VR Protect

Статический анализ исполняемых файлов на содержание вредоносного кода

Н. В. Щелкунова
Вычислительные решения

к.т.н., **Д. А. Щелкунов**
Рекрипт

О проблеме

Число вредоносных
программ постоянно растёт

Новое ВПО
и семейства ВПО

Крипторы

Обфускаторы

Нужны эффективные алгоритмы
эвристического статического и динамического анализа ПО
с высокой достоверностью результата

Эвристический статический анализ

Анализ структуры, состава, свойств исследуемых сущностей

Ручной

Разработчики сами пишут правила классификации

Часто встречается в продуктах ИБ “старых” поколений

С использованием ML и ИИ

Тренд последних лет: качество выше, трудозатраты ниже

Цели работы

Создать и обучить несколько моделей
для статического анализа ПО PE-формата

Оценить ресурсоёмкость и эффективность
каждой из моделей

Провести сравнительный
анализ моделей

Результаты

В данной работе мы создали и обучили модели, способные (на нашем датасете):

выявить ВПО
с вероятностью не менее **95%**

классифицировать ВПО
с вероятностью не менее **90%**

Анализ и подготовка данных

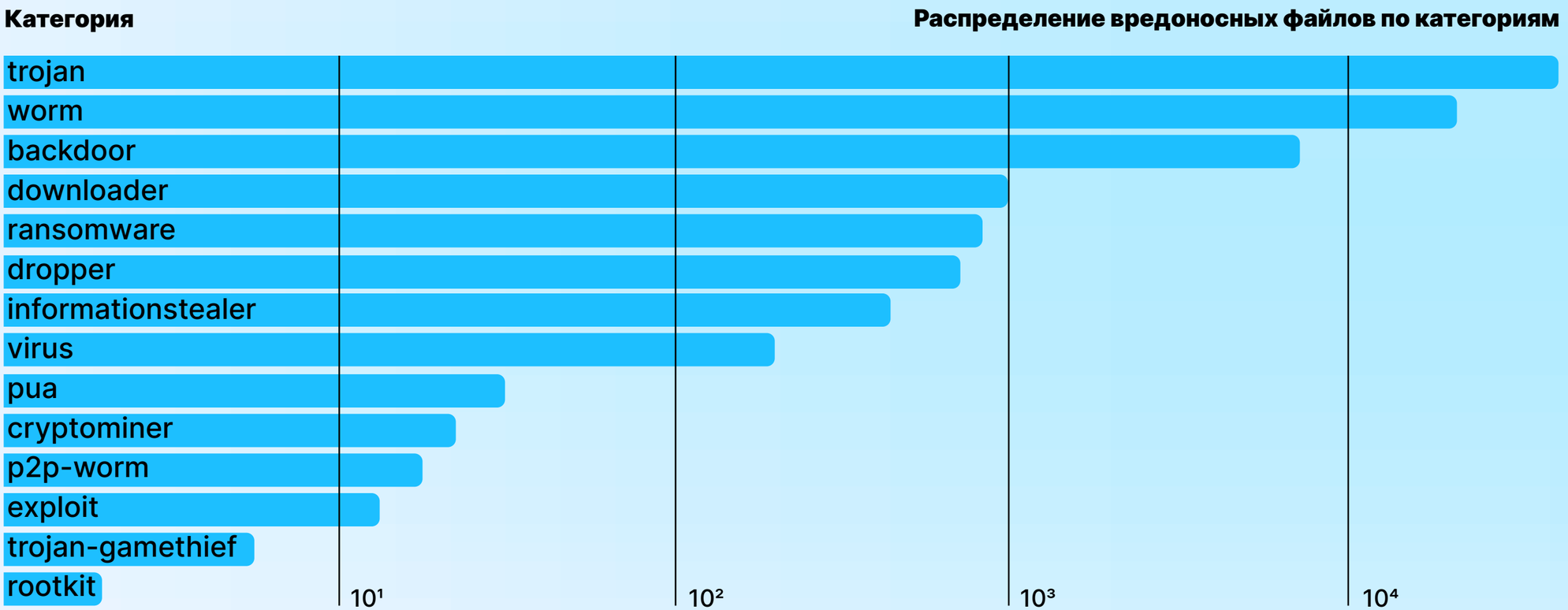
В процессе обучения использовались как открытые датасеты, так и собранные нами данные. Анализировалось ВПО со снятыми упаковщиками. Данные представлены в виде значений признаков PE-файлов, всего 2381 признак. Ниже приведены некоторые из них:

entropy of the byte histogram
occurrences of the string "c:\" (ignore case)
occurrences of "http://" or "https://". (ignore case)
occurrences of the string prefix "HKEY_"
occurrences of "MZ"

length of the bytes of the file
virtual size of the lief parsed binary
whether the binary has a debug section
# of exported functions
# of imported functions

Для классификации на содержание вредоносного кода датасет достаточно сбалансирован – 80 тыс. вредоносных/ 100 тыс. безопасных образцов.

Анализ и подготовка данных



Статический анализ исполняемых файлов на содержание вредоносного кода

Модели

Модели на основе алгоритмов машинного обучения:

- Elastic Net
- Random Forest
- KNN

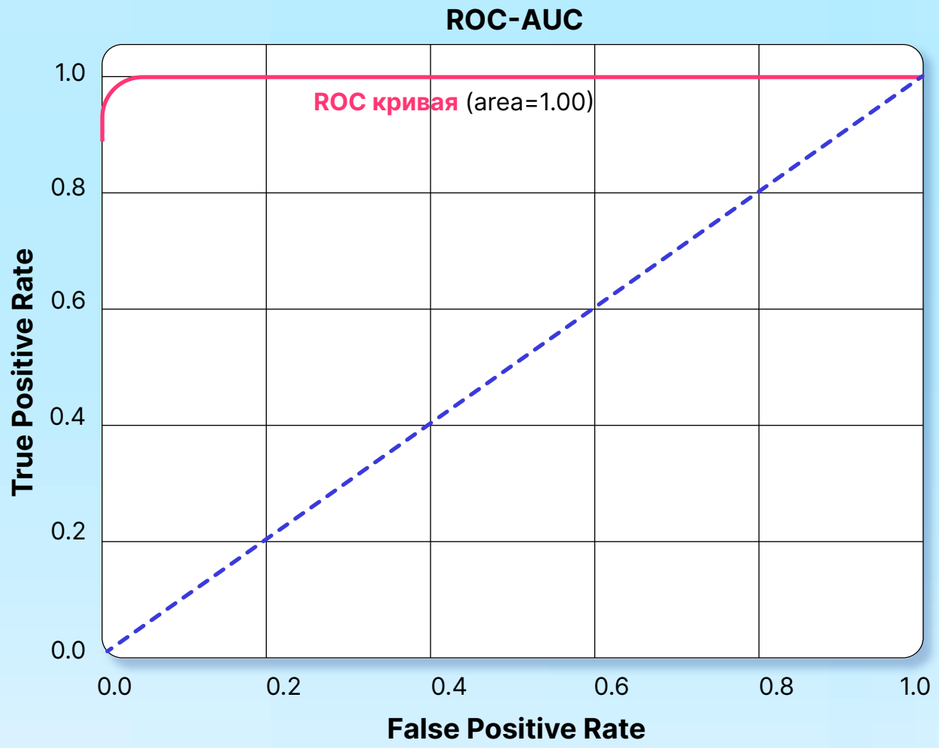
Глубокая сеть прямого распространения

Модели машинного обучения. Elastic Net

Бинарная классификация (на содержание вредоносного кода)

Параметры:
α = 0.015
l1_ratio = 0.025

Результаты:
MSE = 0.0376
MAE = 0.124
ROC AUC = 0.998
Размер модели = 9.84 kB
Время обучения = 1.76 min
Время прогноза для одного примера = 24.2 µs ± 147 ns



Модели машинного обучения. Elastic Net

Многоклассовая классификация (определение категории вредоносного кода)

Параметры:

α = 0.0015
l1_ratio = 0.18
max_iter = 5000

Результаты:

MSE = 3.428
MAE = 0.935
Время прогноза для одного примера = 25.2 µs ± 499 ns
Размер модели = 9.8 kB
Время обучения = 1.59 min

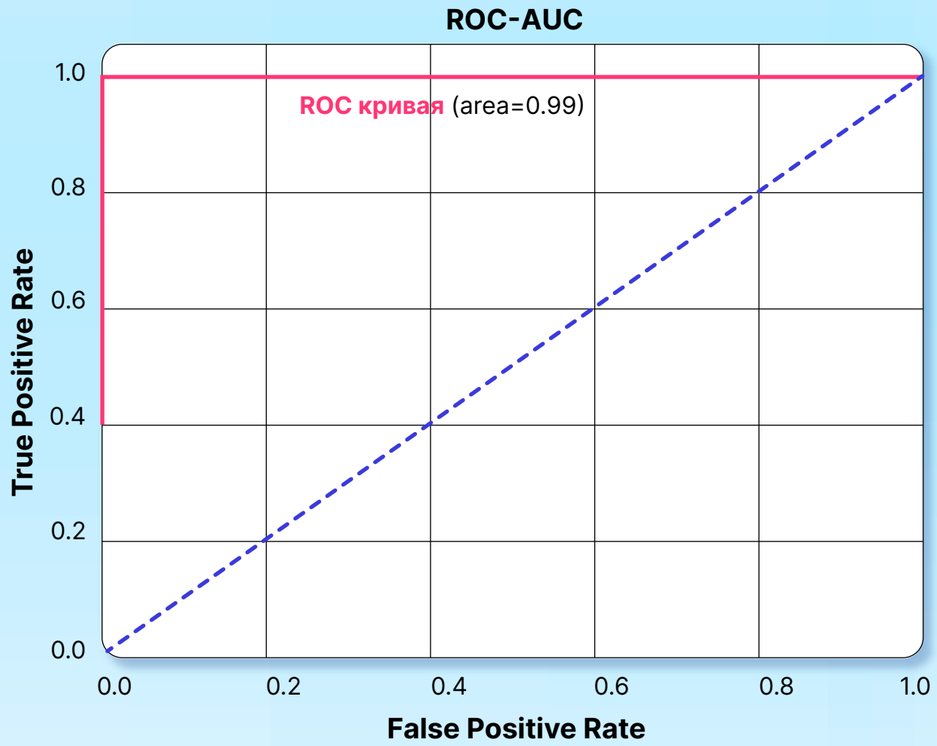
Модели машинного обучения. Random Forest, base model

Бинарная классификация

Параметры:
n_estimators (количество деревьев) = 500
random_state = 42

Результаты:
accuracy = 0.9953
f1 = 0.9932
ROC AUC = 0.994
Размер модели = 90.4 MB
Время обучения = 1.89 min

Время прогноза для одного примера = 24.3 ms ± 250 µs



Модели машинного обучения. Random Forest, base model

Многоклассовая классификация

Параметры:

n_estimators
(количество деревьев)
 = 500

random_state = 42

Результаты:

accuracy = 0.9449
f1 = 0.9399
Размер модели = 352 MB
Время обучения = 39.69 s

Время прогноза
для одного примера
= 25,2 ms ± 274 µs

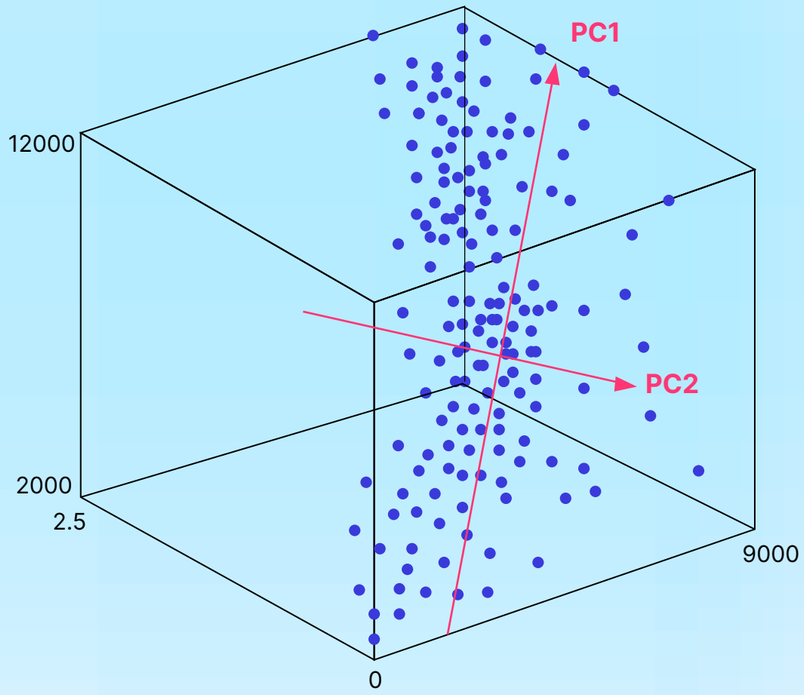
Модели машинного обучения. Random Forest, PCA

Random Forest

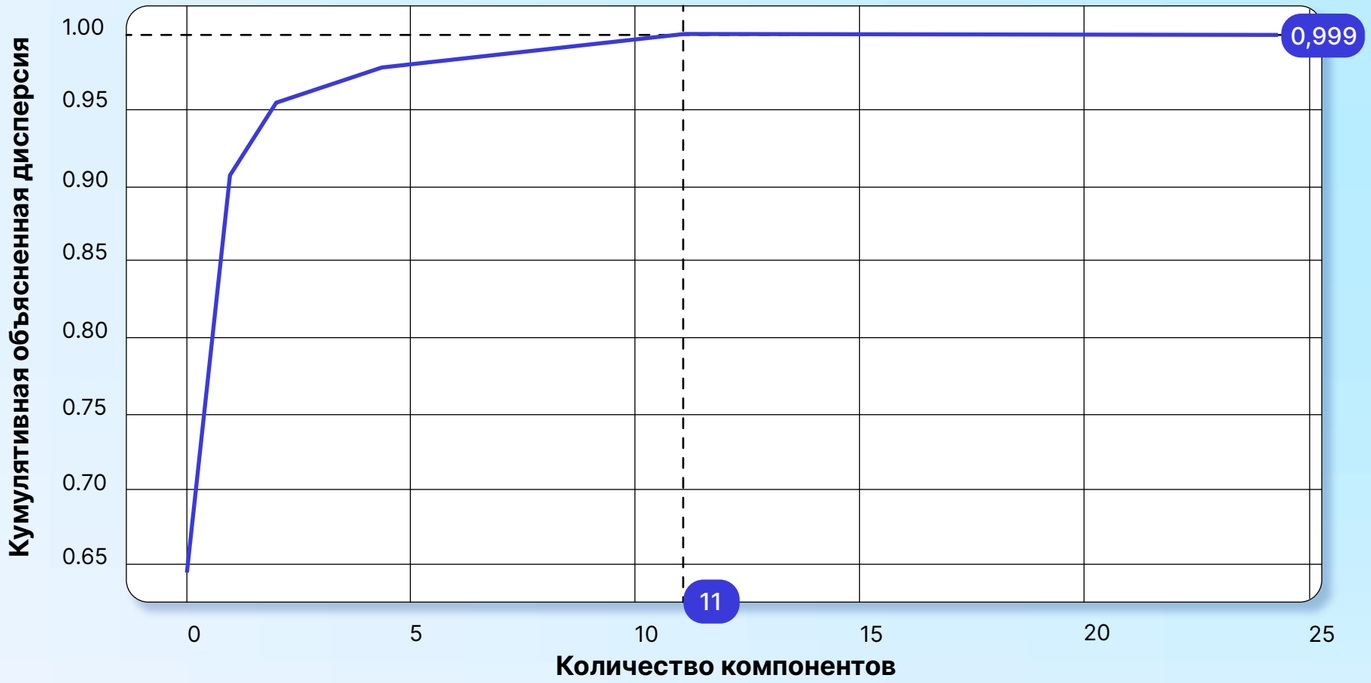
Основной минус Random Forest заключается в объёмности вычислений, поэтому мы провели исследования по использованию алгоритма PCA (principal component analysis).

RCA

В основе PCA лежит идея нахождения новых признаков, называемых главными компонентами, которые максимально коррелируют с исходными данными и при этом ортогональны друг другу. Эти главные компоненты формируют новый базис в пространстве признаков, исключая лишнюю информацию и снижая размерность.



Модели машинного обучения. Random Forest, PCA



Бинарная классификация (на содержание вредоносного кода)

Зависимость кумулятивной объясненной дисперсии от количества компонентов

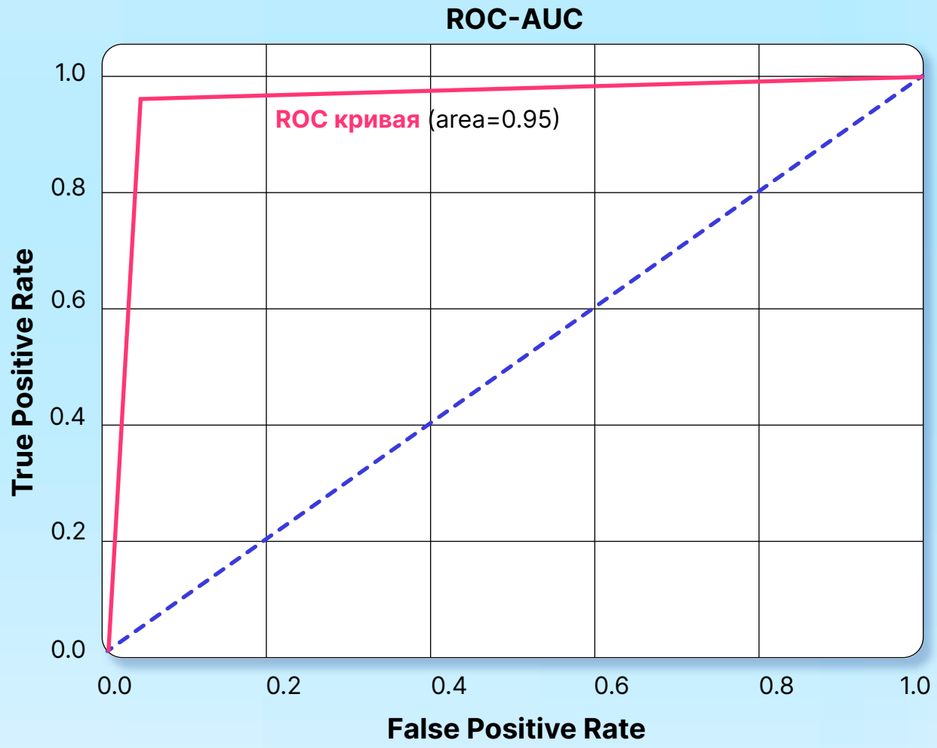
Модели машинного обучения. Random Forest, PCA

Бинарная классификация

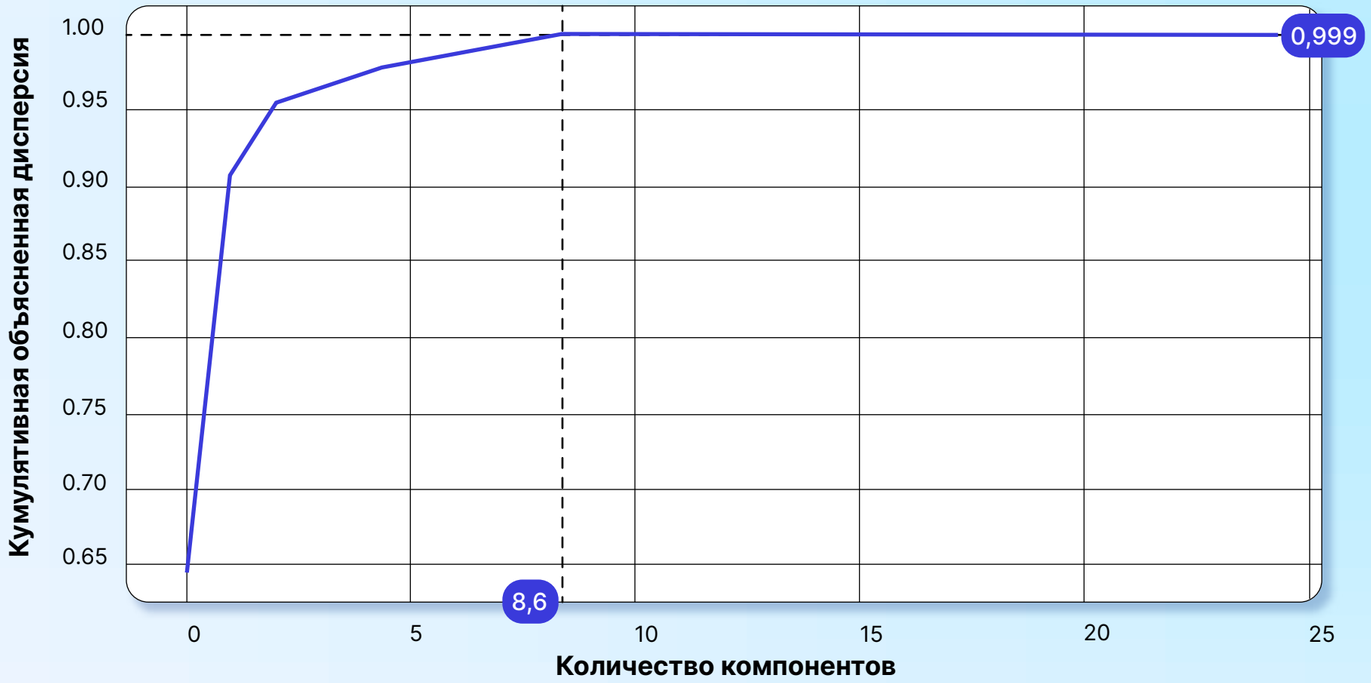
Параметры:
n_estimators (количество деревьев) = 500
random_state = 42

Результаты:
accuracy = 0.953
f1 = 0.949
ROC AUC = 0.951
Размер модели = 122 MB
Время обучения = 9 s

Время прогноза
для одного примера
42.4 ms ± 179 µs



Модели машинного обучения. Random Forest, PCA



Многоклассовая классификация (определение категории вредоносного кода)

Зависимость кумулятивной объясненной дисперсии от количества компонентов

Модели машинного обучения. Random Forest, PCA

Многоклассовая классификация

Параметры:

n_estimators
(количество деревьев)
= 500

random_state = 42

Результаты:

accuracy = 0.9078
f1 = 0.907
Размер модели = 233 MB
Время обучения = 1.191 s

Время прогноза
для одного примера
= 22.9 ms ±149 µs

Модели машинного обучения. RF. Оптимизация гиперпараметров

Бинарная классификация

Гиперпараметры:

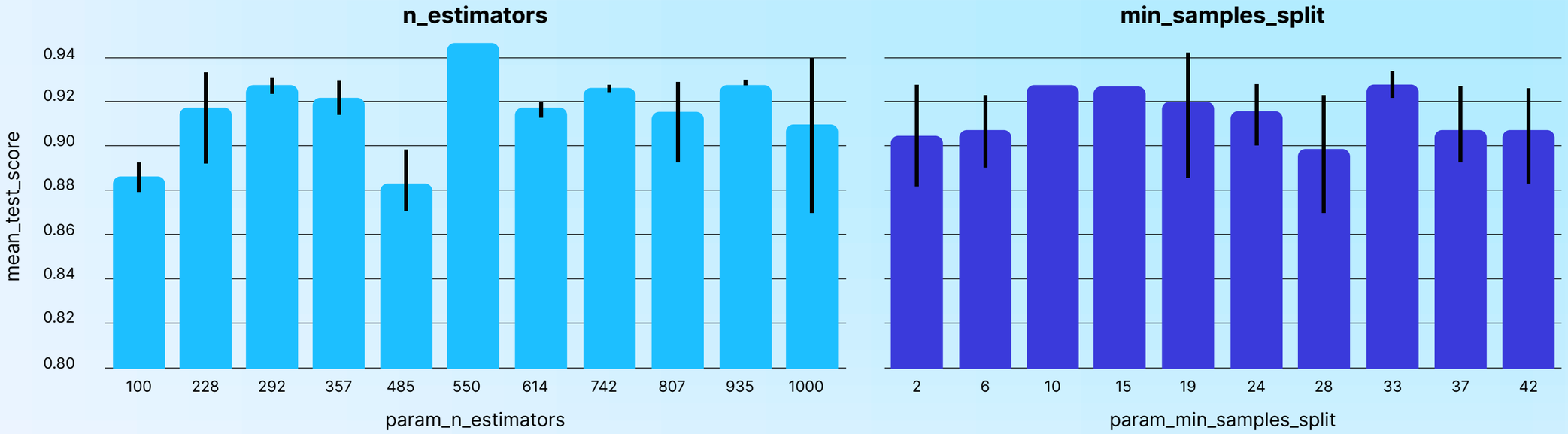
1 этап – RandomizedSearchCV

- n_estimators (число деревьев) - 12 значений из интервала [100,1000]
- max_features (число признаков для выбора расщепления) = [log2, sqrt]
- max_depth (максимальная глубина деревьев) - 15 значений из интервала [5,20]
- min_samples_split (минимальное число объектов, необходимое для того, чтобы узел дерева мог бы расщепиться) - 10 значений из интервала [2,40]
- min_samples_leaf (минимальное число объектов в листьях) - 10 значений из интервала [2,40]
- bootstrap (использование для построения деревьев подвыборки с возвращением) = [True, False]

При значениях параметров n_iter = 30 и cv = 3 было создано 90 RF-моделей из случайных комбинаций представленных выше гиперпараметров.

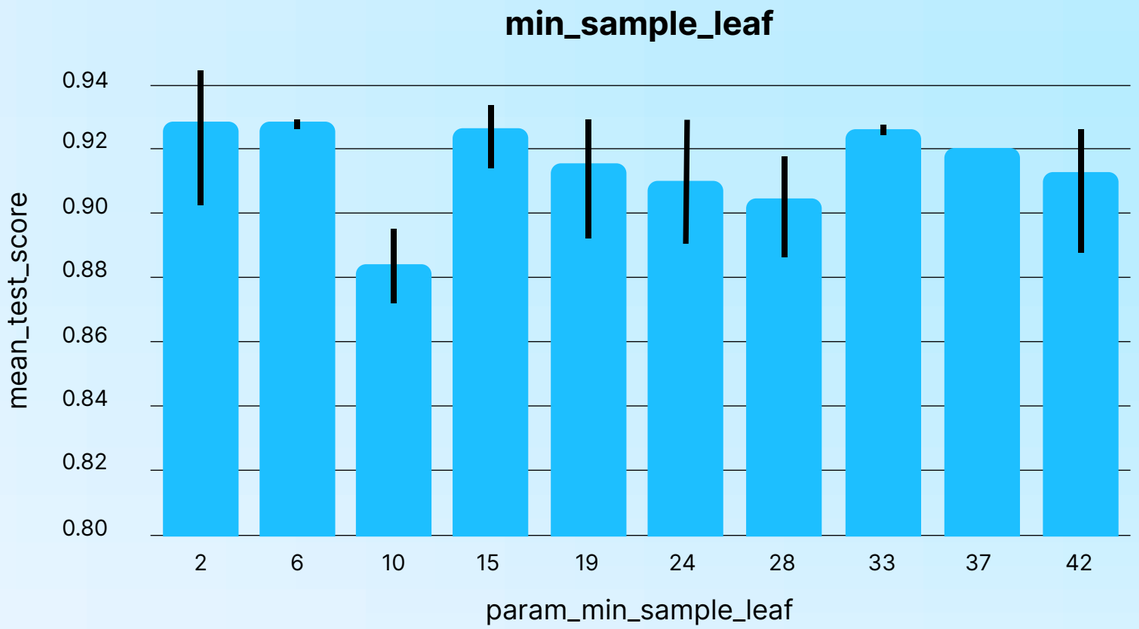
Модели машинного обучения. RF. Оптимизация гиперпараметров

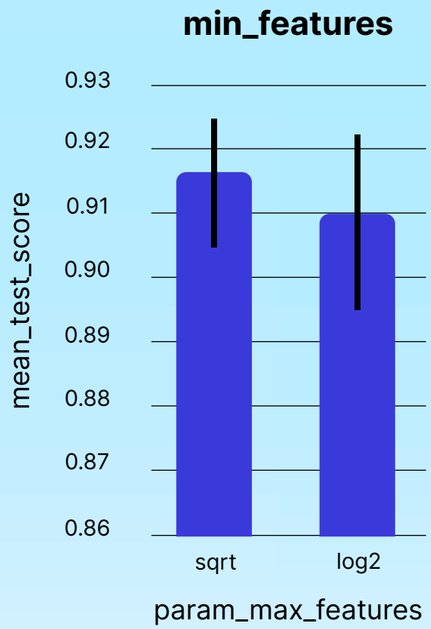
Бинарная классификация



Модели машинного обучения. RF. Оптимизация гиперпараметров

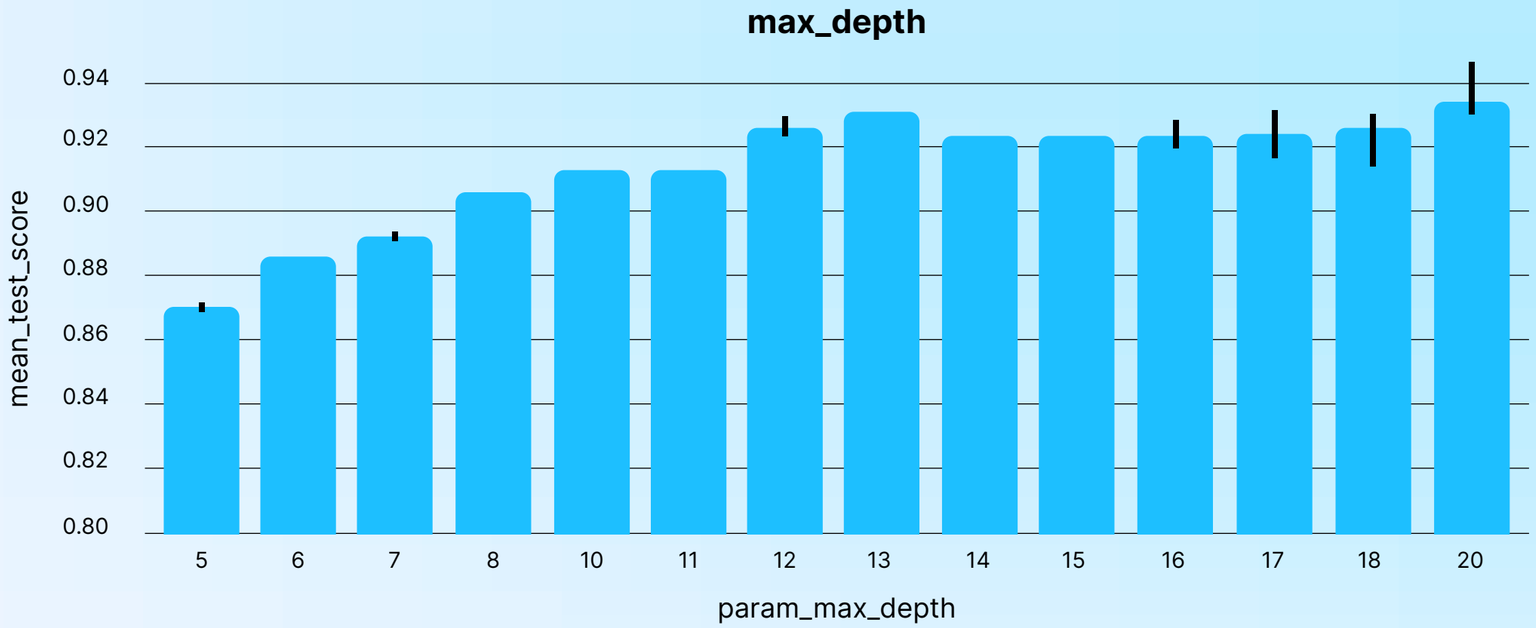
Бинарная классификация

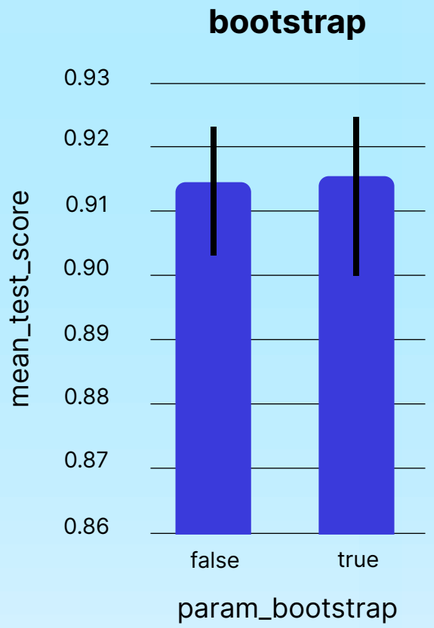




Модели машинного обучения. RF. Оптимизация гиперпараметров

Бинарная классификация





Модели машинного обучения. RF. Оптимизация гиперпараметров

Бинарная классификация

Здесь мы применяем кросс-валидацию по 3 блокам для 96 (3 × 2 × 2 × 2 × 2 × 2) сеансов обучения модели, что даёт 288 сеансов обучения модели.

2 этап – GridSearchCV

Лучшие гиперпараметры:

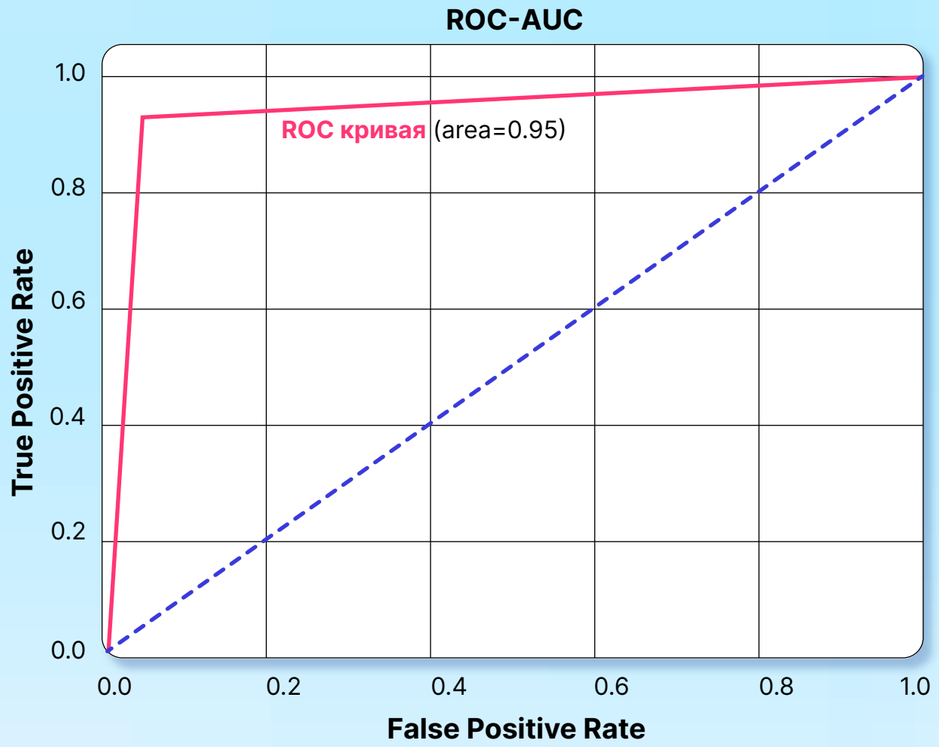
- Bootstrap = False,
- max_depth = 20,
- max_features = 'sqrt',
- min_samples_leaf = 2,
- min_samples_split = 19,
- n_estimators = 1000

Модели машинного обучения. RF. Оптимизация гиперпараметров

Бинарная классификация

Результаты:

accuracy = 0.95
f1 = 0.9405
ROC AUC = 0.948
Размер модели = 209.7 MB
Время обучения = 2.76 min
Время прогноза для одного примера = 80.3 ms ± 487 µs



Модели машинного обучения. RF. Оптимизация гиперпараметров

Многоклассовая классификация

Гиперпараметры:

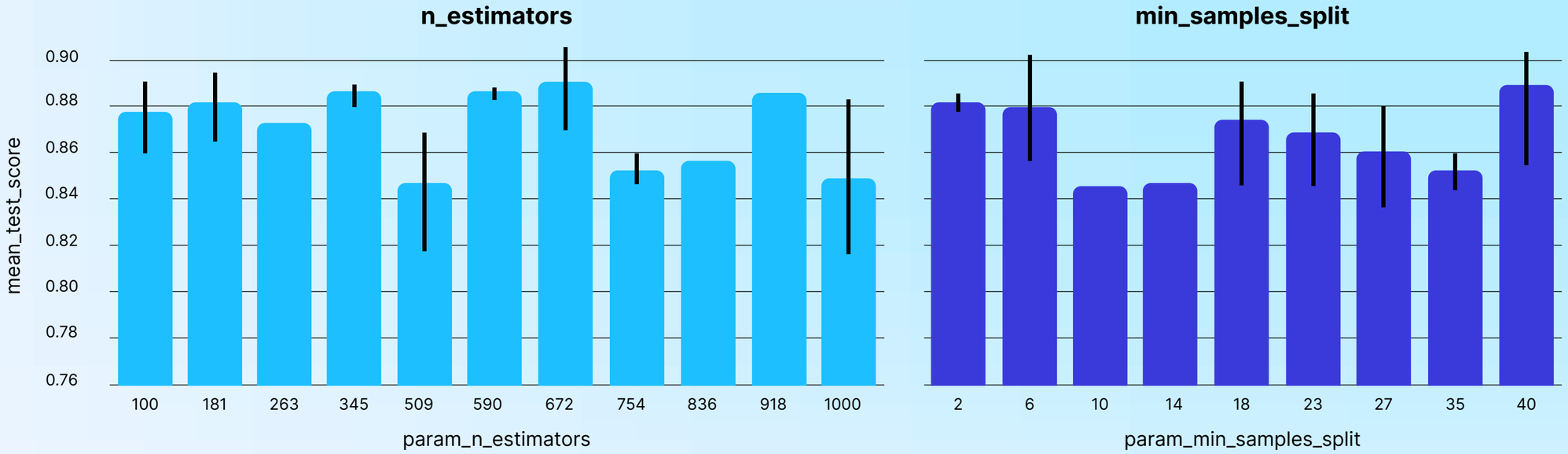
1 этап – RandomizedSearchCV

- n_estimators (число деревьев) - 10 значений из интервала [100,1000]
- max_features (число признаков для выбора расщепления) = [log2, sqrt]
- max_depth (максимальная глубина деревьев) - 15 значений из интервала [5,20]
- min_samples_split (минимальное число объектов, необходимое для того, чтобы узел дерева мог бы расщепиться) - 10 значений из интервала [2,40]
- min_samples_leaf (минимальное число объектов в листьях) - 10 значений из интервала [2,40]
- bootstrap (использование для построения деревьев подвыборки с возвращением) = [True, False]

При значениях параметров n_iter = 30 и cv = 3 было создано 90 RF-моделей из случайных комбинаций представленных выше гиперпараметров.

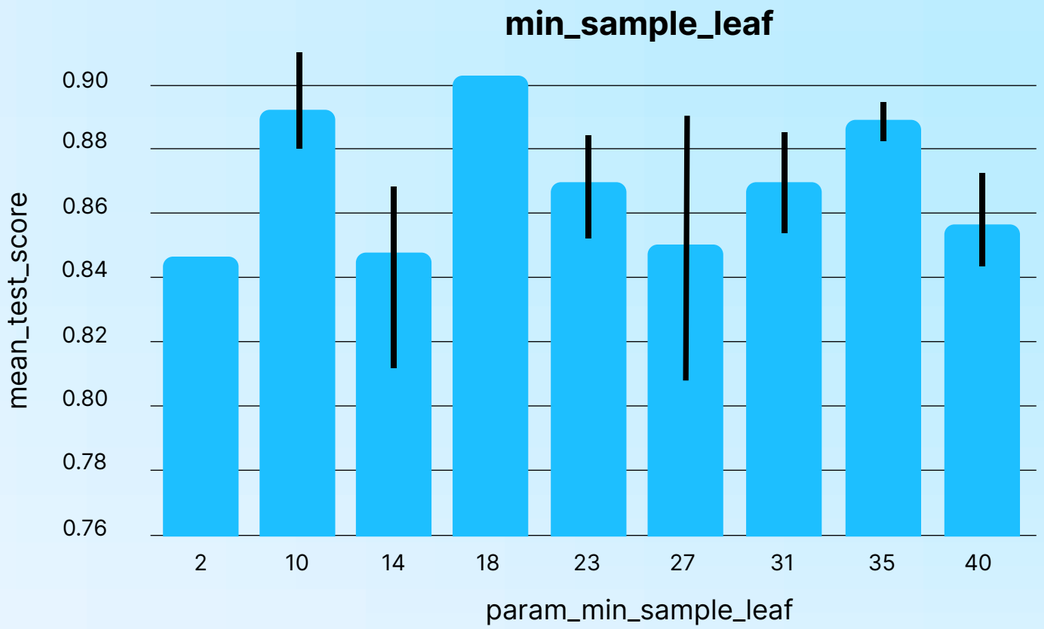
Модели машинного обучения. RF. Оптимизация гиперпараметров

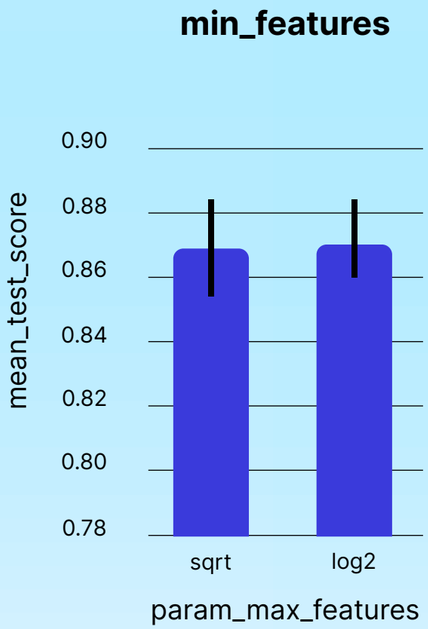
Многоклассовая классификация



Модели машинного обучения. RF. Оптимизация гиперпараметров

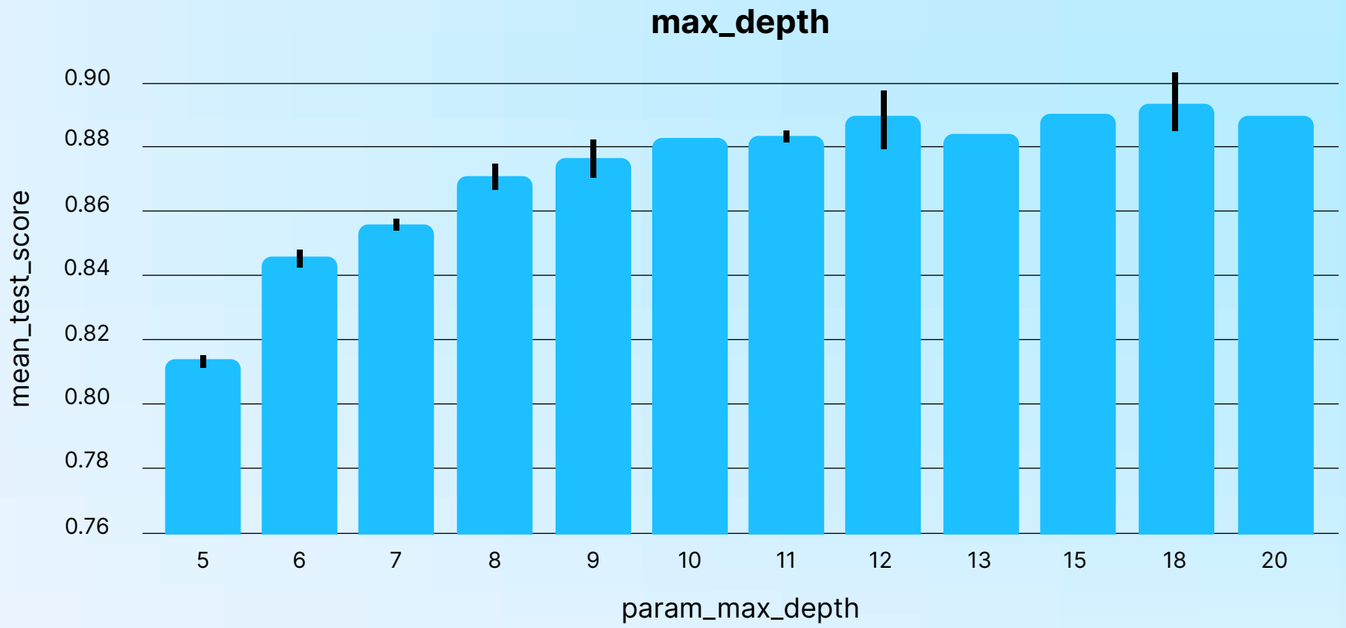
Многоклассовая классификация

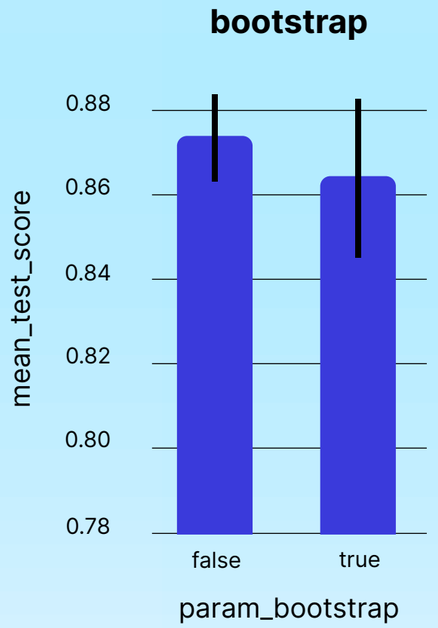




Модели машинного обучения. RF. Оптимизация гиперпараметров

Многоклассовая классификация





Модели машинного обучения. RF. Оптимизация гиперпараметров

Многоклассовая классификация

2 этап – GridSearchCV

Здесь мы также применяем кросс-валидацию по 3 блокам для 96 (3 × 2 × 2 × 2 × 2 × 2) сеансов обучения модели, что даёт 288 сеансов обучения модели.

Лучшие гиперпараметры:

- Bootstrap = False,
- max_depth = 18,
- max_features = 'sqrt',
- min_samples_leaf = 10,
- min_samples_split = 6,
- n_estimators = 672

Результаты:

accuracy = 0.9082

f1 = 0.9048

Время прогноза для одного примера = 31 ms ± 769 µs

Размер модели = 164.5 MB

Время обучения = 23.589 s

Модели машинного обучения. KNN

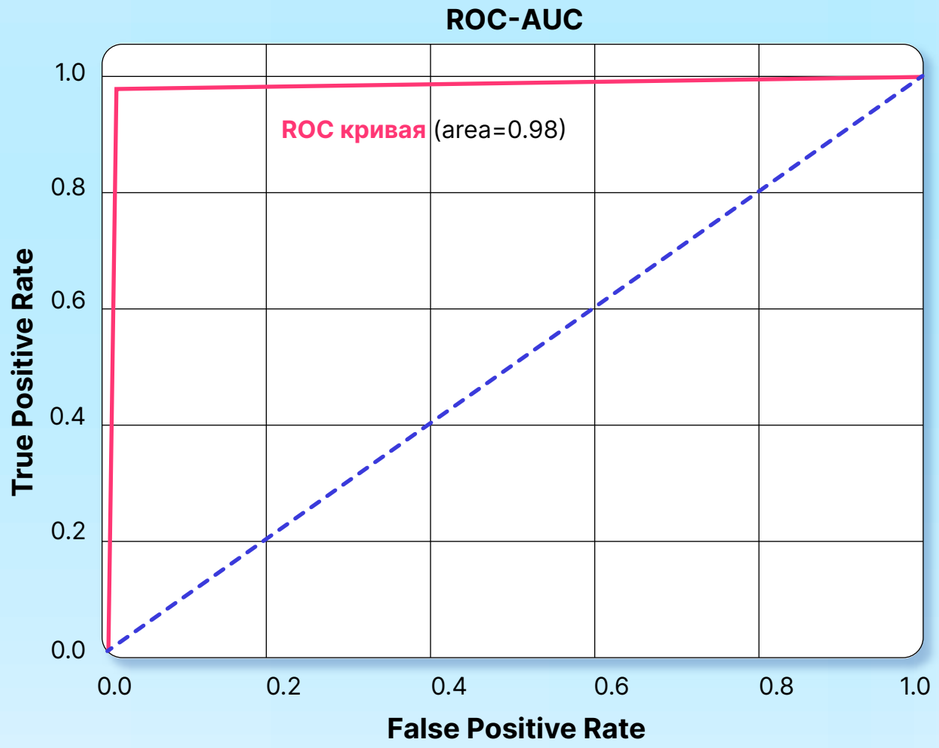
Бинарная классификация

Параметры:
k = 5

Результаты:

accuracy = 0.9811
f1 = 0.9779
ROC AUC = 0.981
Размер модели = 977.2 MB
Время обучения = 87 ms

Время прогноза
для одного примера
= 147 ms ± 1.28 ms



Модели машинного обучения. KNN

Многоклассовая классификация

Параметры:

k = 5

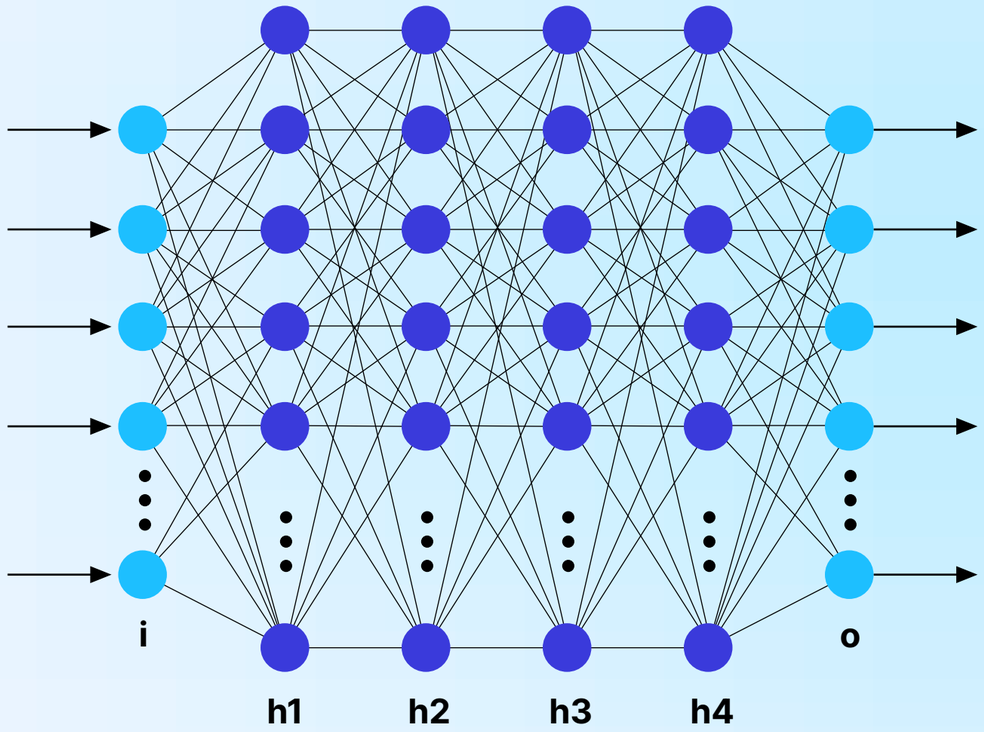
Результаты:
accuracy = 0.9157
f1 = 0.9156

Время прогноза
для одного примера
= 135 ms ± 1.3 ms

Размер модели = 416.5 MB

Время обучения = 41.4 ms

Neural network. Модель



Input layer (i) = 2381
Hidden layer1 (h1) = 6400
Hidden layer2 (h2) = 6400
Hidden layer3 (h3) = 3200
Hidden layer4 (h4) = 1600
Output layer (o) = {2;14}

После слоев i, h1, h2 выполняется пакетная нормализация (BatchNorm).

Neural network

Бинарная классификация

Гиперпараметры:

- скорость обучения: 9·10-5
- размер пакета: 64
- количество эпох: 80
- оптимизатор: Adam (adaptive moment estimation)
- функция активации: ReLU
- функция потерь: CrossEntropyLoss

В результате обучения были получены следующие метрики:

- max accuracy = 0.981475
- min loss = 0.04105

Neural network

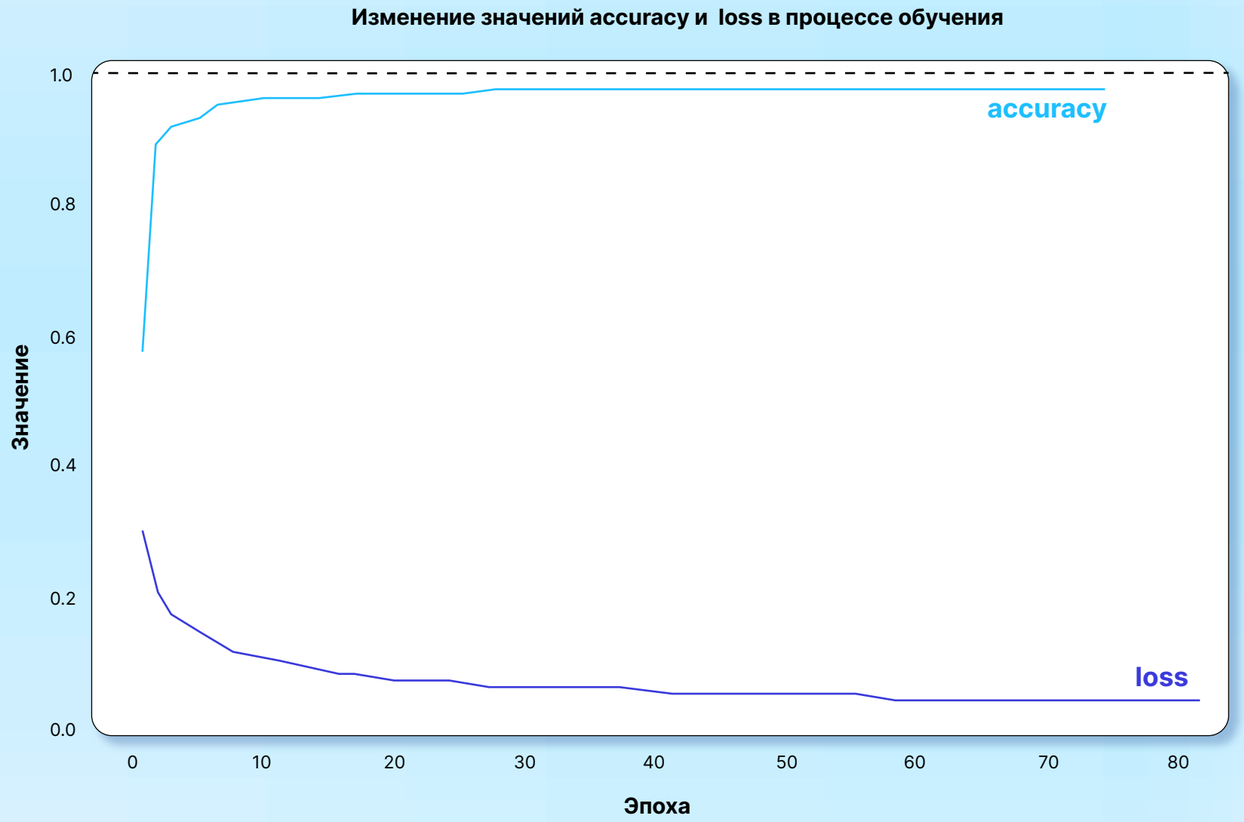
Результаты:

accuracy = 0.9815

Время работы сети на одном примере:
на GPU = 2.49 ms ± 10.8 µs
на CPU = 8.96 ms ± 108 µs

Размер сохраненных весов = 351.4 MB

Время обучения = 57.13 min



Neural network

Многоклассовая классификация

Гиперпараметры:

- скорость обучения: 7·10-5
- размер пакета: 64
- количество эпох: 80
- оптимизатор: Adam (adaptive moment estimation)
- функция активации: ReLU
- функция потерь: CrossEntropyLoss

В результате обучения были получены следующие метрики:

- max accuracy = 0.915307
- min loss = 0.188537

Neural network

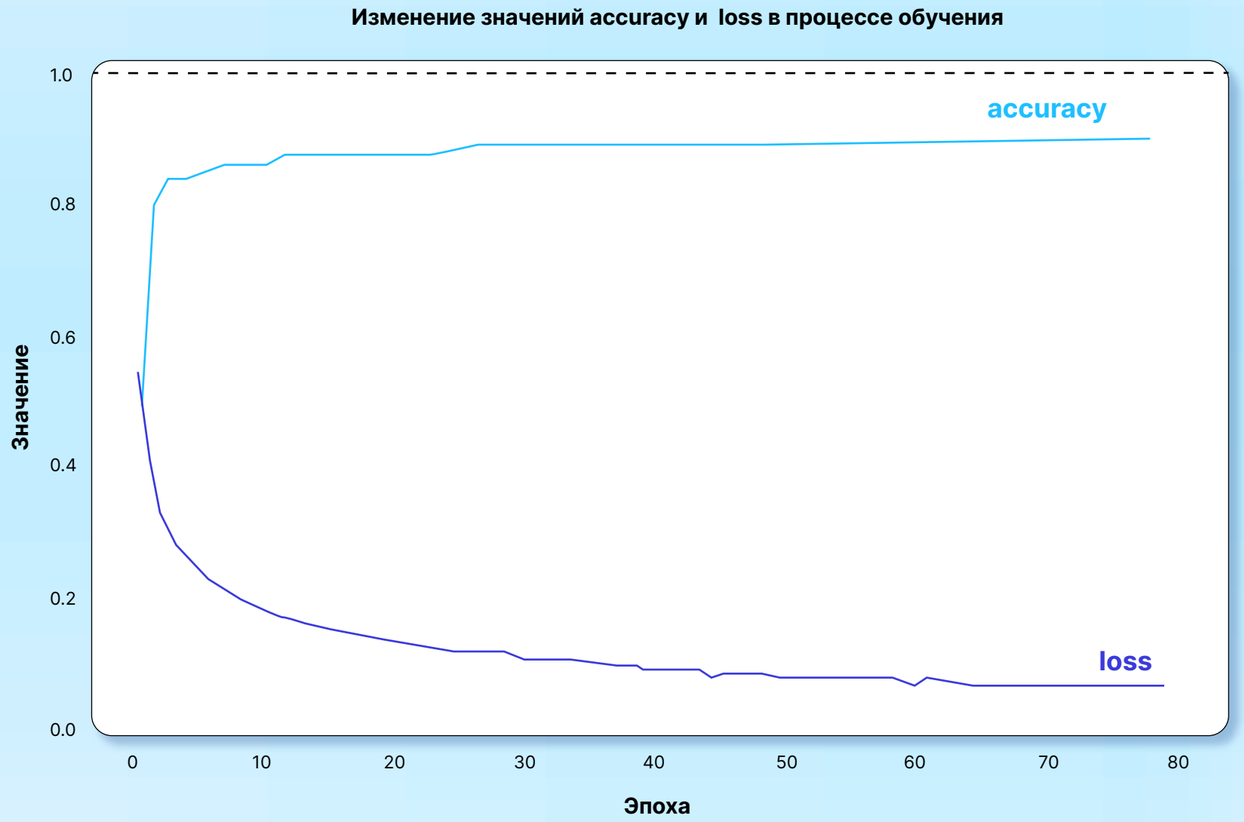
Результаты:

accuracy = 0.91814

Время работы сети на одном примере:
на GPU = 1.94 ms ± 10.3 µs
на CPU = 8.84 ms ± 11.6 µs

Размер сохраненных весов = 351.5 MB

Время обучения = 25.483 min



Результаты

Сводная таблица характеристик всех рассмотренных моделей

model	accuracy	roc auc	running time	model size	training time
Классификация на содержание вредоносного кода					
ElasticNet	0.0376(mse)	0.998	24.2 µs ± 147 ns	9.84 kB	1.76 min
RandomForest					
base	0.9953	0.994	24.3 ms ± 250 µs	90.4 MB	1.89 min
PCA	0.953	0.951	42.4 ms ± 179 µs	122 MB	9 s
optimal parameters	0.95	0.948	80.3 ms ± 200 µs	209.7 MB	2.76 min
KNN	0.9811	0.981	147 ms ± 1.28 ms	977 MB	87 ms
Neural network	0.9815		2.49 ms ± 10.8 µs(GPU) 8.96 ms ± 108 µs(CPU)	351 MB (веса)	57,13 min
Классификация по категориям					
ElasticNet	3.428(mse)		25.2 µs ± 499 ns	9,8 kB	1.59 min
RandomForest					
base	0.9449		25.2 ms ± 274 µs	352 MB	39.693 s
PCA	0.9078		22.9 ms ± 149 µs	233 MB	1.191 s
optimal parameters	0.9082		31 ms ± 769 µs	164.5 MB	23.589 s
KNN	0.9157		135 ms ± 1.3 ms	416.5 MB	41.4 ms
Neural network	0.9181		1.94 ms ± 10.3 µs(GPU) 8.84 ms ± 11.6 µs(CPU)	351.5 MB (веса)	25.483 min

Применение и дальнейшее развитие

Модели, полученные в результате работы,
могут быть использованы в качестве
предварительного классификатора в случае интеграции
со средствами анализа поведения исследуемых объектов

VR Protect

Спасибо
за внимание

ВЫЧИСЛИТЕЛЬНЫЕ РЕШЕНИЯ

Н. В. Щелкунова
Вычислительные решения

Re:Crypt

к.т.н., **Д. А. Щелкунов**
Рекрипт