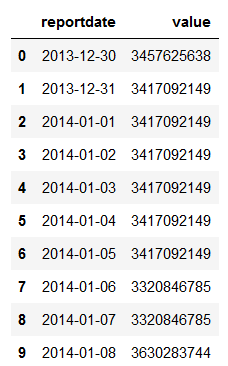
Гридасов Егор

Для решения задачи я пробовал несколько подходов: LSTM, RandomForestRegressor, линейную регрессию. В итоге я пришел к выводу, что для такой задачи, где нужно предсказать до 360 следующих значений, лучше всего подходит линейная регрессия.

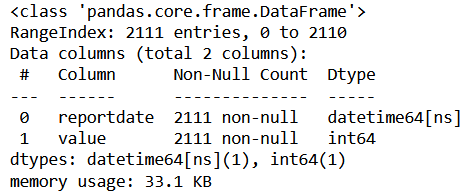
Для начала я обработал данные и привел их к удобному для работы виду.



**Рисунок 1 — Преобразование данных**

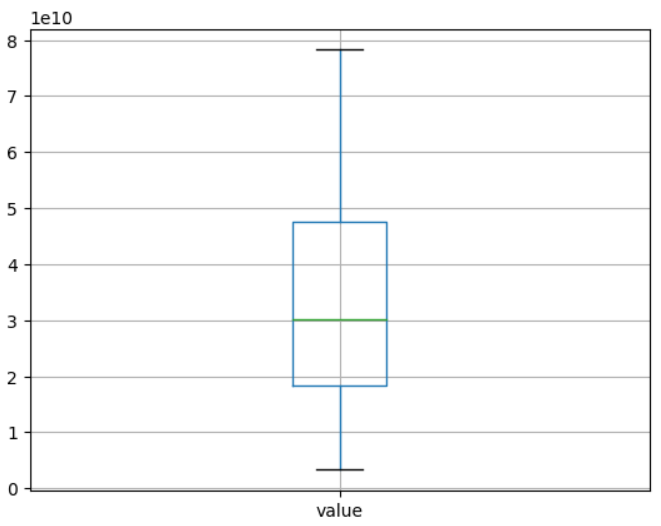


**Рисунок 2 — Преобразованный датасет**

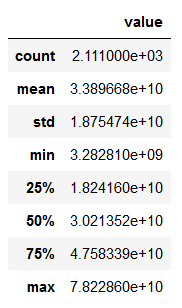


**Рисунок 3 — Преобразованный датасет**

После я решил поближе познакомиться с данными и выяснить, есть ли выбросы. Пришел к выводу, что данные не имеют выбросок.

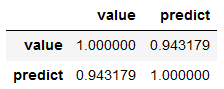


**Рисунок 4 — boxplot по значениям**



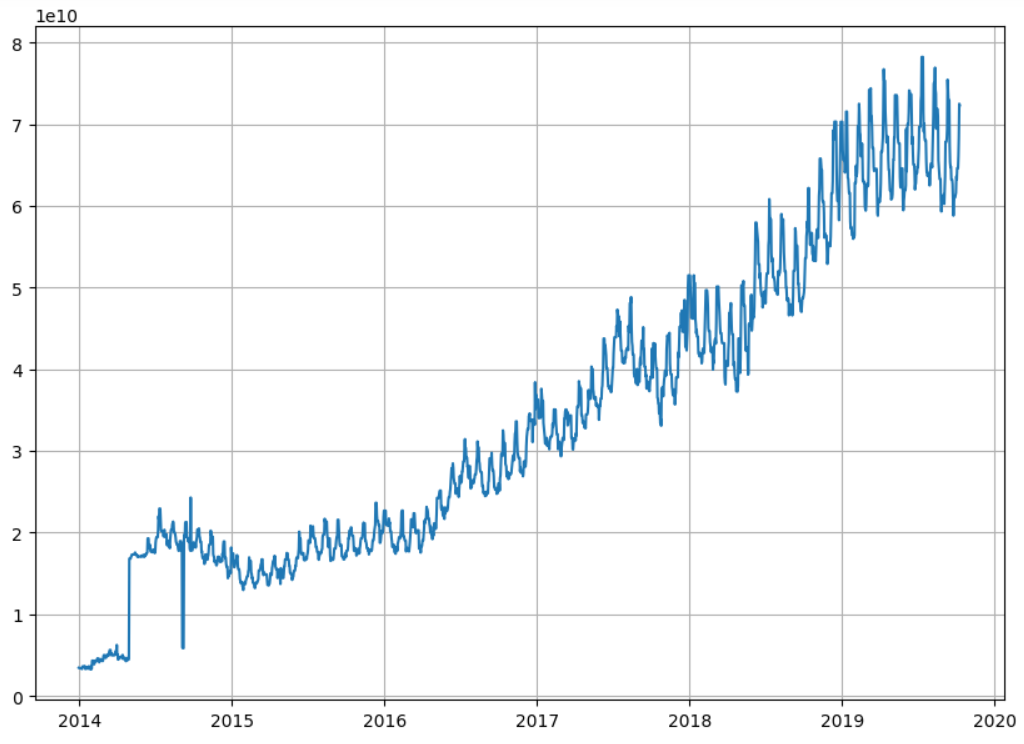
**Рисунок 5 — Сводка по данным value из датасета**

Также я построил матрицу корреляций, сместив данные для предсказания на максимально возможный промежуток в рамках задачи: 360 значений. В итоге обнаружилась сильная корреляция, что очень хорошо (0.94).



**Рисунок 6 — Матрица корреляций**

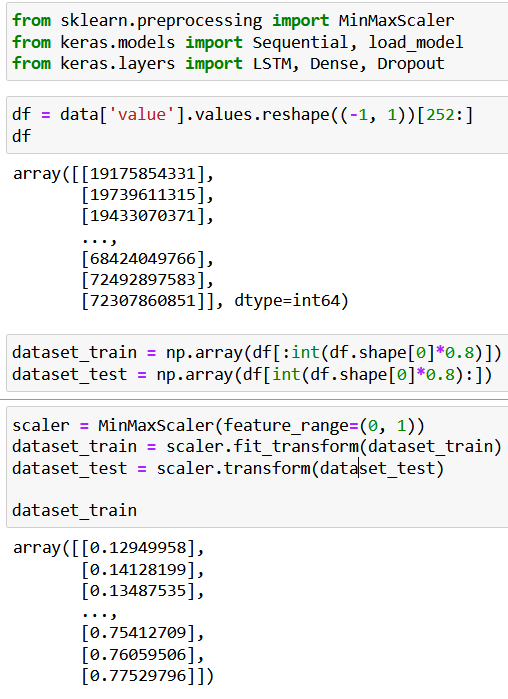
После я построил график изначальных значений, чтобы понять, как ведут себя данные с течением времени.



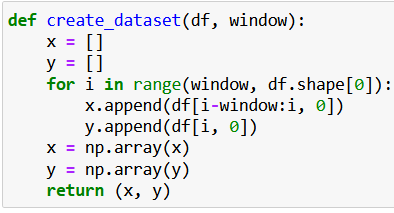
**Рисунок 7 — График значений датасета**

Первые 250 значений не видно закономерности, но постепенно данные обретают некоторую траекторию развития.

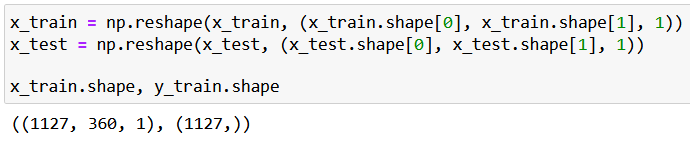
Для начала я попробовал построить LSTM-модель, для проверки разделив датасет и обработав данные, сместив их.



**Рисунок 8 — Подготовка данных для обучения**

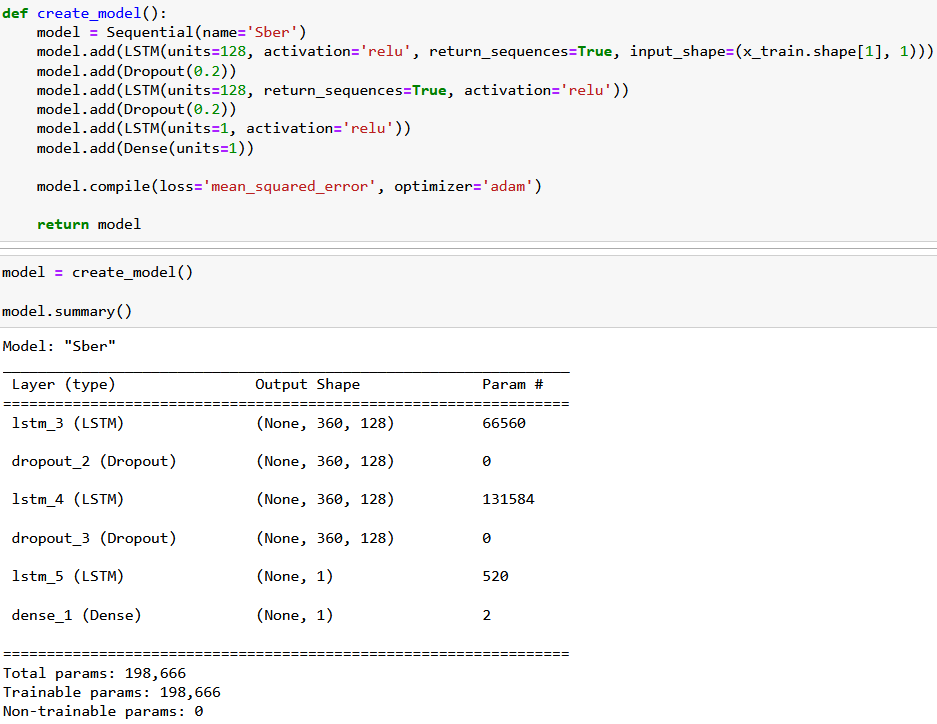


**Рисунок 9 — Функция для подготовки данных**

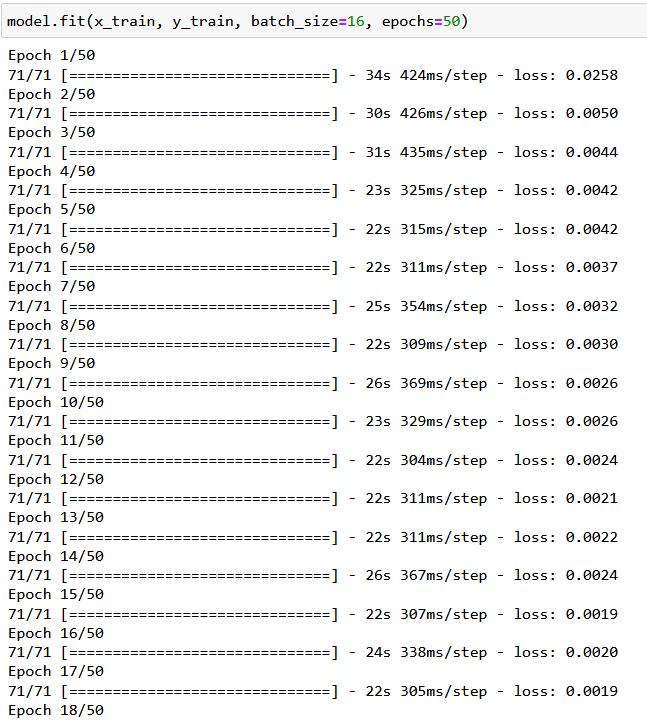


**Рисунок 10 — Итоговая размерность данных**

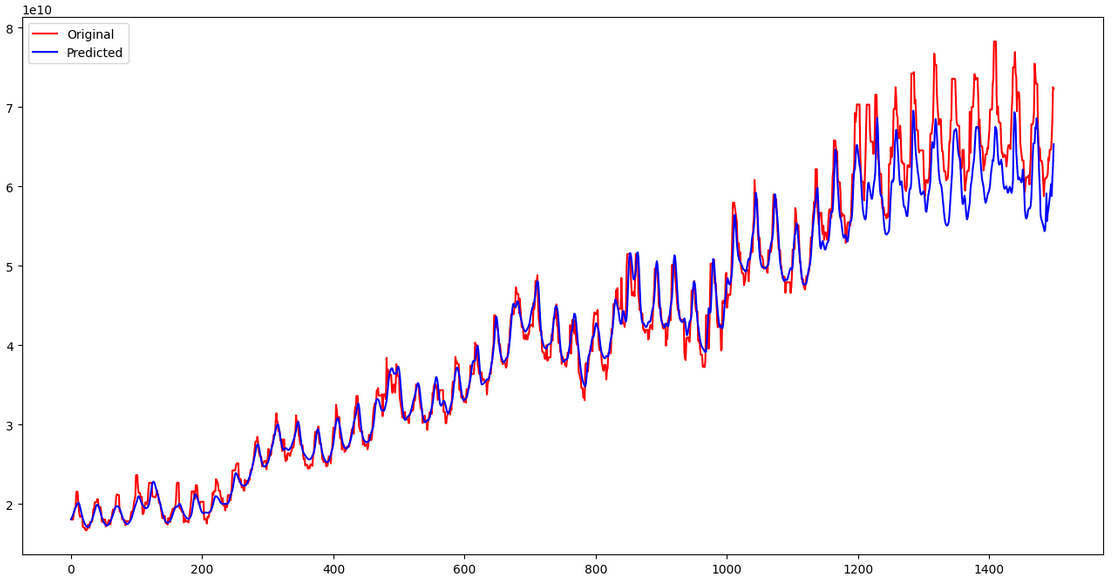
После того, как данные были подготовлены, я решил, что настало время для создания и обучения модели.



**Рисунок 11 — Созданная модель**



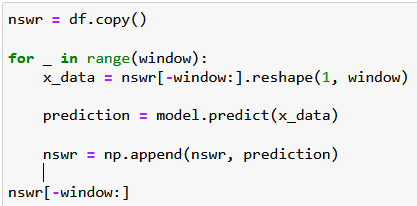
**Рисунок 12 — Процесс обучения модели**



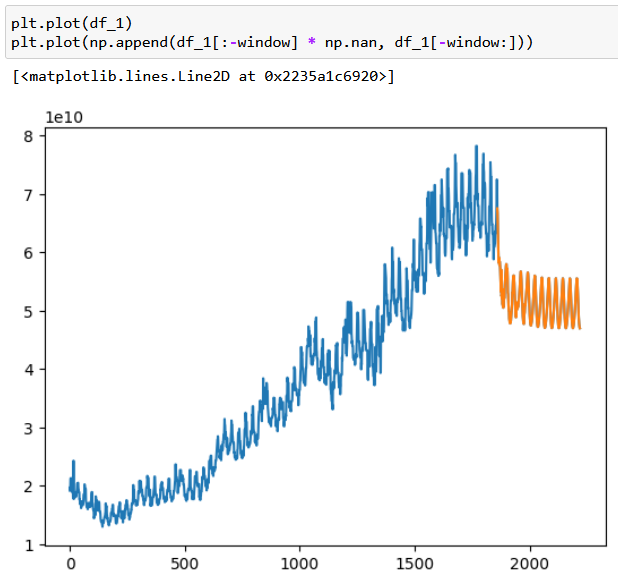
**Рисунок 13 — Тестирование модели**

Несмотря на то, что модель хорошо показывает себя на тренировочных данных, на тестовых она ведет себя уже не так хорошо, что мне показалось плохим знаком. К сожалению, мои опасения подтвердились, когда я попробовал предсказать следующие 360 значений.

Для предсказаний я брал последние 360 значений для создания одного нового. Добавлял это значение в изначальный массив и таким образом двигался вперед.



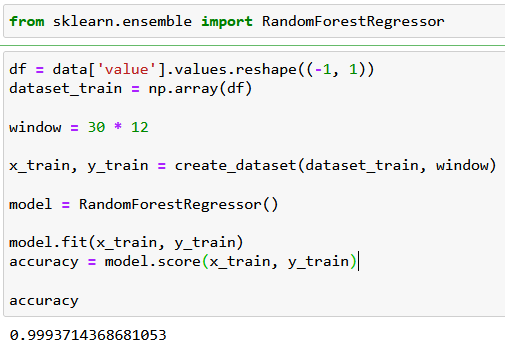
**Рисунок 14 — Процесс предсказания**



**Рисунок 15 — Тестирование модели**

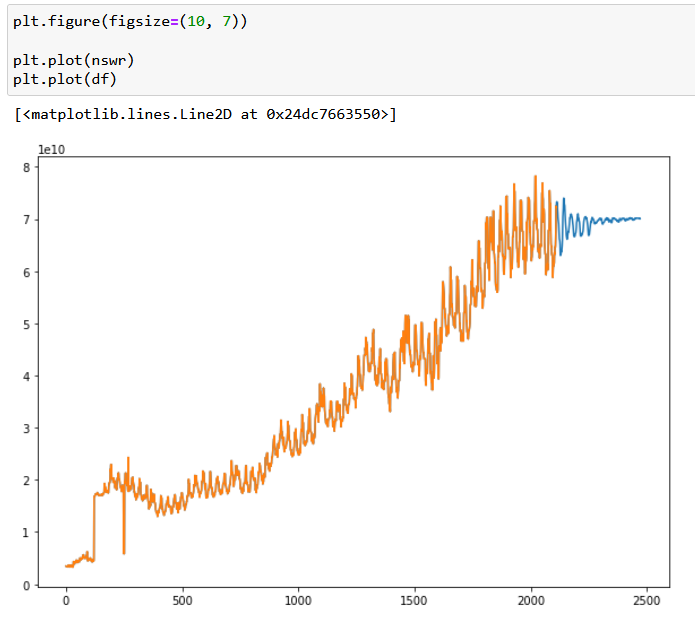
Как мы видим, модель не только занижает значения сразу после первых предсказаний, но и подстраивается под паттерн последних значений, что приводит к бесконечной синусоиде. Я пытался настроить модель и тренировать ее в разных пропорциях и с разными слоями, но результат был одинаковый, если не хуже.

Следующим я попробовал RandomForestRegressor.



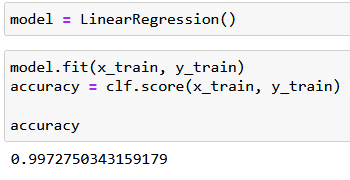
**Рисунок 16 — Создание, обучение и проверка модели**

Эту модель я пытался обучить на всей генеральной совокупности, но это привело к похожему результату, что и LSTM – модель предсказывает значения, подстраивается под новые и в итоге идет по своему паттерну. В данном случае модель ушла в затухающую синусоиду.



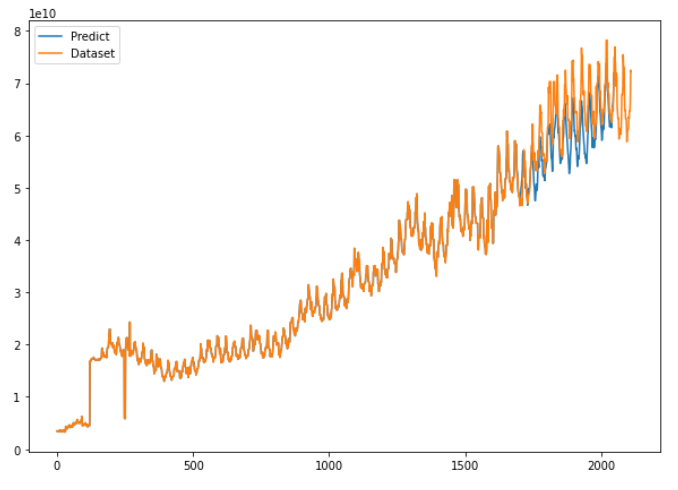
**Рисунок 17 — Построение графика с изначальным датасетом и предсказанием**

Последним вариантом я попробовал линейную регрессию. Линейная регрессия имеет недостатки, но отсутствие выбросов немного улучшает ситуацию.



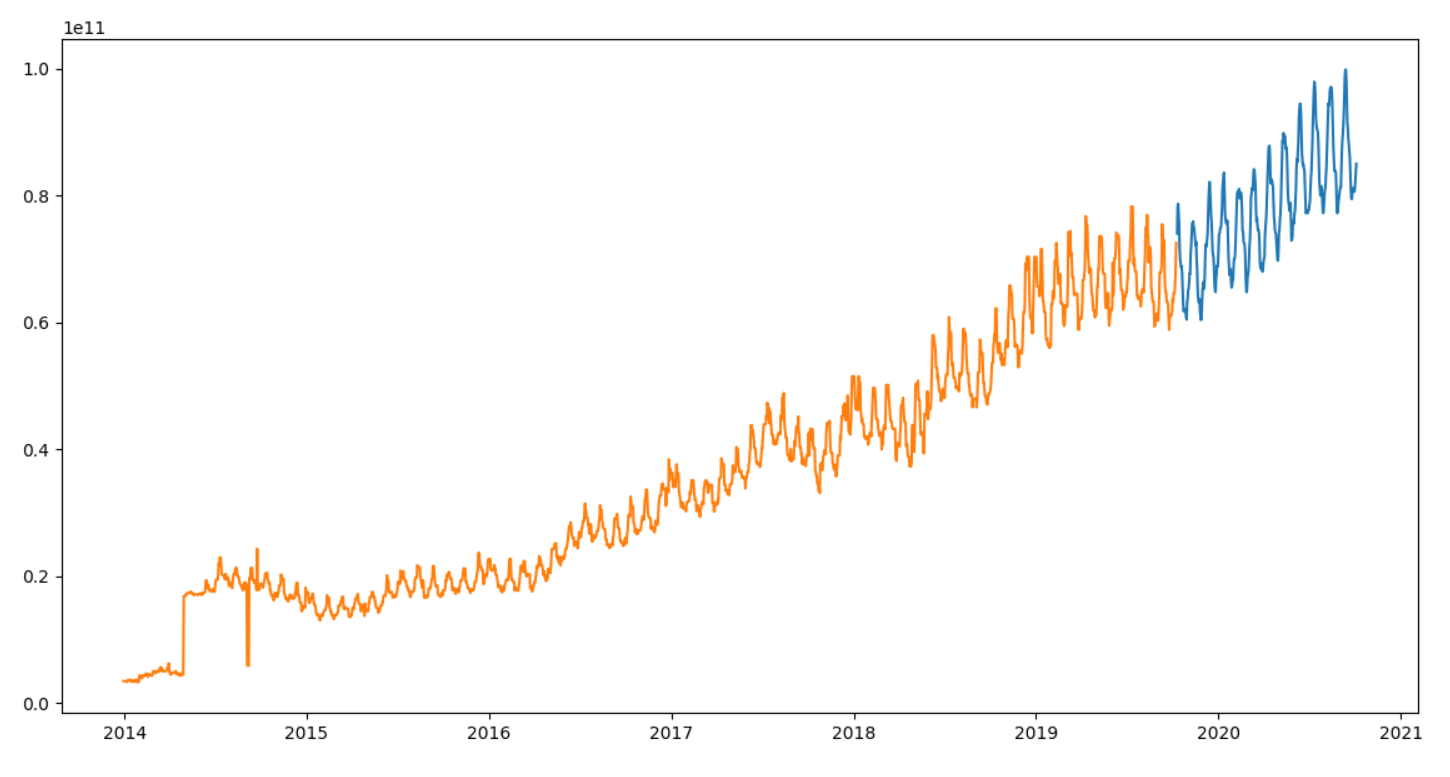
**Рисунок 17 — Построение модели и обучение на тренировочных данных**

В результате предсказания на тренировочных данных получились в достаточной мере верными, чтобы выбрать эту модель для выполнения задания.



**Рисунок 18 — Результат работы модели на тренировочных данных**

Дальше я создал два файла: один для создания модели, а второй — main — для выполнения основной программы. По итогу обучения пользователь может сохранить модель, если ее надо будет использовать в дальнейшем.



**Рисунок 18 — Результат работы программы на всей генеральной совокупности с окном в 360 значений**

В результате работы была построена модель, предсказывающая 30 \* количество\_месяцев значений. Результат работы программы заключается в сохранении модели и предсказаний в виде графика.