

Методы искусственного интеллекта в анализе данных

«Прогноз качества воздуха» - Ильяда Алушай.

«Лучшая производительность набора данных Abalone с NN» - Данил Кириленко

Dataset 1 : <https://archive.ics.uci.edu/ml/datasets/Air+Quality>

Dataset 2 : <https://archive.ics.uci.edu/ml/datasets/Abalone>

Data information (Информация о данных)

Данные получены из репозитория машинного обучения UCI. Он был зарегистрирован 5 химическими датчиками оксидов металлов, расположенными в сильно загрязненном районе итальянского города, и мы решили проанализировать один из них, CO (поскольку это задача многомерного временного ряда). Набор данных содержит 9538 объектов с марта 2004 г. по февраль 2005 г.

Ниже приводится описание набора данных:

- 0 Date (DD/MM/YYYY)
- 1 Time (HH.MM.SS)
- 2 True hourly averaged concentration CO in mg/m³ (reference analyzer)
- 3 PT08.S1 (tin oxide) hourly averaged sensor response (nominally CO targeted)
- 4 True hourly averaged overall Non Metanic HydroCarbons concentration in microg/m³ (reference analyzer)
- 5 True hourly averaged Benzene concentration in microg/m³ (reference analyzer)
- 6 PT08.S2 (titania) hourly averaged sensor response (nominally NMHC targeted)
- 7 True hourly averaged NOx concentration in ppb (reference analyzer)
- 8 PT08.S3 (tungsten oxide) hourly averaged sensor response (nominally NOx targeted)
- 9 True hourly averaged NO2 concentration in microg/m³ (reference analyzer)
- 10 PT08.S4 (tungsten oxide) hourly averaged sensor response (nominally NO2 targeted)
- 11 PT08.S5 (indium oxide) hourly averaged sensor response (nominally O3 targeted)
- 12 Temperature in Â°C
- 13 Relative Humidity (%)
- 14 AH Absolute Humidity

Data preprocessing (Предварительная обработка данных)

В этих наблюдениях отсутствуют значения, помеченные как «-200», которые мы преобразовали как NaN для последующей предварительной обработки.

Мы создали новые переменные из целевых переменных и заменили значение NaN их средним значением.

Чтобы получить индекс datetime, мы объединили столбцы «Date» и «Time», преобразовав тип данных из строки в datetime и сохранили новый «Datetime» в списке.

Data Analysis (Анализ данных)

Графики графиков временных рядов уровней S1 (CO).

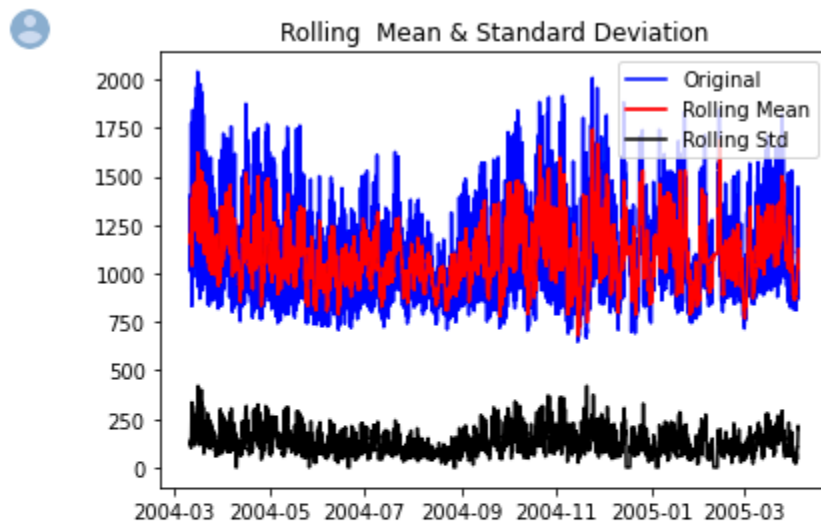
Мы посмотрели на графики за более короткий период: «2004-10-05» - «2004-10-08», S1-S5 и заметили, что шаблон постоянно повторяется в течение определенного периода времени.

Checking for Stationarity of Time series (Проверка стационарности временных рядов)

Временной ряд является стационарным, если его статистические свойства, такие как среднее значение, дисперсия, остаются постоянными во времени и автоковариация, которая не зависит от времени.

Мы использовали график скользящей статистики вместе с тестом Дики-Фуллера, и вот результаты:

```
test_stationarity(S1)
```



Results of Dickey-Fuller Test:

Test Statistic	-9.732748e+00
p-value	8.914162e-17
#Lags Used	3.800000e+01
Number of Observations Used	9.318000e+03
Critical Value (1%)	-3.431052e+00
Critical Value (5%)	-2.861850e+00
Critical Value (10%)	-2.566935e+00
dtype:	float64

С постоянным скользящим средним и скользящей дисперсией, а также с р-значением теста Дики-Фуллера, близким к 0, мы можем сказать, что S1 является слабым и стационарным.

Сборка test_stationary также для S2, S3, S4, S5.

Мы попытались сделать серию стационарной, используя логарифмическое преобразование.

Moving Average (Скользящая средняя)

В этом подходе мы берем среднее значение «к» последовательных значений в зависимости от частоты временных рядов. Здесь мы можем взять среднее значение за последний год, т.е. за последние 12 значений. В Pandas есть определенные функции для определения скользящей статистики.

Мы берем «взвешенное скользящее среднее», где более поздним значениям присваивается больший вес. Эта TS имеет еще меньшие вариации в среднем и стандартном отклонении по величине. Кроме того, статистика теста меньше критического значения 1%, что лучше, чем в предыдущем случае..

Eliminating Trend and Seasonality (Устранение тренда и сезонности)

Два метода:

1. Разница (снятие разницы с определенным временным лагом)
2. 2. Декомпозиция (моделирование тренда и сезонности и удаление их из модели)

Мы использовали оба этих метода для нашего ряда S1, и наш временной ряд, наконец, очень близок к стационарному.

Forecasting a Time Series (Прогнозирование временного ряда)

Использование модели ARIMA / ARIMAX для прогнозирования будущих значений временного ряда S1.

Прогнозирование ARIMA для стационарного временного ряда - это не что иное, как линейное (например, линейная регрессия) уравнение. Предикторы зависят от параметров (p, d, q) модели ARIMA:

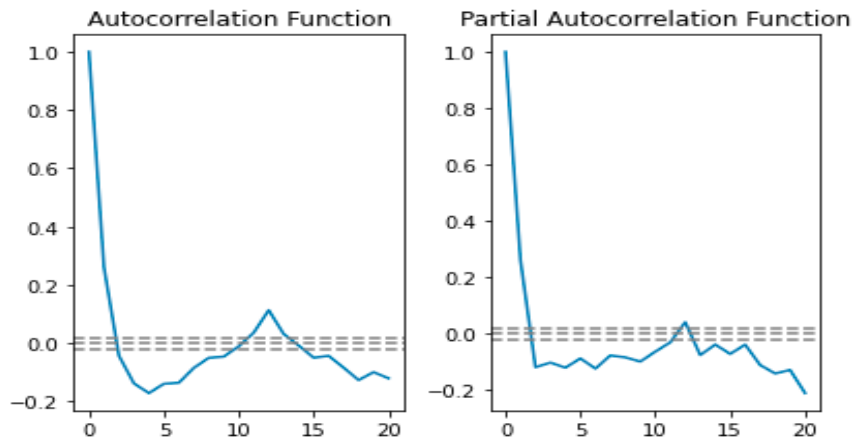
1. Количество терминов AR (авторегрессивных) (p)
- 2.Количество условий скользящей средней (q)
- 3.Количество отличий (d)

Здесь важно знать, как определить значение «p» и «q». Для определения этих чисел мы используем два графика.

1.Функция автокорреляции (ACF): это мера корреляции между TS и самой лаговой версией.

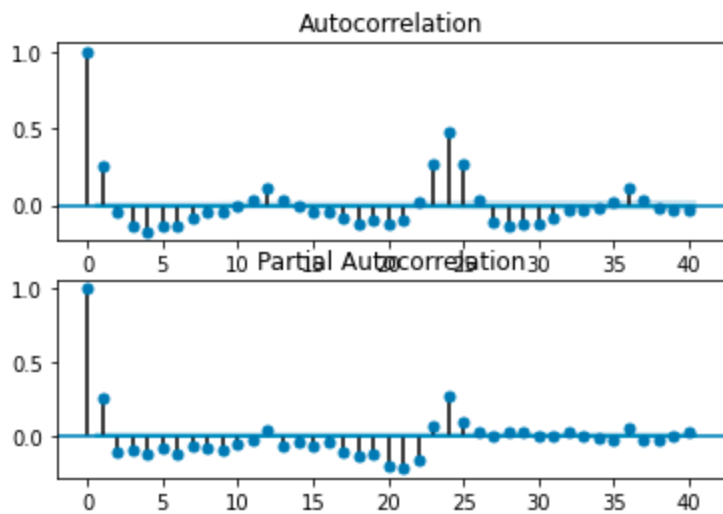
2.Функция частичной автокорреляции (PACF): измеряет корреляцию между TS с запаздывающей версией самого себя, но после устранения вариаций, уже объясненных промежуточными сравнениями.

Мы создали графики ACF и PACF:



На этом графике две пунктирные линии по обе стороны от 0 представляют собой доверительные интервалы. Их можно использовать для определения значений «p» и «q».

Способ 2:

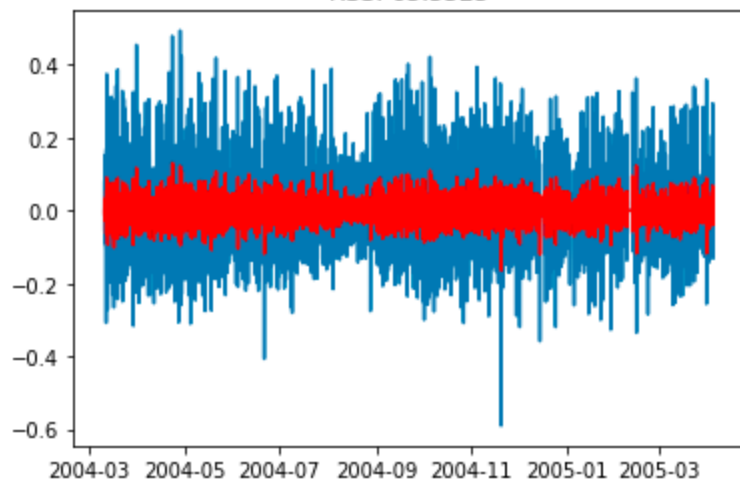


После этого мы создали модель ARIMA, сочетающую модели AR и MA. Здесь мы видим, что модели AR и MA имеют почти одинаковый RSS, но их сочетание значительно лучше.

Модель AR и AM:

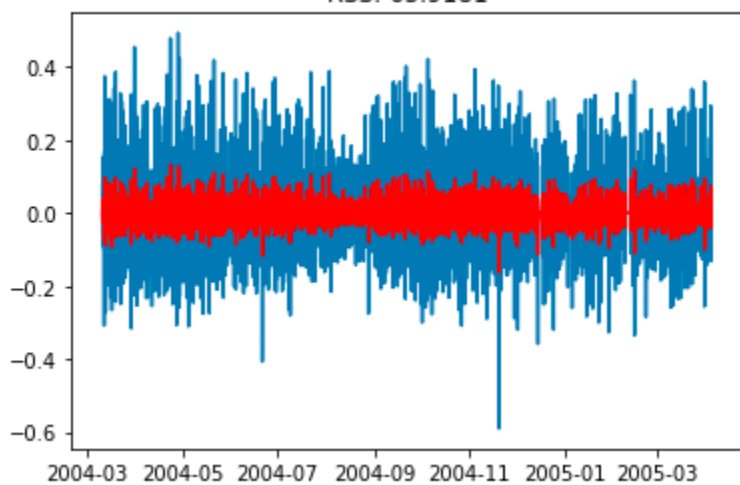
Text(0.5, 1.0, 'RSS: 65.5525')

RSS: 65.5525



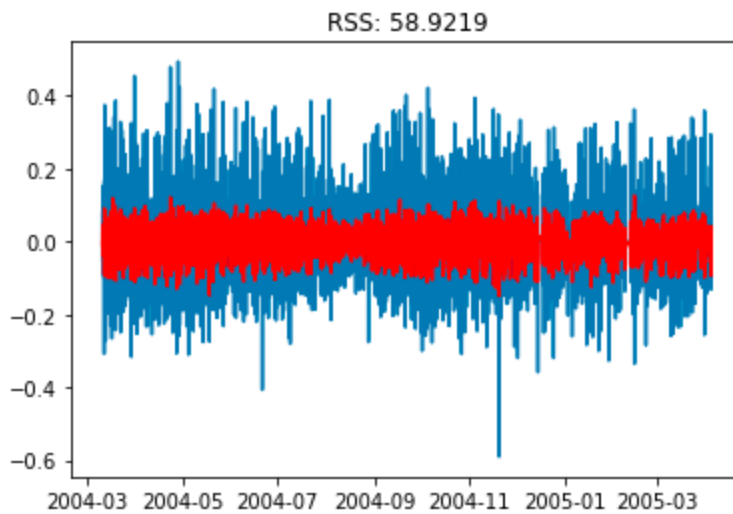
Text(0.5, 1.0, 'RSS: 65.9181')

RSS: 65.9181



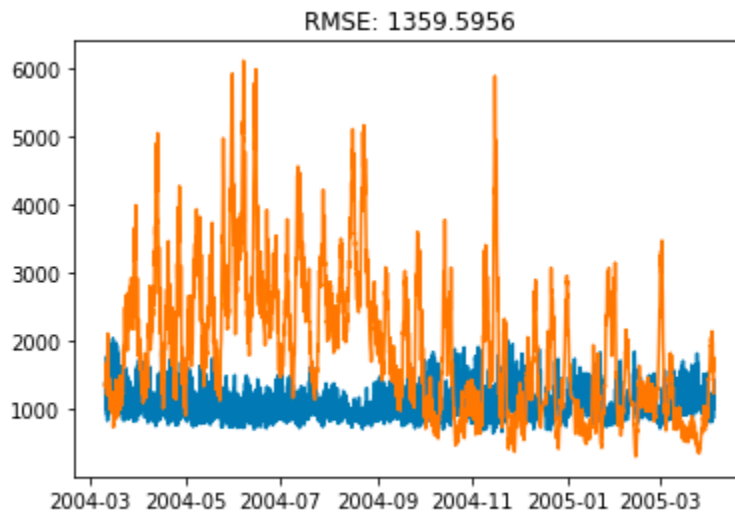
Комбинированная модель ARIMA: :

```
Text(0.5, 1.0, 'RSS: 58.9219')
```



Здесь мы видим, что модели AR и MA имеют почти одинаковый RSS, но их сочетание значительно лучше.

```
Text(0.5, 1.0, 'RMSE: 1359.5956')
```



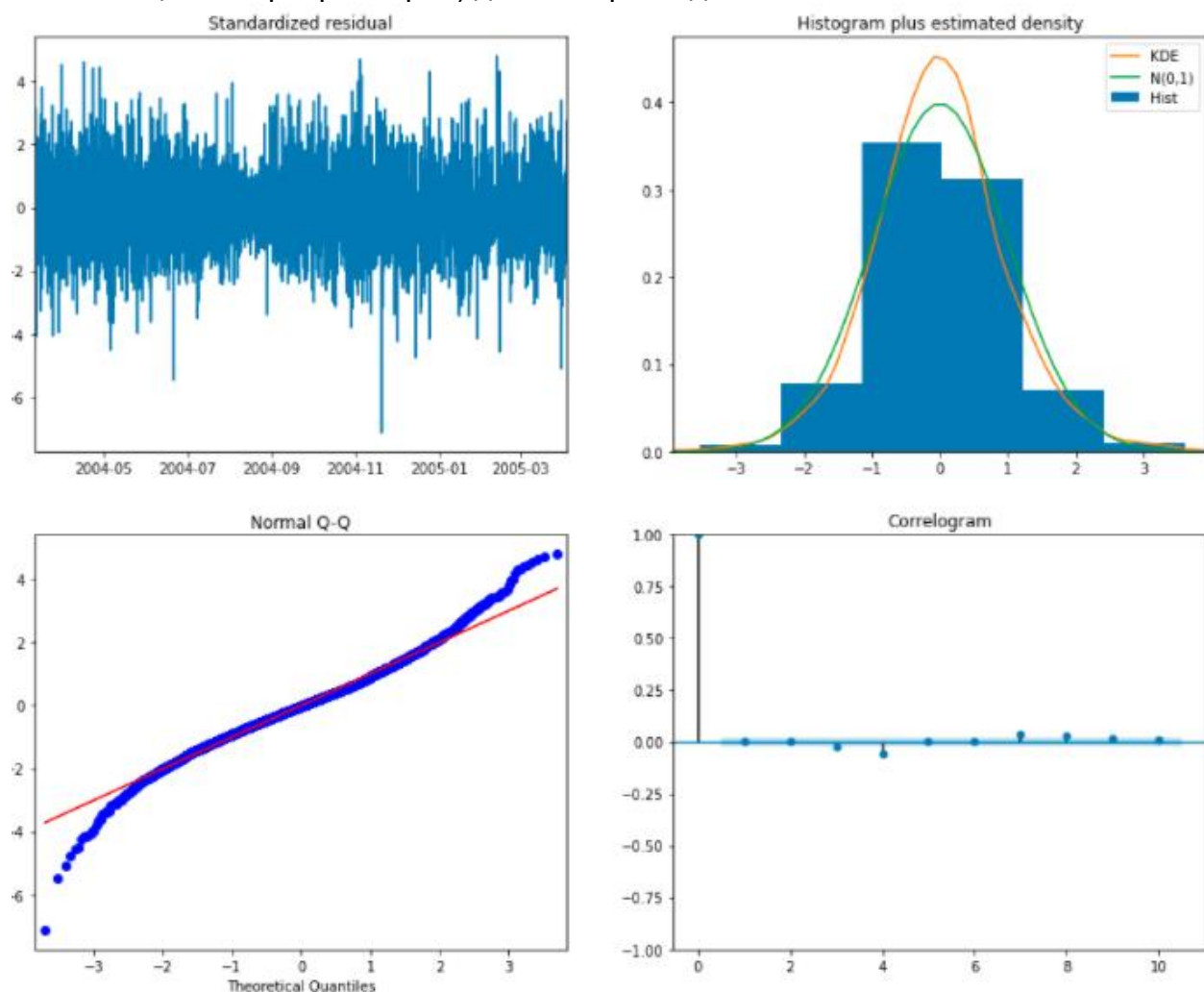
С помощью прогнозов мы видим, что эта модель не так хороша, как должна быть, и среднеквадратичная ошибка также очень высока.

Another Method – SARIMAX (Другой метод -SARIMAX))

	coef	std err	z	P> z	[0.025	0.975]
ar.L1	0.8702	0.006	152.905	0.000	0.859	0.881
ma.L1	0.2064	0.010	21.554	0.000	0.188	0.225
ar.S.L12	-0.1714	0.010	-17.111	0.000	-0.191	-0.152
ma.S.L12	-0.9370	0.003	-275.473	0.000	-0.944	-0.930
sigma2	0.0051	5.75e-05	89.050	0.000	0.005	0.005

Выбор параметров для модели временных рядов ARIMA

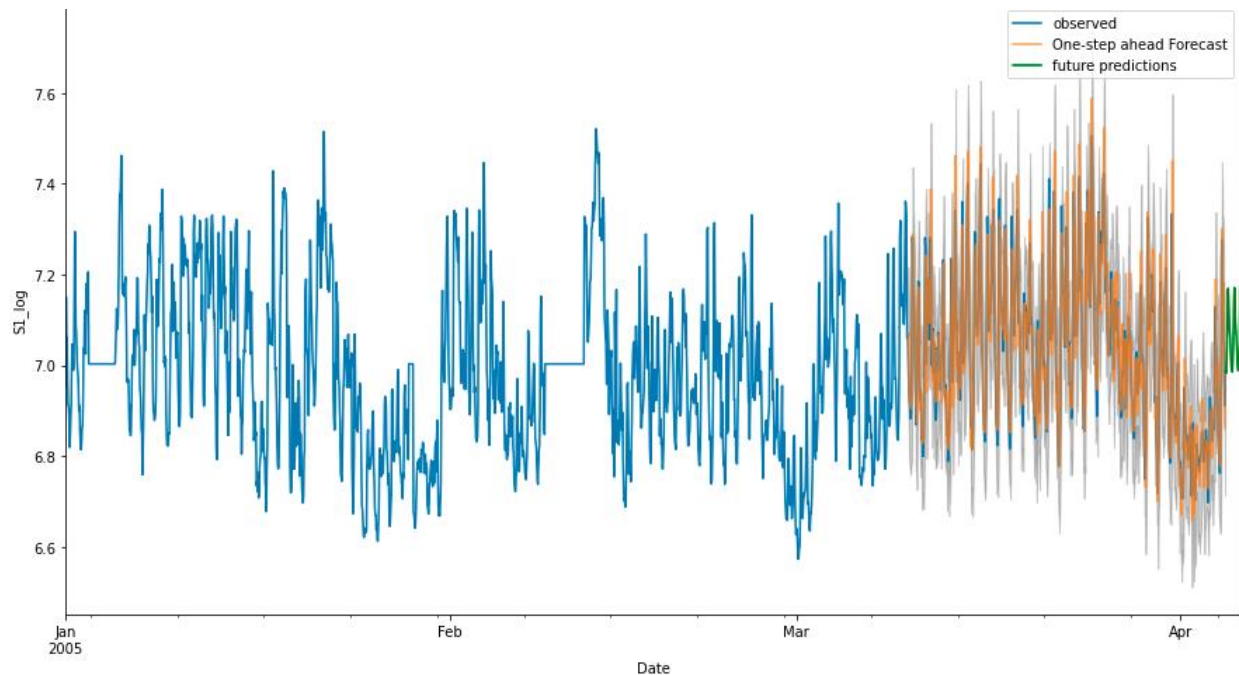
В статистике и машинном обучении этот процесс известен как поиск по сетке (или оптимизация гиперпараметров) для выбора модели.



1. На правом верхнем графике мы видим, что красная линия KDE близко следует за линией $N(0,1)$ (где $N(0,1)$ - стандартное обозначение для нормального распределения со средним 0 и стандартным отклонением 1). Это хороший показатель того, что остатки распределены нормально.
2. qq-график в нижнем левом углу показывает, что упорядоченное распределение остатков (синие точки) следует линейному тренду

выборки, взятых из стандартного нормального распределения с $N(0, 1)$. Опять же, это явный признак того, что остатки распределены нормально. 3. Невязки во времени (верхний левый график) не показывают явной сезонности и выглядят как белый шум. Это подтверждается графиком автокорреляции (то есть корелограммой) в правом нижнем углу, который показывает, что остатки временных рядов имеют низкую корреляцию с запаздывающими версиями самих себя.

Validating Forecasts (Проверка прогнозов)



Mean Squared Error of forecast : 0.005
Mean Absolute Percentage Error: 0.76%
The Root Mean Squared Error of our prediction is 0.07

Имея низкий уровень ошибки, мы заключаем, что этот прогноз точен.

Предсказание возраста морских ушек (часть 2)

Из набора данных из предыдущей задачи были выделены два самых часто встречающихся класса (9 и 10), которые между собой сбалансированы (689 и 634 образца). В качестве baseline была использована простая логистическая регрессия (sklearn.linear_model.LogisticRegression), эта модель показала следующие результаты:

	precision	recall	f1-score	support
9.0	0.57	0.76	0.65	153
10.0	0.71	0.49	0.58	178
accuracy			0.62	331
macro avg	0.64	0.63	0.62	331
weighted avg	0.64	0.62	0.61	331

Roc-auc score: 0.6295439524124256

Wall time: 9 ms

Для улучшения полученных результатов искалась подходящая архитектура полносвязной нейронной сети, среди различных вариантов лучше всего себя показала следующая модель:

Layer (type)	Output Shape	Param #
Linear-1	[-1, 10]	110
Linear-2	[-1, 80]	880
Linear-3	[-1, 50]	4,050
Linear-4	[-1, 1]	51

Total params: 5,091

Trainable params: 5,091

Non-trainable params: 0

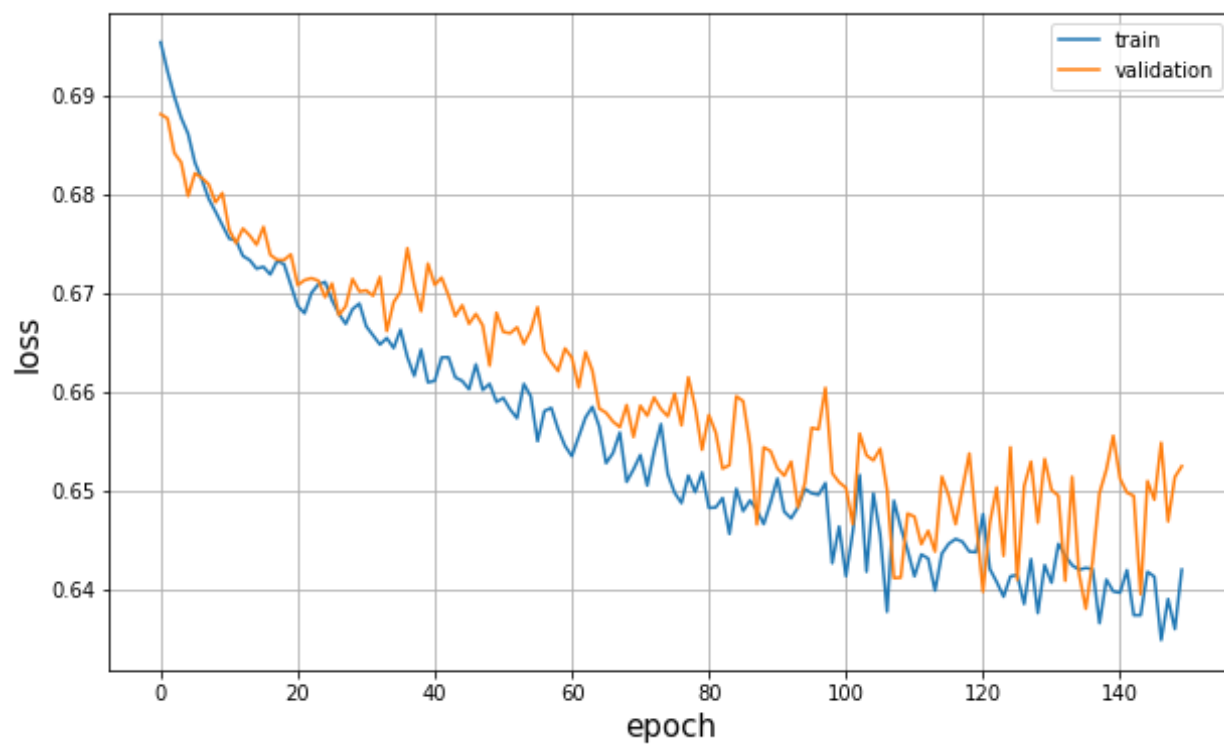
Input size (MB): 0.00

Forward/backward pass size (MB): 0.00

Params size (MB): 0.02

Estimated Total Size (MB): 0.02

Кроме этого использовался Dropout с вероятностью 0,1. Во время обучения при получении лучшего результата на валидации веса модели сохранялись, в итоге, при оценке использовалась та модель, которая лучше все себя показала на валидационном множестве за все время обучения, ниже графики обучения и итоговые результаты обученной модели.



	precision	recall	f1-score	support
0.0	0.57	0.78	0.66	153
1.0	0.72	0.49	0.58	178
accuracy			0.63	331
macro avg	0.65	0.64	0.62	331
weighted avg	0.65	0.63	0.62	331

Roc-auc score: 0.6867885731071455