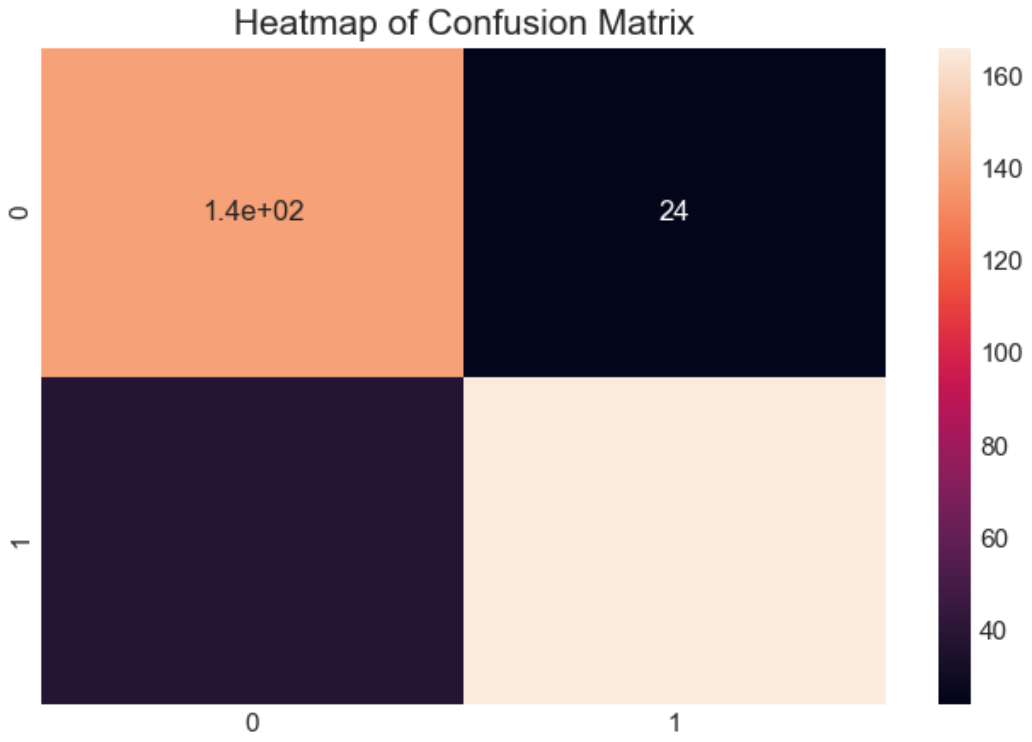


График №95. ROC Curve

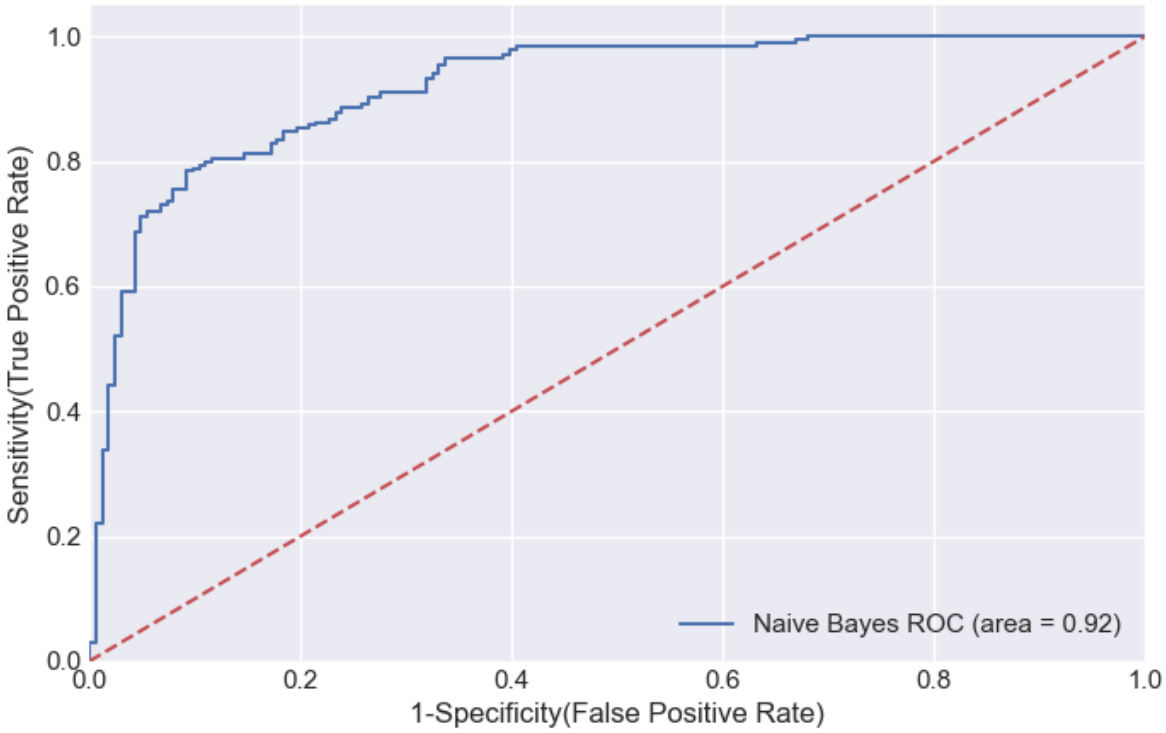
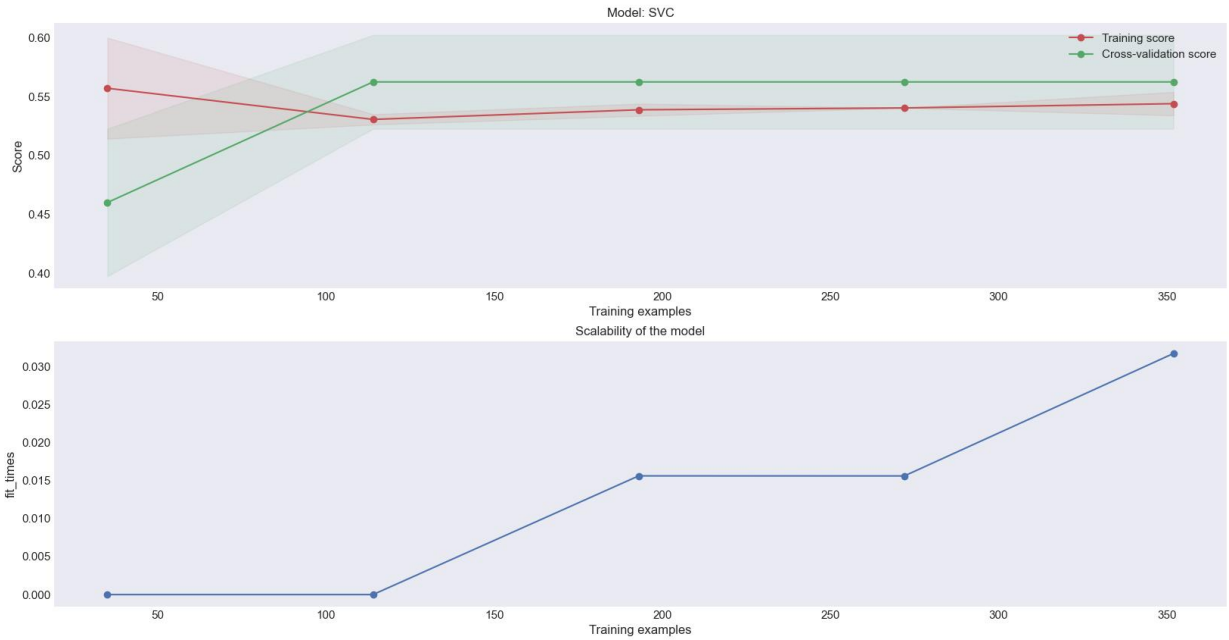


График №96. Score plot



Модель: SVC

Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training samples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new test samples to one category or the other, making it a non-probabilistic binary linear classifier. Reference Wikipedia.

Таблица №32. Таблица классификации

Classes+Metrics	*precision*	*recall*	*f1-score*	*support*
class 0	0.0	0.0	0.0	164.0
class 1	0.553	1.0	0.712	203.0
accuracy	0.553	0.553	0.553	0.553
macro avg	0.277	0.5	0.356	367.0
weighted avg	0.306	0.553	0.394	367.0

График №97. Confusion Matrix

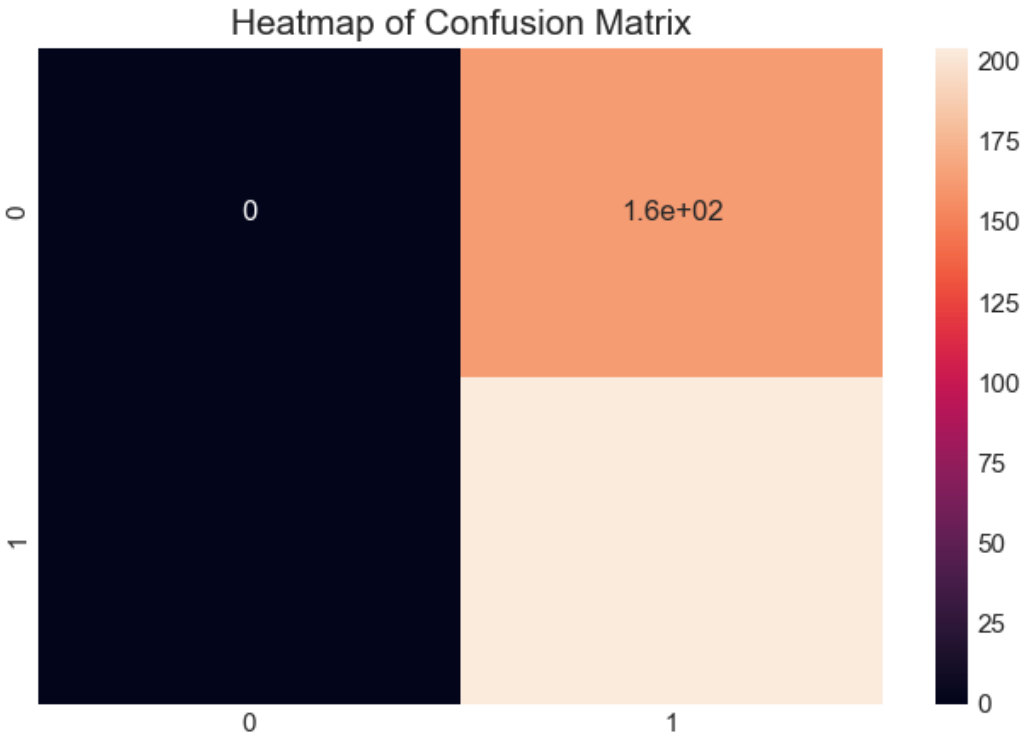


График №98. ROC Curve

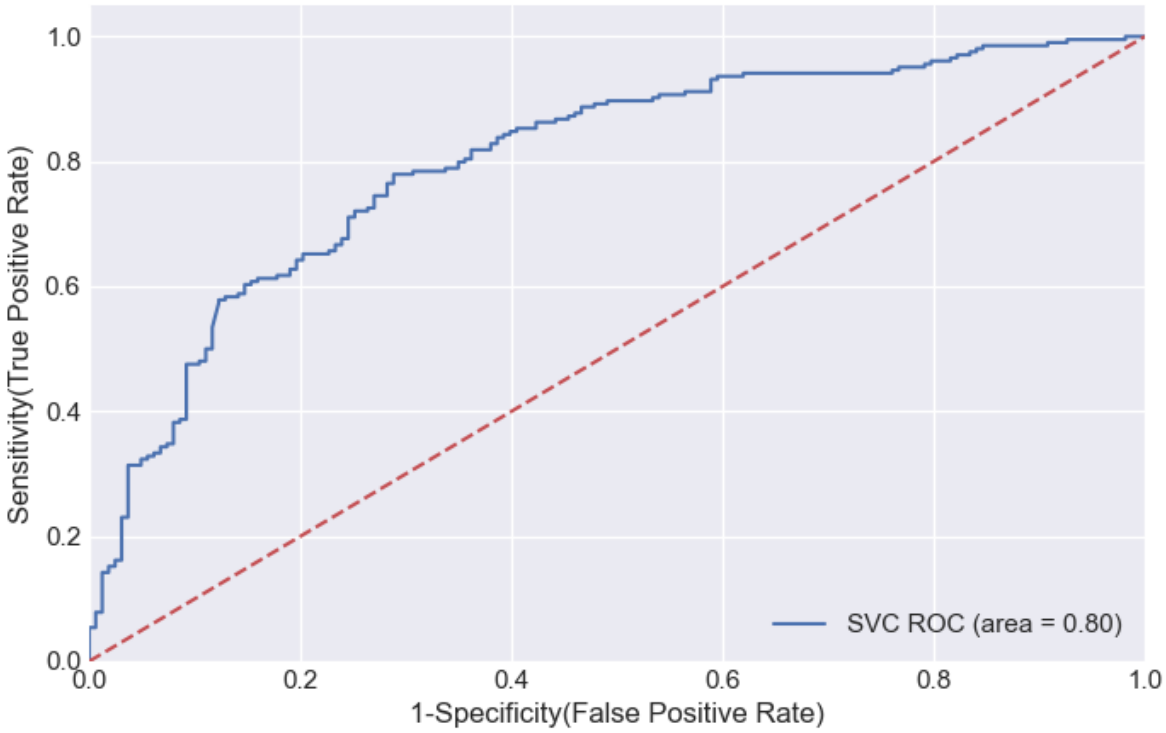
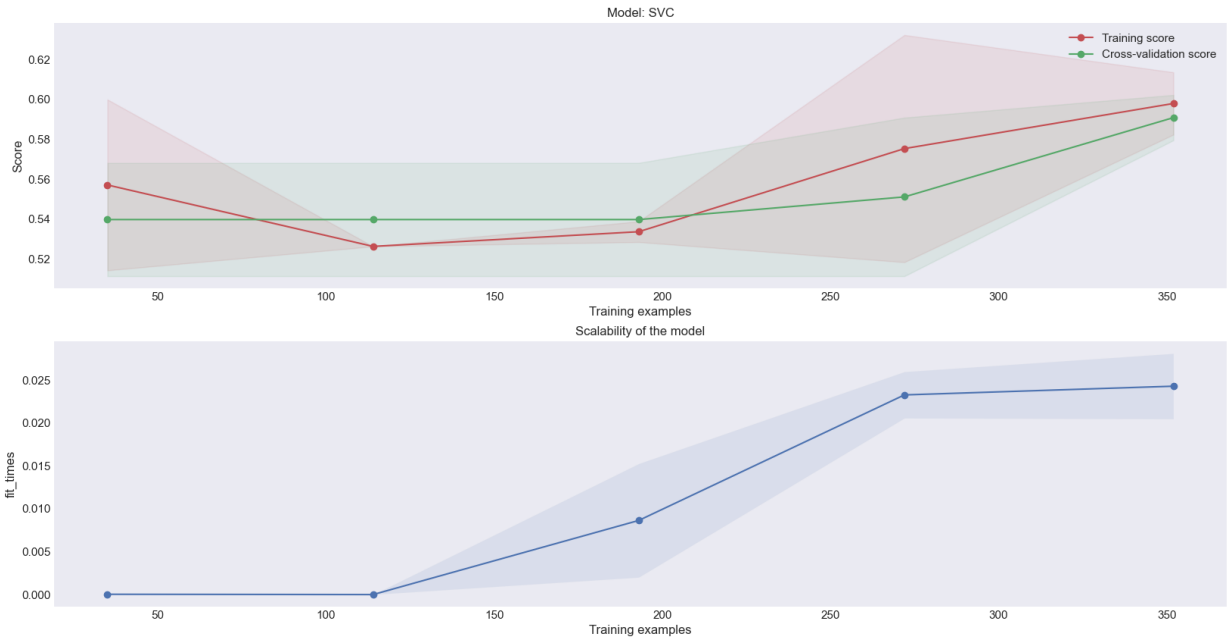


График №99. Score plot



Результаты моделирования

Моделирование осуществлялось при помощи методов машинного обучения. Для исследования проблемы были выбраны 18 моделей в том числе:

- Linear Regression
- Logistic Regression
- Perceptron
- Linear SVC
- MLPClassifier
- Decision Tree Classifier 1
- Stochastic Gradient Decent
- RidgeClassifier
- BaggingClassifier
- AdaBoostClassifier 1
- GradientBoostingClassifie
- KNeighborsClassifier
- DecisionTreeClassifier 2
- RandomForestClassifier
- XGBClassifier
- AdaBoostClassifier 2
- Naive Bayes
- SVC

В результате работы каждой модели полуячена следующая информация в форме таблиц и графиков:

- Таблица классификация
- Матрица Confusion Matrix
- ROC график
- Score график

Таблица классификация даёт оценку модели по критериям:

- ***precision***
- ***recall***
- ***f1-score***

ROC график дапёт оценку модели по критерию AUC, т.е. значению площади под кривой. Что означает – чем ближе этот показатель к значению равному 1, тем выше сответствие прогнозных значений модели реальным входным данным.

Оценка моделей и выбор наилучших для использования в диагности ССЗ

В результате анализа и моделирования получена оценка исходных данных и характкристики для всех моделей. Итоговые результаты представлены в виде таблиц и графиков, объединенный график ROC приведен рис. №100, обобщенные таблицы №33-№35 показывают сравнительные характеристики моделей.

График №100. ROC-график для всех моделей

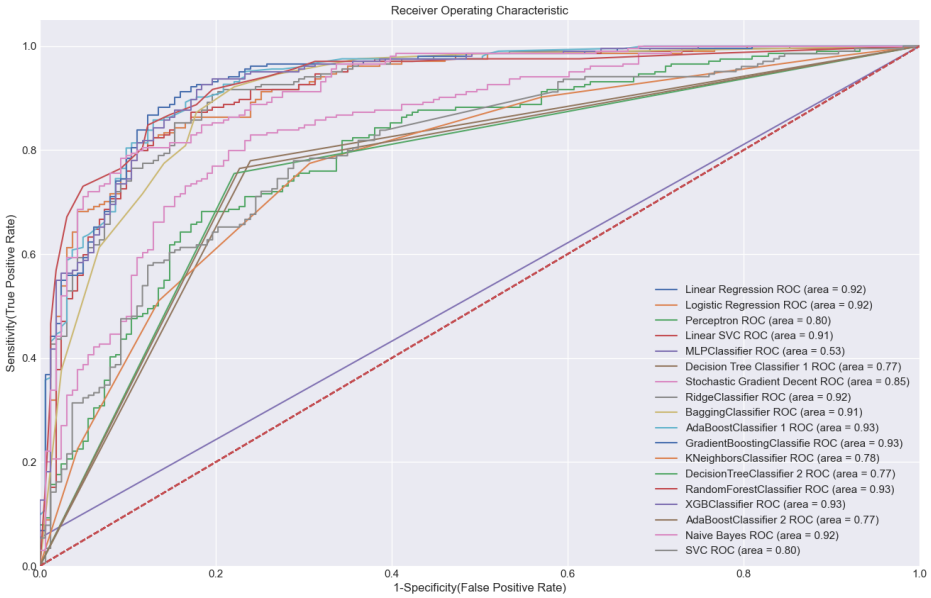


График №101 r2_score %. Линейный график A для всех моделей

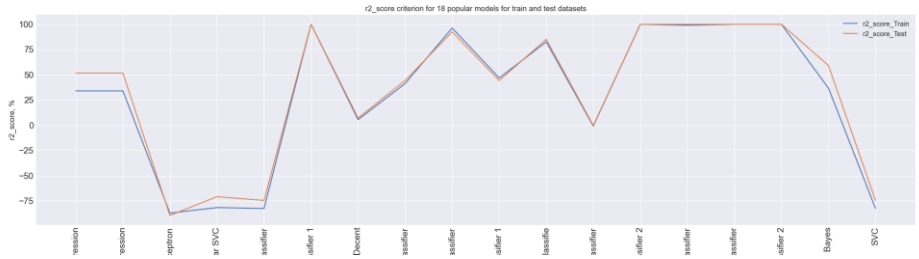


График №102 ACC%. Линейный график B для всех моделей

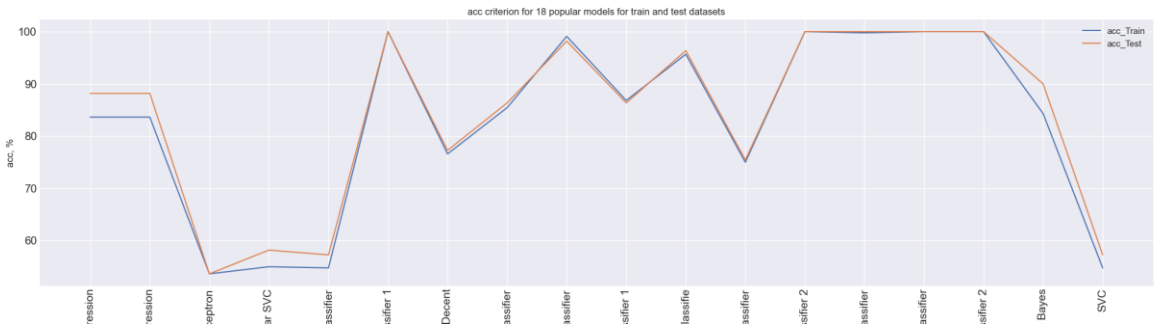


График №103 rmse %. Линейный график C для всех моделей

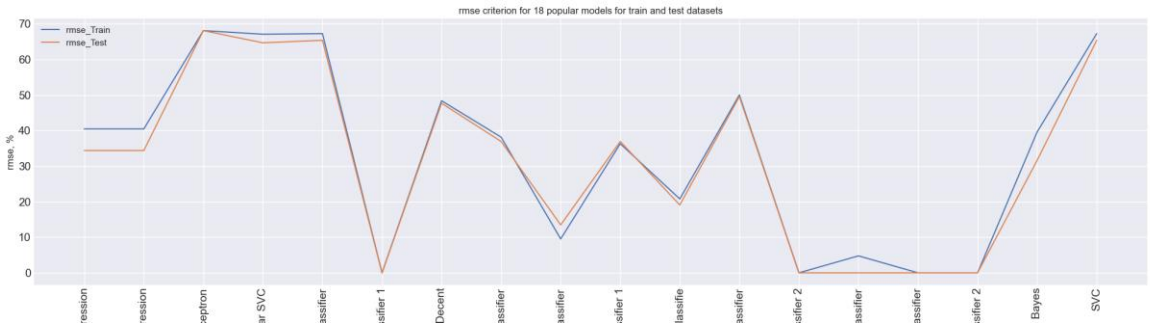


Table №33. Характеристика лучших моделей после первого этапа

Model	r2_sco	r2_score_	acc_Tra	acc_Tes	acc_Diff	rmse_Tr	rmse_	re_Trai	re_Test
Decision Tree	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
DecisionTreeCla	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
XGBClassifier	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
AdaBoostClass2	100.0	100.0	100.0	100.0	0.0	0.0	0.0	0.0	0.0
RandomForestCl	99.08	100.0	99.77	100.0	-	4.77	0.0	0.42	0.0
BaggingClassifie	95.42	88.64	98.86	97.27	1.59000	10.66	16.51	2.09	4.55
GradientBoostin	81.68	88.64	95.45	97.27	-	21.32	16.51	8.37	4.55
AdaBoostClass1	62.45	69.7	90.68	92.73	-	30.53	26.97	17.15	12.12

Table №34. Характеристика всех моделей после второго этапа

Model	acc_train
RandomForestClas	100.0
Decision Tree	100.0
DecisionTreeClass	100.0
BaggingClassifier	99.09
GradientBoosting	95.82
XGBClassifier	91.27
AdaBoostClassifie	91.09
RidgeClassifier	85.09
Logistic	84.73
Linear Regression	84.18
KNeighborsClassif	78.18
Linear SVC	55.45
SVC	55.45
Perceptron	54.36
Stochastic	49.27
MLPClassifier	44.55

Table №35. Характеристика всех моделей после третьего этапа

Model	r2_score_train	acc_train	rmse_train	re_train
RandomForestClas	100.0	100.0	0.0	0.0
Decision Tree	100.0	100.0	0.0	0.0
DecisionTreeClass	100.0	100.0	0.0	0.0

Model	*r2_score_train*	*acc_train*	*rmse_train*	*re_train*
BaggingClassifier	96.32	99.09	9.53	1.64
GradientBoosting	83.07	95.82	20.45	7.54
XGBClassifier	64.67	91.27	29.54	15.74
AdaBoostClassifie	63.93	91.09	29.85	16.07
RidgeClassifier	39.65	85.09	38.61	26.89
Logistic	38.17	84.73	39.08	27.54
Linear Regression	35.97	84.18	39.77	28.52
KNeighborsClassif	11.68	78.18	46.71	39.34
Linear SVC	-80.33	55.45	66.74	80.33
SVC	-80.33	55.45	66.74	80.33
Perceptron	-84.74	54.36	67.55	82.3
Stochastic	-105.35	49.27	71.22	91.48
MLPClassifier	-124.49	44.55	74.47	100.0

График №104. График основных метрик для лучших моделей: AUC, F1, Precision, Accuracy_Test

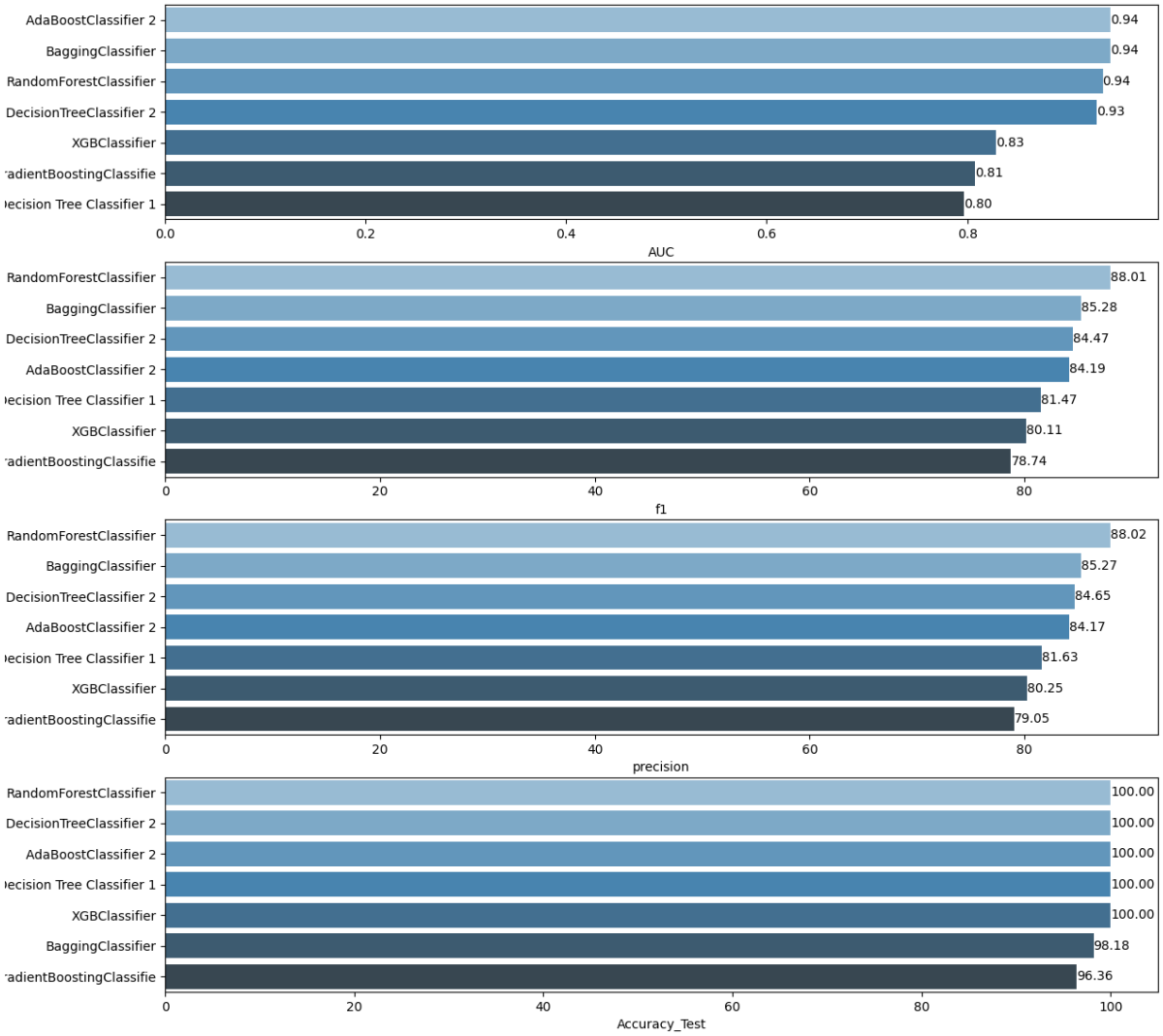
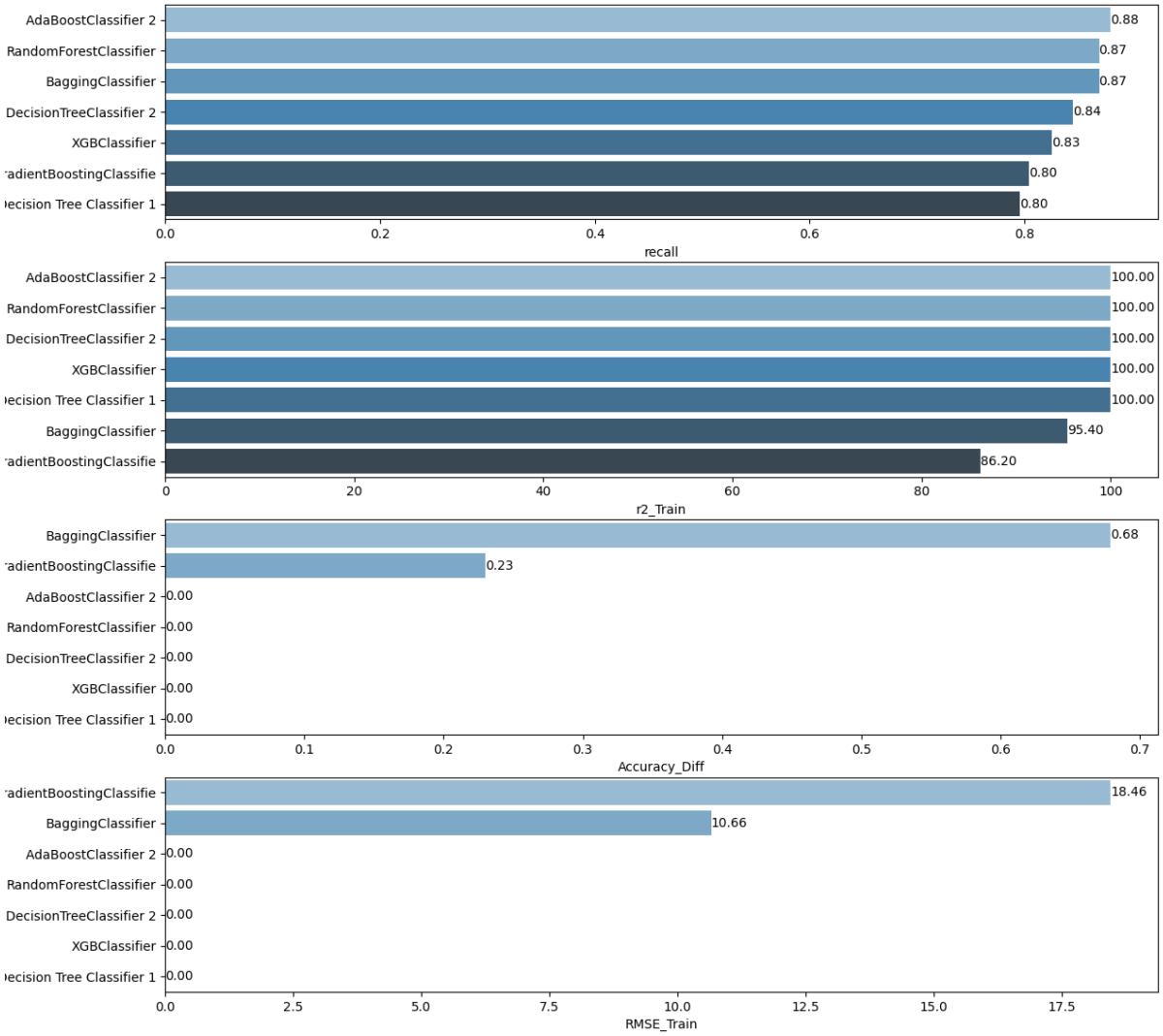


График №105. График основных метрик для лучших моделей: Recall, r2_Train, Accuracy_Diff, RMSE_Train



Заключение

Системы и модели поддержки принятия решений на основе машинного обучения для раннего прогнозирования и диагностики находят широкое применение в здравоохранении. Эти системы помогают пациентам и медицинскому персоналу улучшить процесс принятия решений и раннее прогнозирование возникновения ССЗ у кардио-пациентов. По сравнению с другими известными алгоритмами и системами прогнозирования, мы обнаружили, что алгоритмы машинного обучения лучше работают в прогнозировании и диагностике ССЗ у кардио-пациентов.

Лучшими алгоритмами машинного обучения в нашем исследовании стали:

1. Decision Tree Classifier 1
2. DecisionTreeClassifier 2
3. RandomForestClassifier
4. XGBClassifier
5. AdaBoostClassifier 2
6. BaggingClassifier
7. GradientBoostingClassifie

Другие модели прогнозирования, основанные на машинном обучении, также были протестированы, но они показали худшие результаты, а их точность была меньше, чем у этих моделей, поэтому эти 7 моделей были доработаны в нашем исследования и рекомендуются к применению к использованию или проверки в аналогичных исследованиях.

В данной статье мы применили алгоритмы машинного обучения для раннего прогнозирования ССЗ у кардио-пациентов. Производительность этих моделей была сравнена с нашей моделью ансамбля мягкого голосования на основе машинного обучения. По результатам эксперимента мы обнаружили, что производительность нашего классификатора ансамбля мягкого голосования превзошла производительность других моделей машинного обучения. Кроме того, прогностические факторы для классификатора ансамбля мягкого голосования отличались от регрессионных моделей.