Министерство науки и высшего образования Российской Федерации

ФГАОУ ВПО «Уральский Федеральный Университет

имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ

Школа бакалавриата

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРЕДСКАЗАНИЕ КОЛИЧЕСТВА ВЫБРОСОВ УГЛЕКИСЛОГО ГАЗА ПРИ ПРОИЗВОДСТВЕ ЭЛЕКТРОЭНЕРГИИ В ОПРЕДЕЛЕННЫЙ ГОД**

Пояснительная записка к курсовому проекту по модулю

«Методы анализа BigData»

090302 0000 726 ПЗ

090302 0000 715 ПЗ

Подпись Дата

Преподаватель \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ Мирвода С.Г.

Студенты \_\_\_\_\_\_\_ \_\_\_\_\_\_\_ Катаев А.Ю.

\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ Катаев С.М.

Группа РИ-450004

Екатеринбург 2019

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 2](#_Toc535766534)

[ПОСТАНОВКА ЗАДАЧИ 3](#_Toc535766535)

[ОПИСАНИЕ ТЕСТОВОГО НАБОРА ДАННЫХ 4](#_Toc535766536)

[ПОДГОТОВКА ДАННЫХ 5](#_Toc535766537)

[РАЗРАБОТКА МОДЕЛИ 17](#_Toc535766538)

[РАЗРАБОТКА СЕРВИСА 21](#_Toc535766539)

[ЗАКЛЮЧЕНИЕ 22](#_Toc535766540)

[ПРИЛОЖЕНИЕ 24](#_Toc535766541)

# 

# ВВЕДЕНИЕ

С каждым годом во всем мире растёт потребление электроэнергии, большинство новых технологий требуют все возрастающих затрат электроэнергии. Для выработки электроэнергии создано большое число электростанций, работающих на том или ином виде топлива или силе.

В данной работе будет изучена выработка углекислого газа при производстве электроэнергии при сжигании различных видов топлива во всем мире.

Информацию, использованную в этой работе можно использовать для прогнозирования будущих выработок углекислого газа и для оценки влияния данного фактора на экологию.

Машинное обучение позволяет более детально изучить зависимости различных факторов и выявить их отношения. Так в данной работе использование машинного обучения позволит предсказывать эмиссию углекислого газа от электростанций (сжигающих различные виды топлива) в будущие года.

# ПОСТАНОВКА ЗАДАЧИ

Целями работы являются рассмотрение и применение методов статистики и машинного обучения для анализа набора данных.

В ходе работы необходимо сделать следующее:

* Выбрать набор данных;
* Построить статистическую модель и натренировать ее;
* Разработать приложение с пользовательским интерфейсом для запроса параметров для прогнозирования;
* Передать параметры в модель и получить прогноз;
* Отобразить прогнозные значения.

# ОПИСАНИЕ ТЕСТОВОГО НАБОРА ДАННЫХ

Тестовый набор данных – эмиссия углекислого газа при производстве электроэнергии во всем мире от 1973 до 2014гг из официального статистического центра.

Описание столбцов данных.

* Столбец «MSN»: содержит общую информацию по записи в таблице;
* Столбец «YYYYMM»: содержит дату измерения;
* Столбец «Value»: содержит непосредственно значение эмиссии СО2 на дату измерения;
* Столбец «Column\_Order»: содержит тип сжигаемого топлива;
* Столбец «Description»: содержит подробное описание типа сжигаемого топлива;
* Столбец «Unit»: содержит описание единицы измерения значения эмиссии;

Информация о данных:

* MSN 5094 non-null object
* YYYYMM 5094 non-null int64
* Value 5094 non-null object
* Column\_Order 5094 non-null int64
* Description 5094 non-null object
* Unit 5094 non-null object

dtypes: int64(2), object(4)

# ПОДГОТОВКА ДАННЫХ

Импортируем библиотеки (Приложение 1).

Получение набора данных временных рядов приведено в Приложении 2.

Общую сумму выбросов CO2 от каждого вида топлива за каждый год можно просмотреть в строке, обозначенной 13 месяцем.

Подготовку данных начнем с удаления строк без datetimeindex и конвертации индекс в поле даты и времени, корректируем ошибки и удаляем общую сумму за год.

После первоначальной очистки данных у нас осталось вместо 5094 записей 4707.

Набор данных имеет 8 источников энергии выбросов CO2. Далее сгруппируем набор данных по выбросам CO2 в зависимости от типа источника энергии и визуализируем ее (Рисунок 1) (Приложение 3).

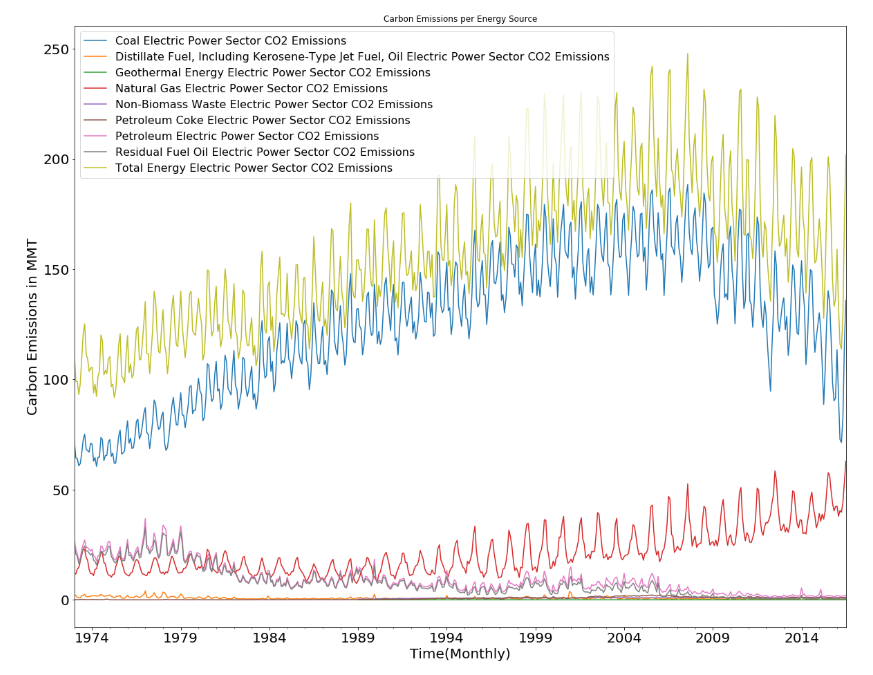


Рисунок 1 – Выбросы СО2 в зависимости от типа источника энергии

Визуализируем, как влияет тренд использования топлива и сезонности на выброс CO2 от каждого источника энергии.

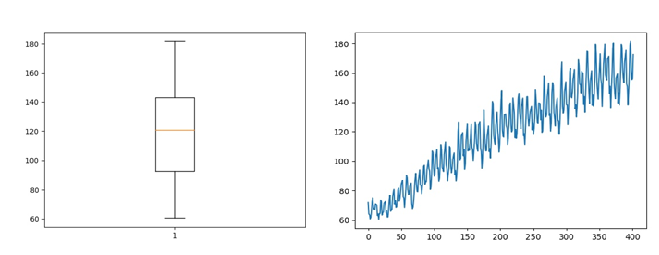


Рисунок 2 - Данные по использованию угля

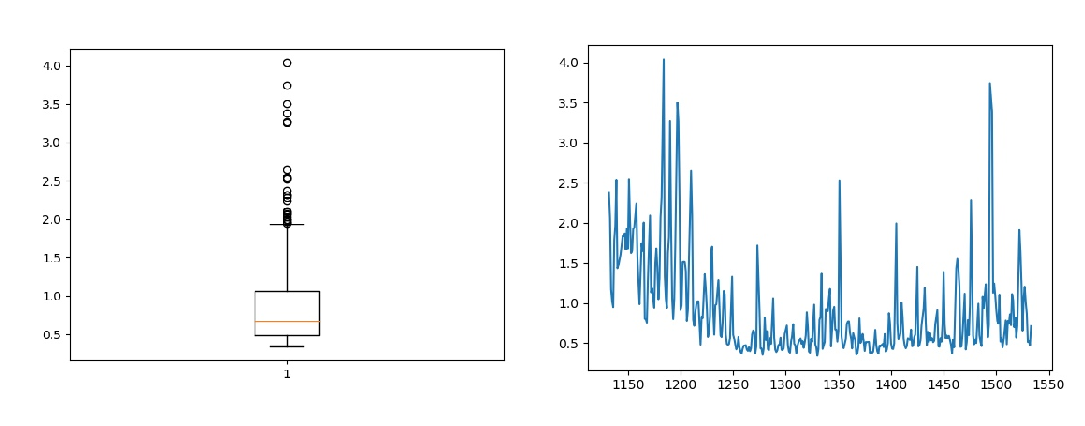


Рисунок 3 – Данные по использованию керосина

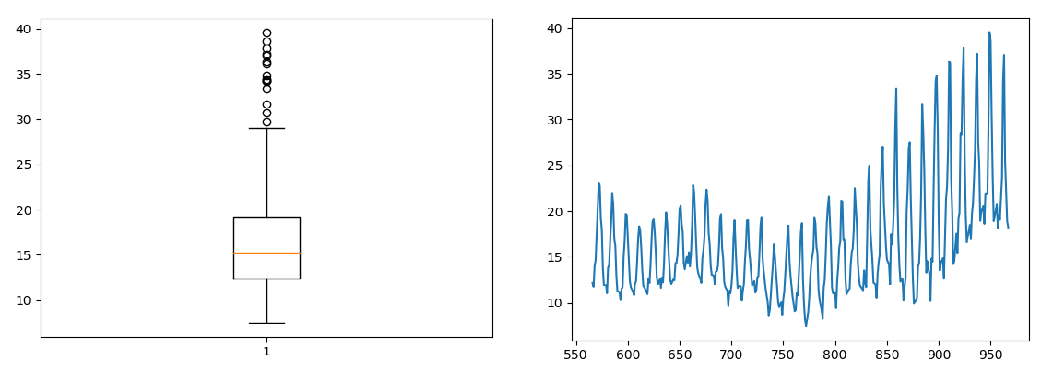


Рисунок 4 - Данные по использованию природного газа

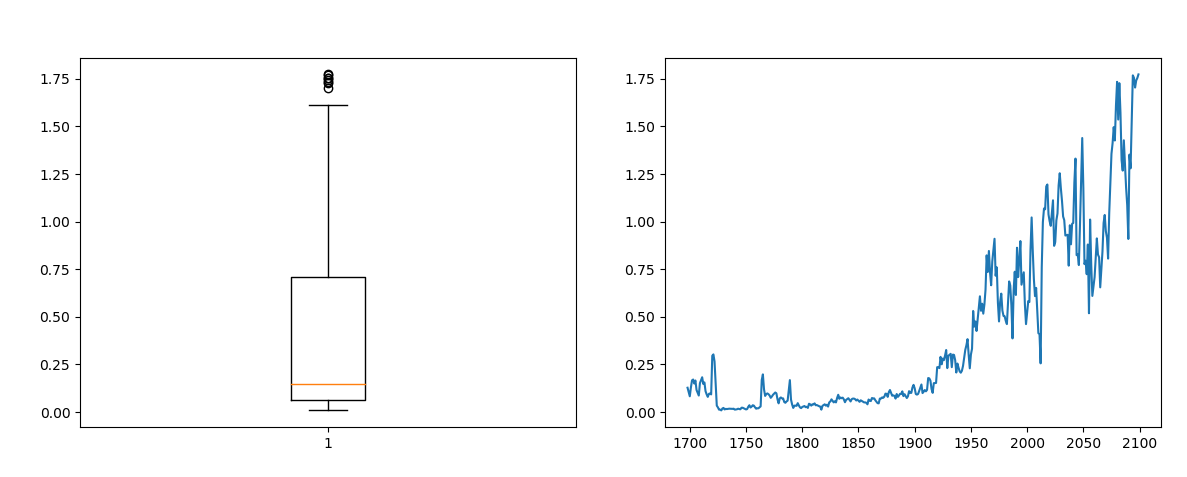


Рисунок 5 – Данные по использованию нефтяного кокса

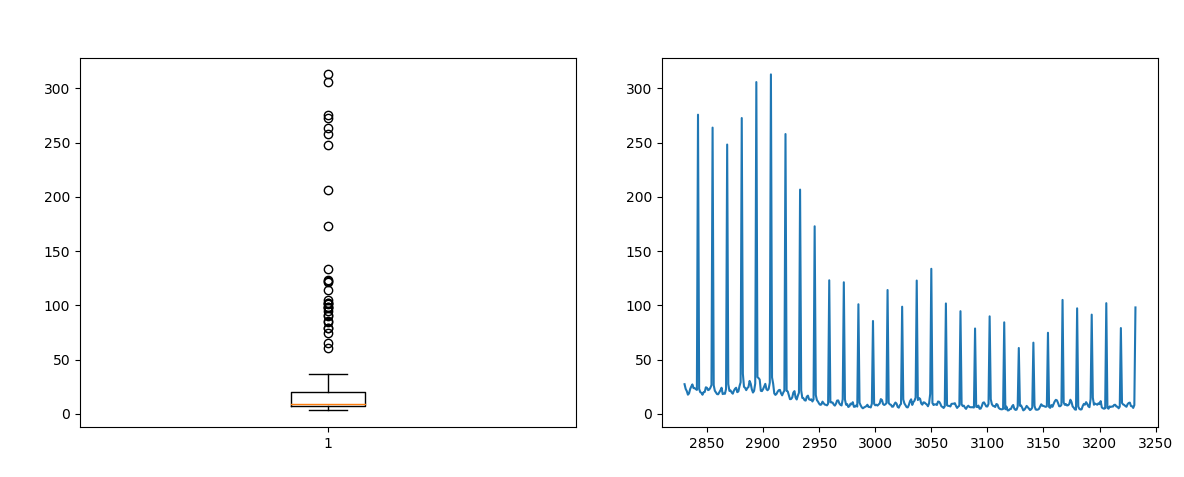


Рисунок 6 – Данные по использованию нефти

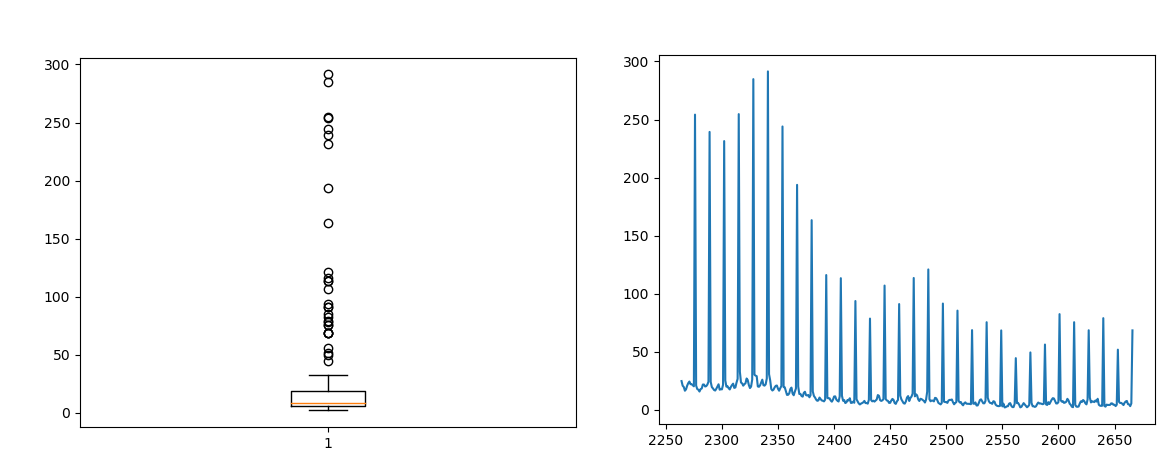


Рисунок 6 – Данные по использованию мазута

Построим гистограмму по всем источникам энергии (Приложение 4).

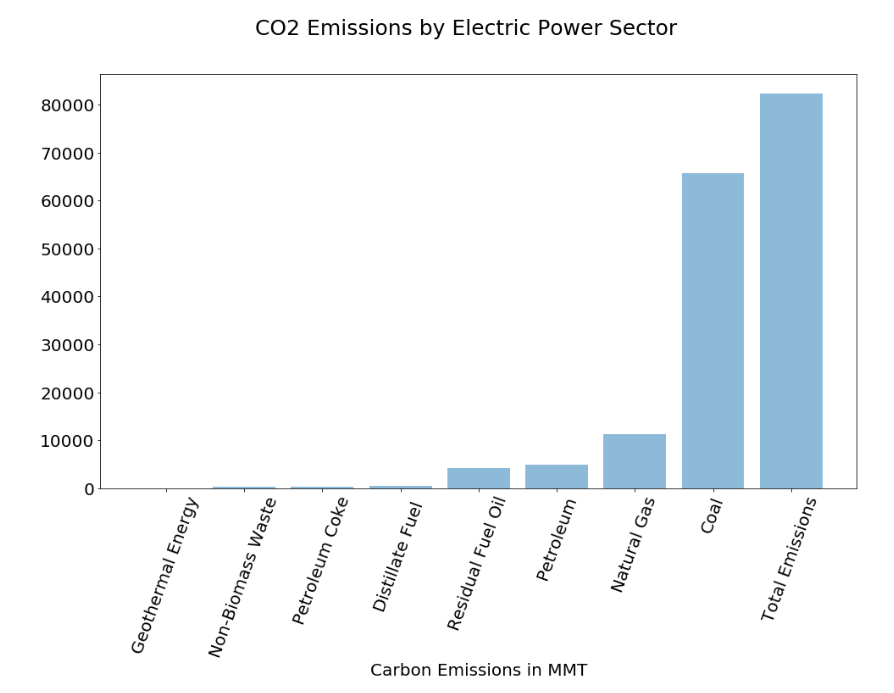


Рисунок 7 – Гистограмма по источникам энергии

По получившимся графикам можно отметить выбросы для всех типов кроме данных по использованию угля. Очистим данные от выбросов.

Для разработки модели и дальнейшего прогнозирования в данной работе будут использованы выбросы CO2 при использовании природного газа. Для начала уберем выбросы (Приложение 5).

Анализ стационарности по данным.

Создадим график набора данных временного ряда. Из этого получим представление об общей тенденции и сезонности данных. Затем используем статистический метод для оценки тенденции и сезонности набора данных. Если будут выявлены некоторые тенденции или сезонности, то удалим их из серии для преобразования нестационарного набора данных в стационарный, а оставшиеся значения будут дополнительно проанализированы.

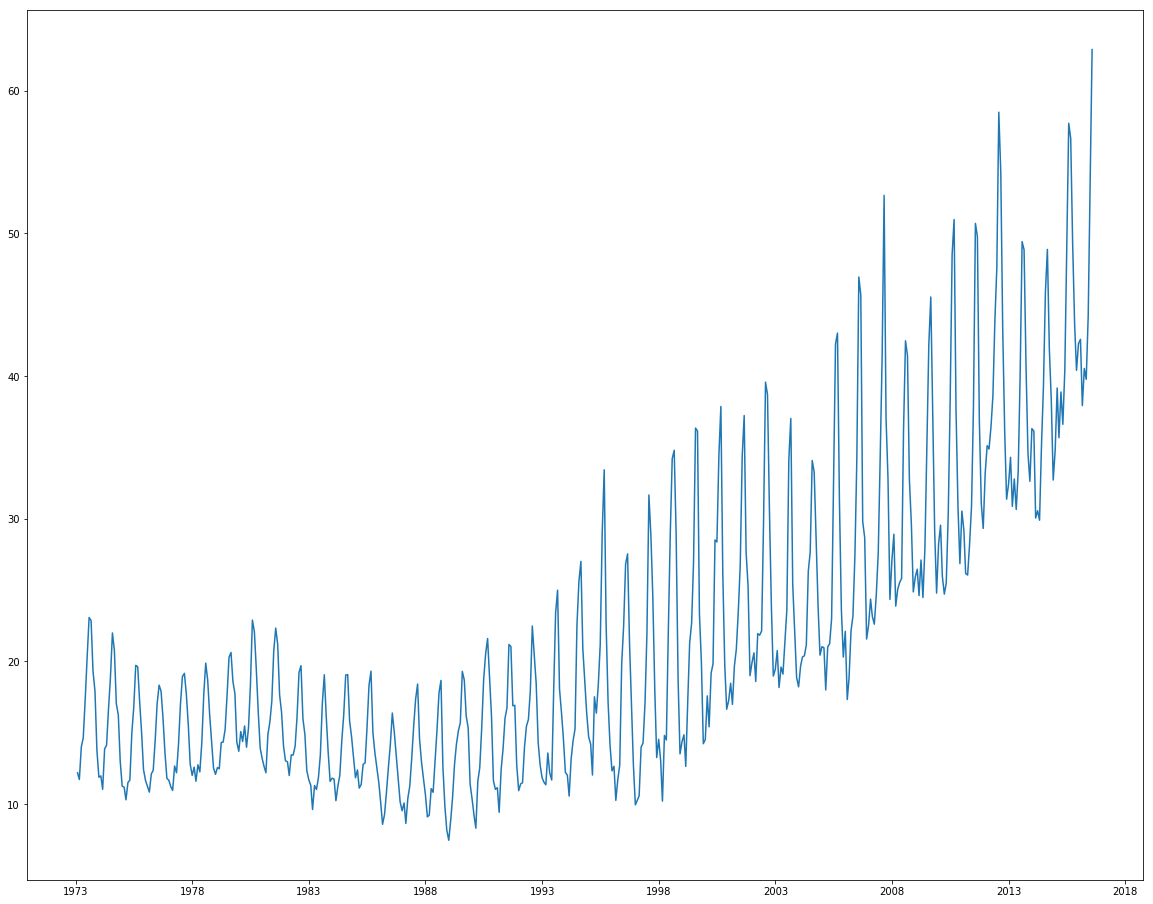


Рисунок 8 – Графический анализ стационарности

Из рисунков видно, что существует тенденция в наборе данных о выбросах CO2 с сезонными колебаниями. Таким образом, мы можем сделать вывод о том, что набор данных не является стационарным.

Проведем расширенный тест Дики – Фуллера. Формальный способ проверки стационарности набора данных заключается в построении графика скользящего среднего или скользящей дисперсии и рассмотрении, изменяется ли среднее значение ряда и дисперсии во времени. Этот подход будет обрабатываться методом TestStationaryPlot (). Второй способ проверить стационарность - использовать статистический тест (тест Дики-Фуллера). Нулевая гипотеза для теста состоит в том, что временные ряды нестационарны. Результаты теста сравнивают статистические и критические значения теста (предельное значение) на разных уровнях достоверности. Если результат теста будет меньше, чем критическое значение, то мы можем отклонить нулевую гипотезу и сказать, что ряд является стационарным. Этот метод будет обрабатываться методом TestStationaryAdfuller(), приведенным в Приложении 6. Результат – Рисунок 9.

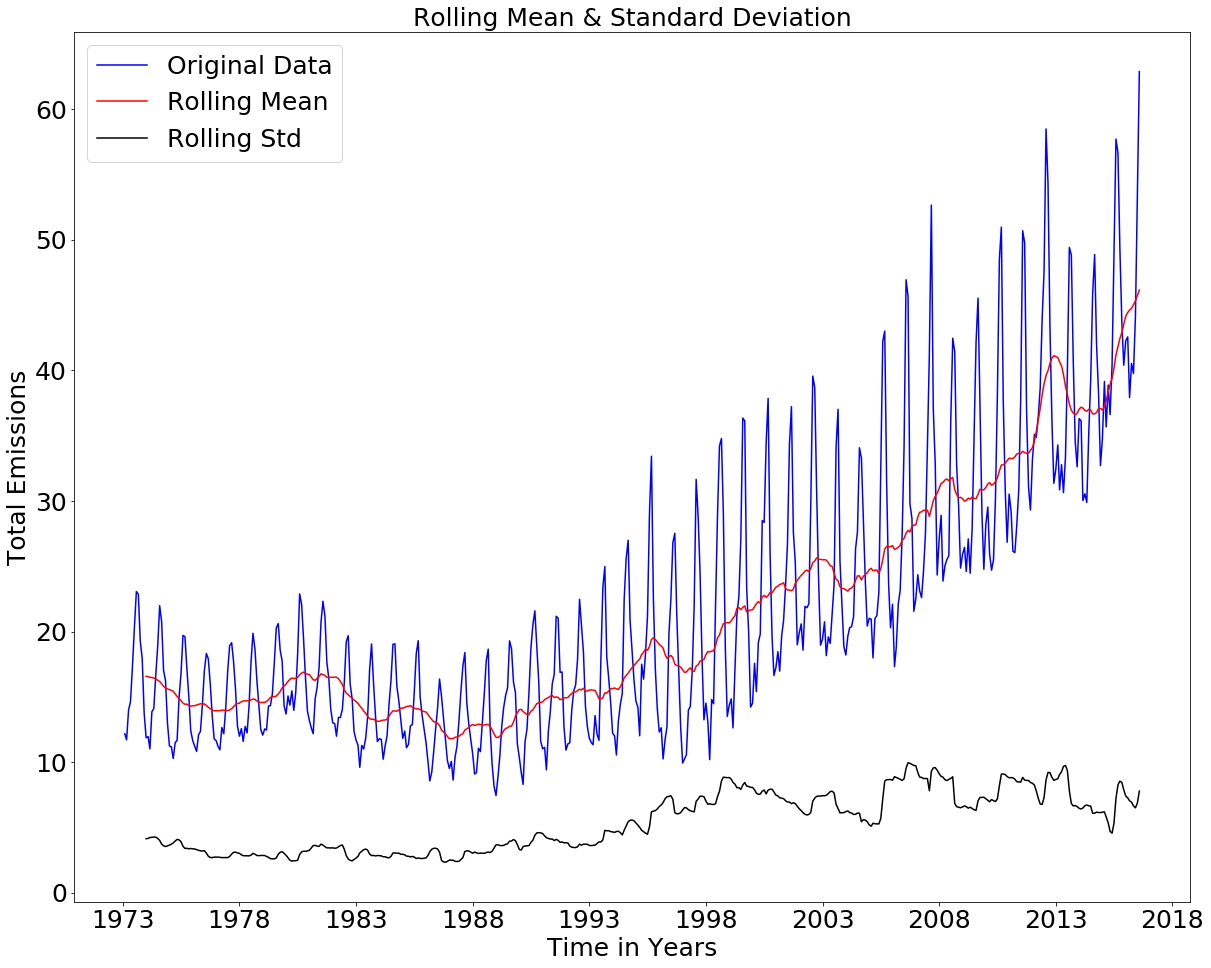


Рисунок 9 – График скользящего среднего

Среднее значение выбросов и изменение стандартного отклонения (черная линия) явно меняются со временем. Это показывает, что у серии есть тенденция. Это означает, что данные нестационарные. Кроме того, уровни достоверности составляют 90%, 95% и 99%. Следовательно, ряд нестационарный.

Трансформируем ряд нестационарных данных в стационарный. Наиболее распространенные методы, используемые для оценки или моделирования тренда, а затем удаления его из временного ряда:

* Агрегация - взятие среднего за период времени;
* Сглаживание - взятие скользящих средних;
* Полиномиальная подгонка - подгонка регрессионной модели.

Рассмотрим каждый из них.

Скользящее среднее. В этом методе мы берем среднее значение последовательных значений «k» в зависимости от частоты временных рядов (в данном случае 12 месяцев в год). Здесь мы возьмем среднее значение за последний 1 год (Приложение 7) (Рисунок 10).

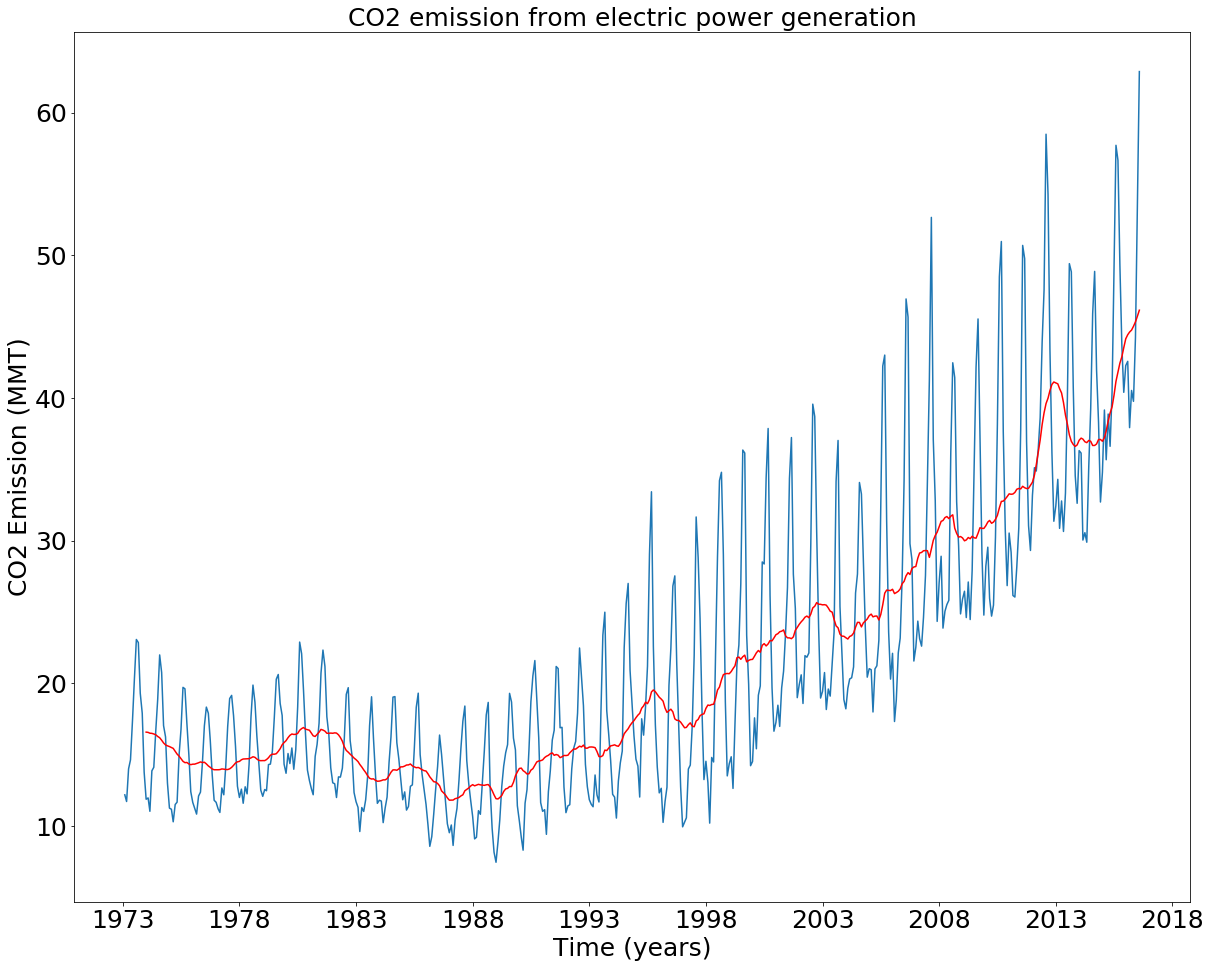


Рисунок 10 – Агрегация данных

Красная линия - скользящее среднее. Вычтем скользящее среднее из исходного ряда (Рисунок 11).

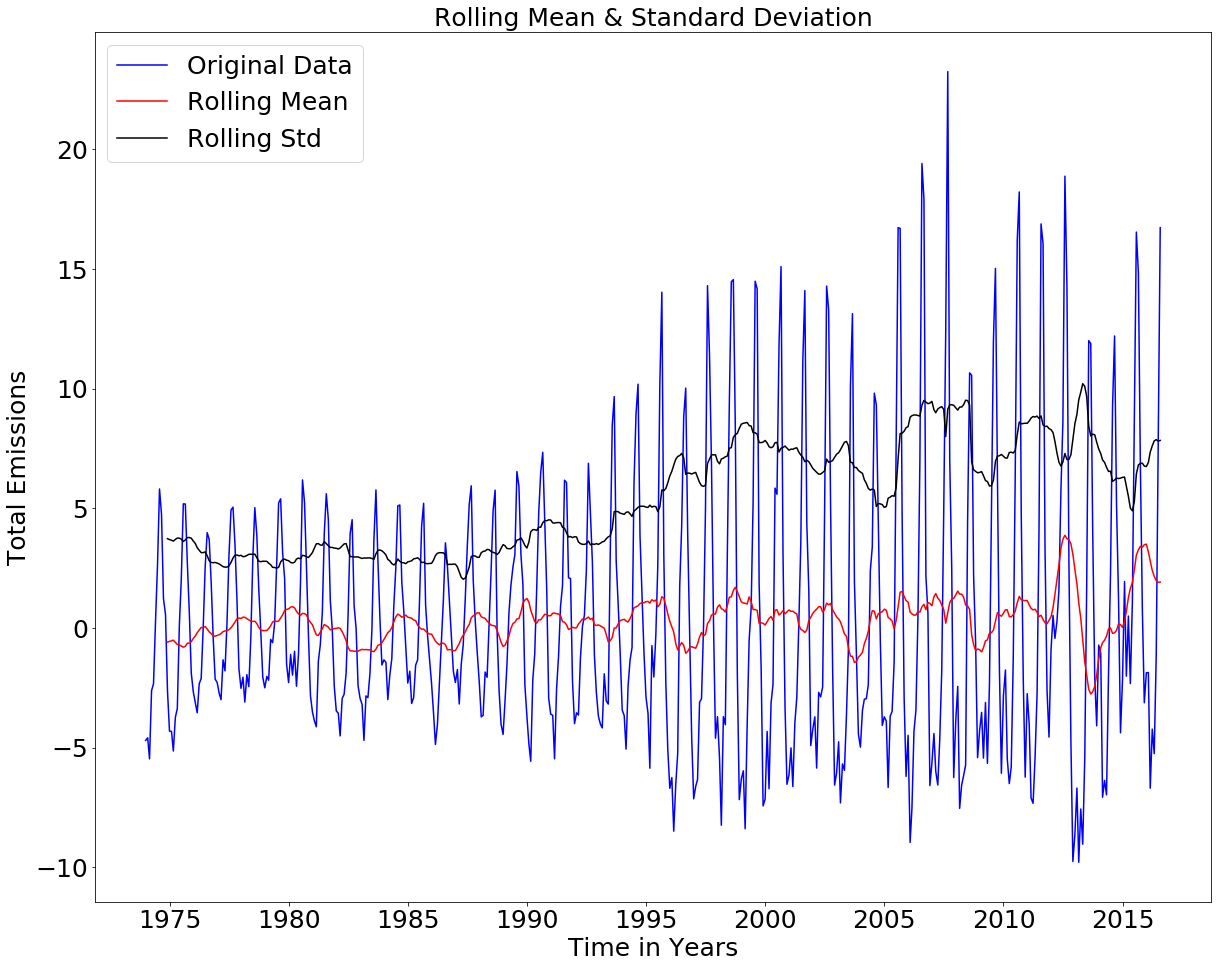


Рисунок 11 – Агрегация данных

Скользящие средние значения лишь немного изменяются. Можно с высокой уверенностью 99% утверждать, что набор данных является стационарным рядом.

Экспоненциально взвешенная скользящая средняя. Второй рассматриваемый метод - взять взвешенное скользящее среднее, где более поздние значения имеют больший вес (Приложение 8) (Рисунок 12).

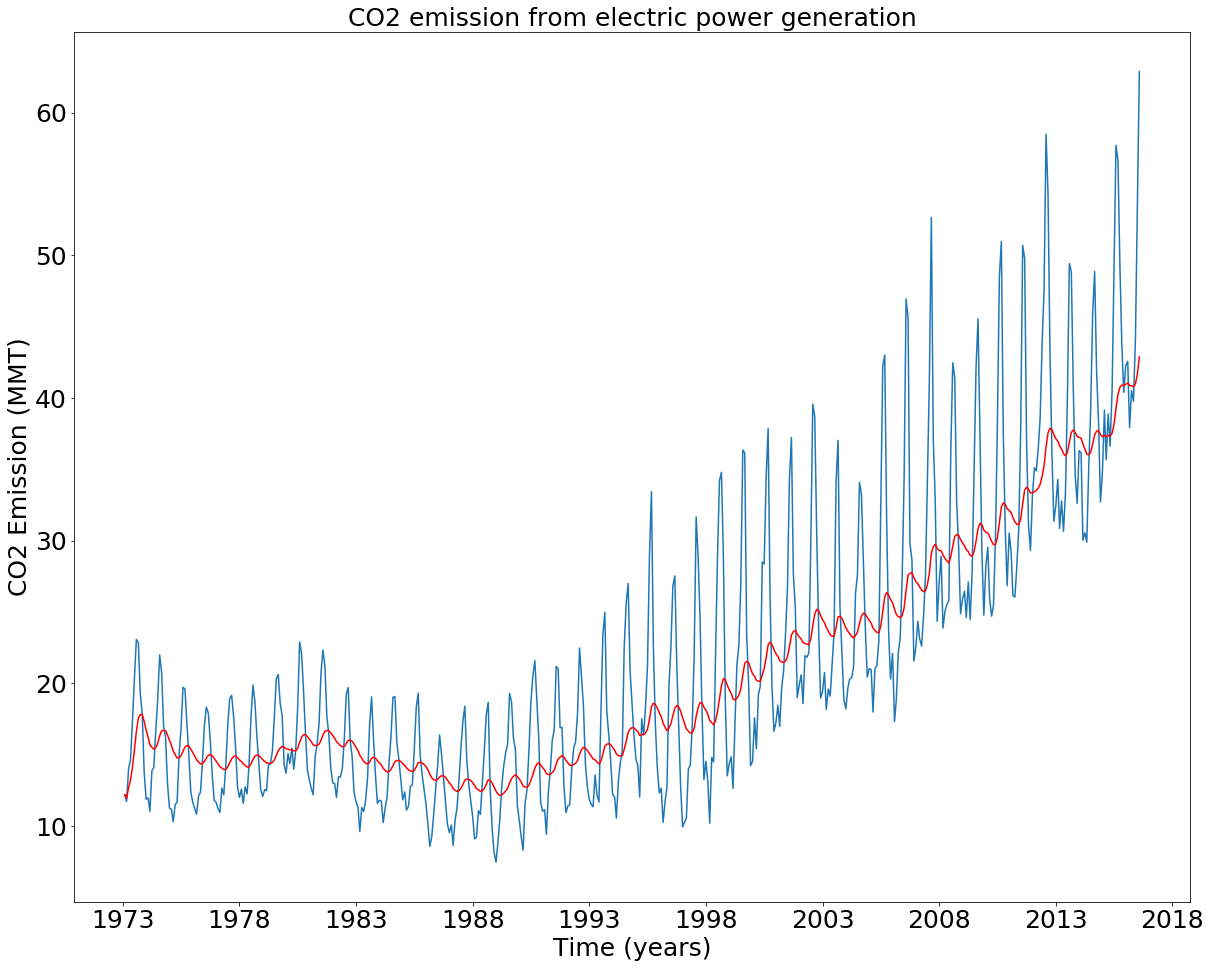


Рисунок 12 - Сглаживание

Красная линия - скользящее среднее. Вычтем скользящее среднее из исходного ряда (Рисунок 13).

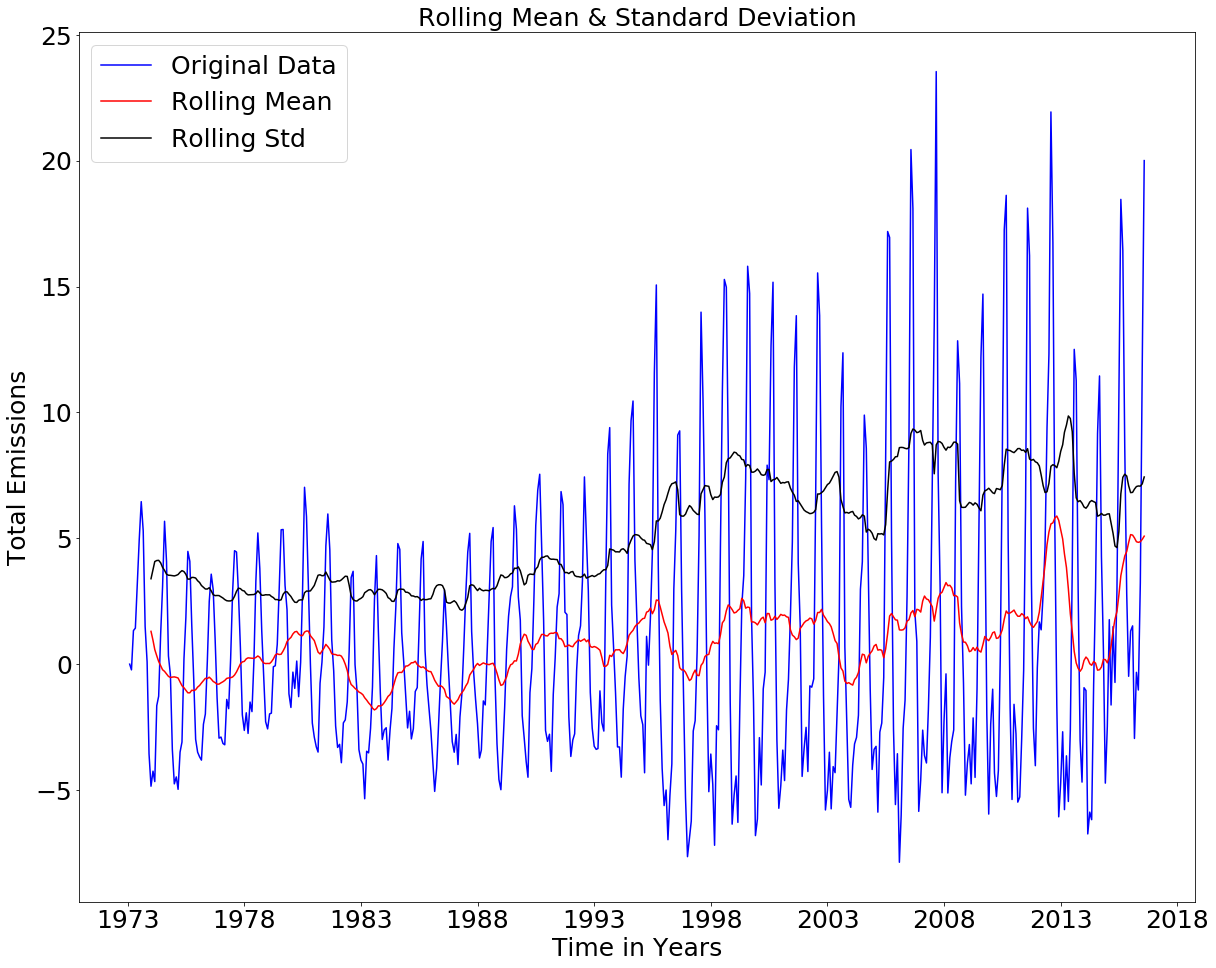


Рисунок 13 – Сглаживание

Этот временной ряд имеет меньшие вариации среднего и стандартного отклонения по сравнению с исходным набором данных. Кроме того, тестовая статистика меньше критического значения 5% и 10%, что лучше, чем в первоначальном случае. В этом случае мы можем сказать, что при уровне достоверности 95% ряд является стационарным рядом.

Дифференциация. Предпоследний рассматриваемый метод борьбы с трендом и сезонностью - берем разницу исходного наблюдения в конкретный момент с предыдущим моментом. Это хорошо работает для улучшения стационарности (Рисунок 14).

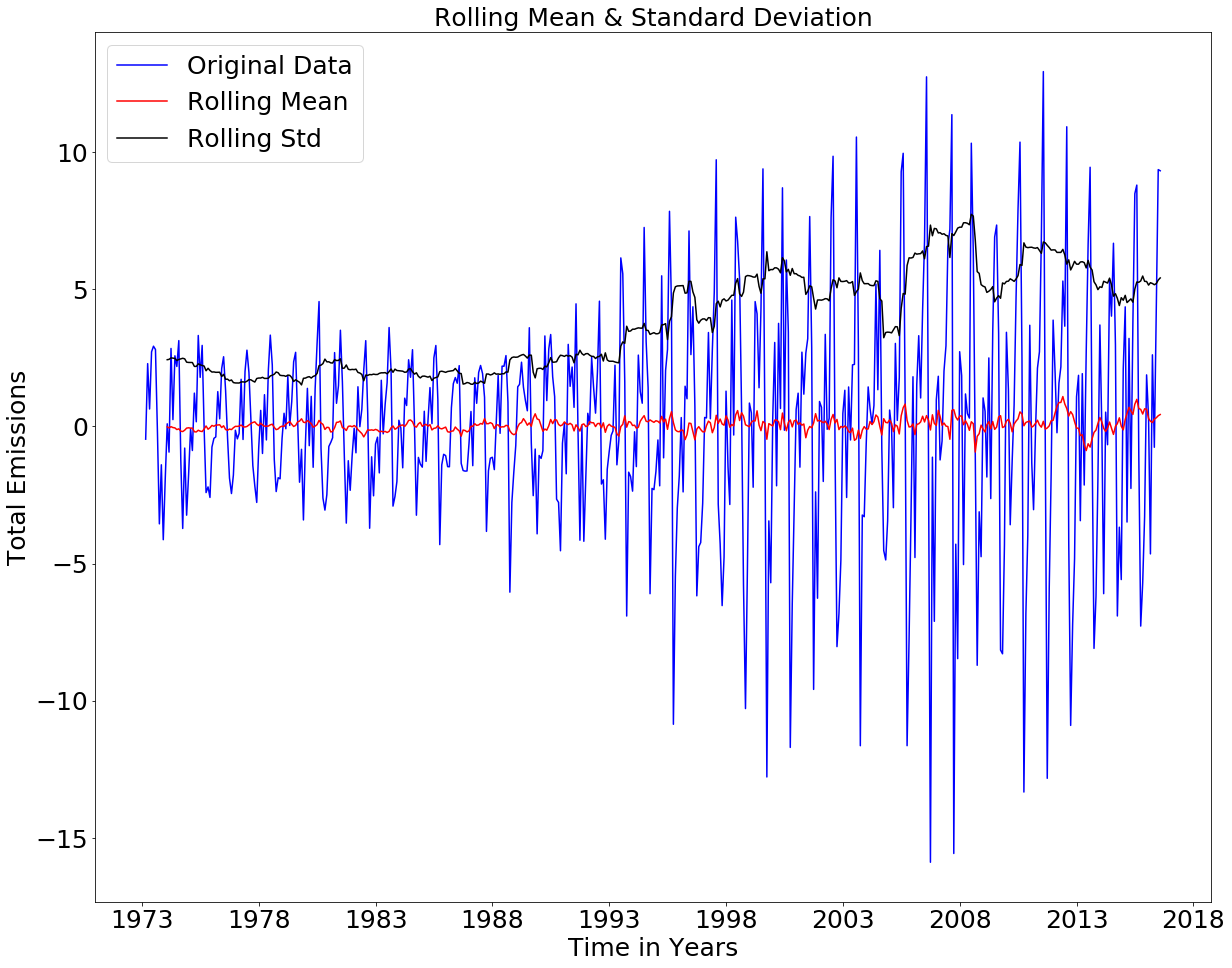


Рисунок 14 – Подгонка полинома

Первая разница значительно улучшает стационарность серии. Используем сезонную разницу для удаления сезонности данных и посмотрим, как это влияет на стационарность данных (Рисунок 15).

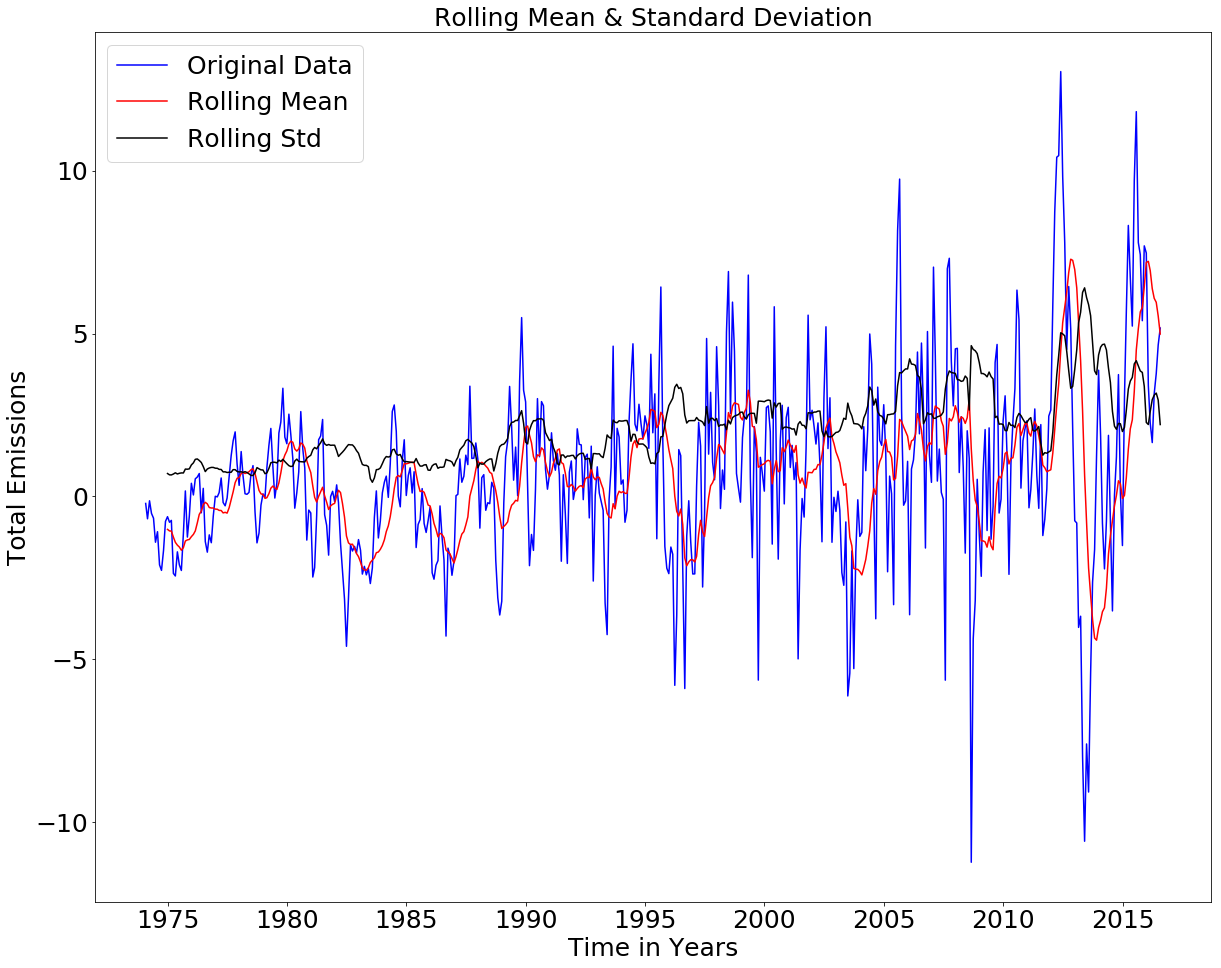


Рисунок 15 – Подгонка полинома

По сравнению с исходными данными сезонная разница также улучшает стационарность ряда. Следующим шагом является получение первой разницы сезонной разницы (Рисунок 16).

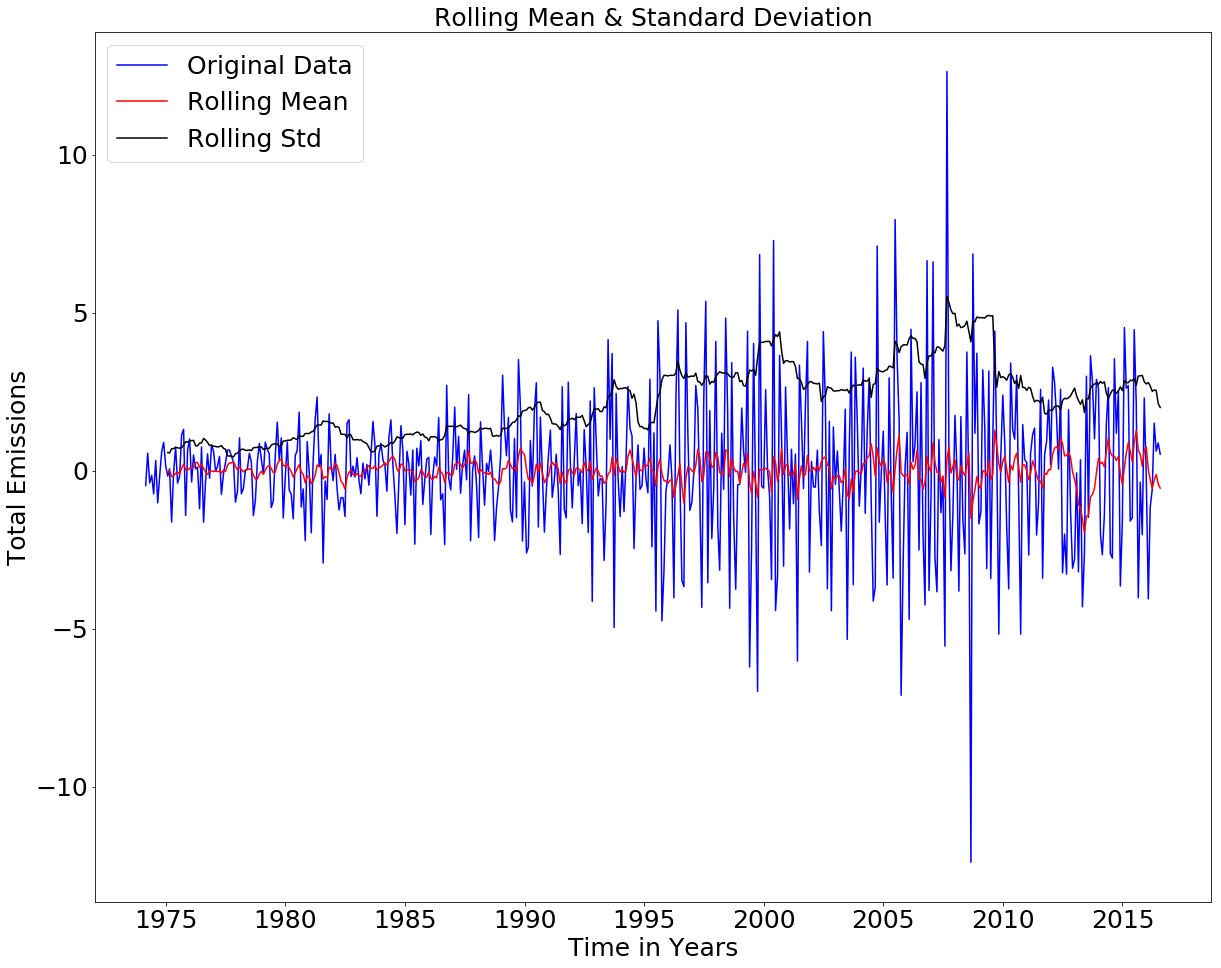


Рисунок 16 – Подгонка полинома

Теперь, если мы посмотрим на результаты теста и значение p, то с учетом первой сезонной разницы наш набор данных временных рядов станет стационарным.

Разложение. В этом методе это производится путем моделирования, как тренда, так и сезонности, и удаления их из модели (Приложение 9) (Рисунок 17).

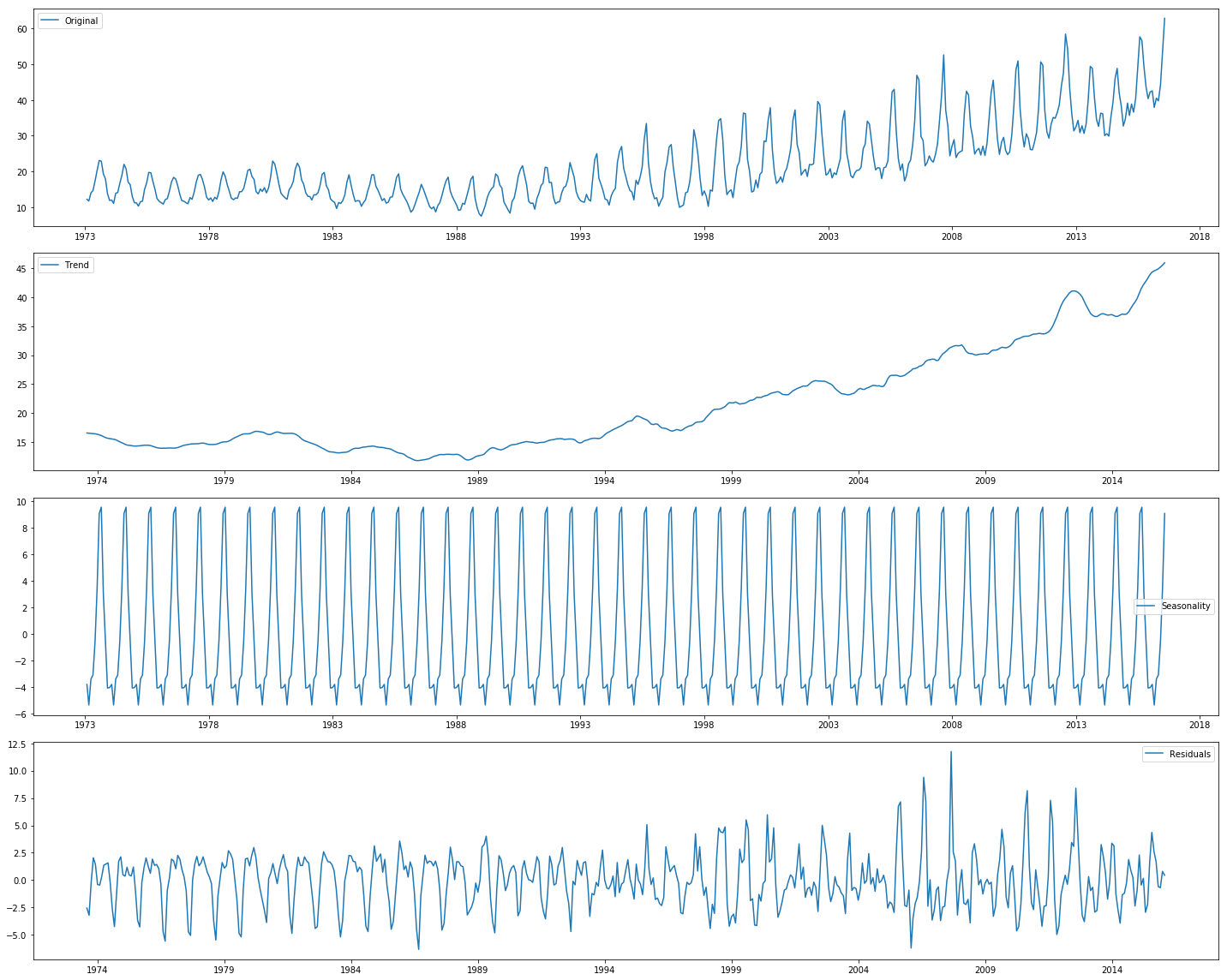


Рисунок 17 - Разложение

Здесь можно увидеть, что тренд и сезонность удалены из данных, и мы можем смоделировать остатки. Проверим стационарность остатков (Приложение 10) (Рисунок 18).

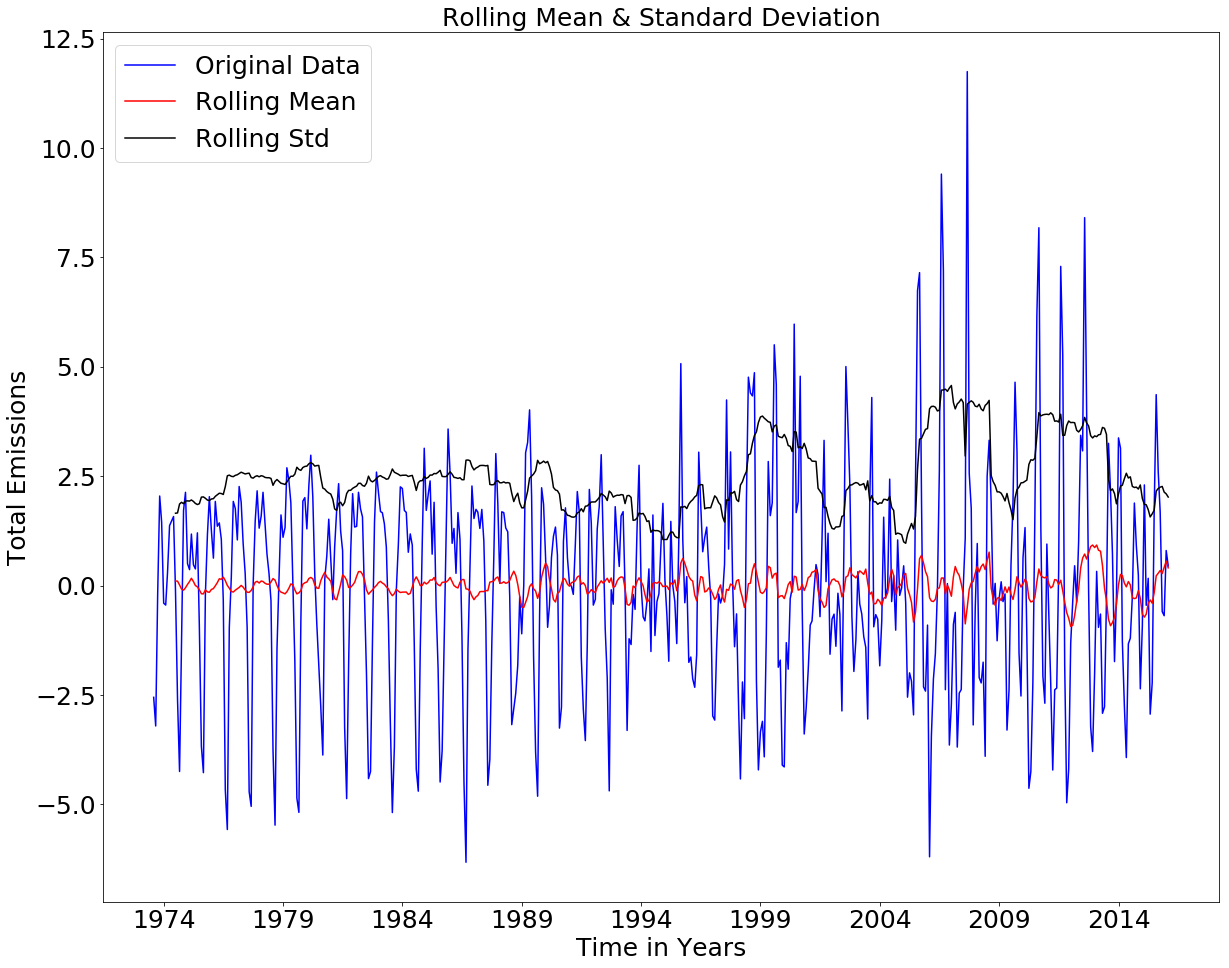


Рисунок 18 – Проверка стационарности оставшихся данных

Как можно увидеть по графику, теперь данные стационарны и можно приступить к построению модели.

# РАЗРАБОТКА МОДЕЛИ

Для построения моделей была выбрана модель ARIMA – интегрированная модель авторегрессии — скользящего среднего – модель и методология анализа временных рядов. Найти оптимальные параметры для моделей ARIMA с помощью графического метода отнимает много времени. Мы будем подбирать оптимальные значения параметров, используя метод поиска по сетке. Для каждой комбинации параметров мы подберем новую сезонную модель ARIMA с функцией SARIMAX() из модуля statsmodels и оценим ее. После этого мы выберем тот набор параметров, который даст наилучшую производительность по нашим интересующим критериям.

При оценке и сравнении статистических моделей мы будем использовать значение AIC, которое удобно использовать с моделями ARIMA в statsmodels.

SARIMAX (1, ​​1, 1) x (0, 1, 1, 12) дает самое низкое значение AIC = 2003,553. Поэтому мы будем считать, что это оптимальный вариант из всех комбинаций параметров. Теперь приступим к более глубокому анализу этой конкретной модели.

Столбец коэффицент (Рисунок 19) показывает важность каждого объекта и то, как каждый из них влияет на временной ряд. P> | z | столбец содержит значение важности. Здесь каждый вес имеет значение p, близкое к 0, поэтому целесообразно включить функции в нашу модель.

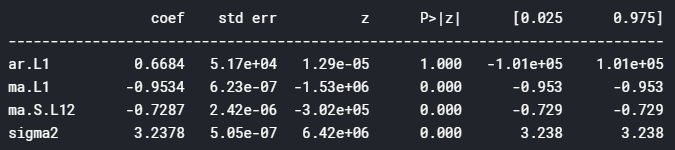


Рисунок 19 – Влияющие на временной ряд объекты

При подборе моделей ARIMA важно проводить диагностику модели, чтобы убедиться, что ни одно из допущений, сделанных моделью, не было нарушено. На Рисунке 20 показано распределение остаточных ошибок. Затем мы получаем график плотности значений остаточных ошибок, предполагая, что ошибки являются гауссовыми, но могут не центрироваться на нуле.

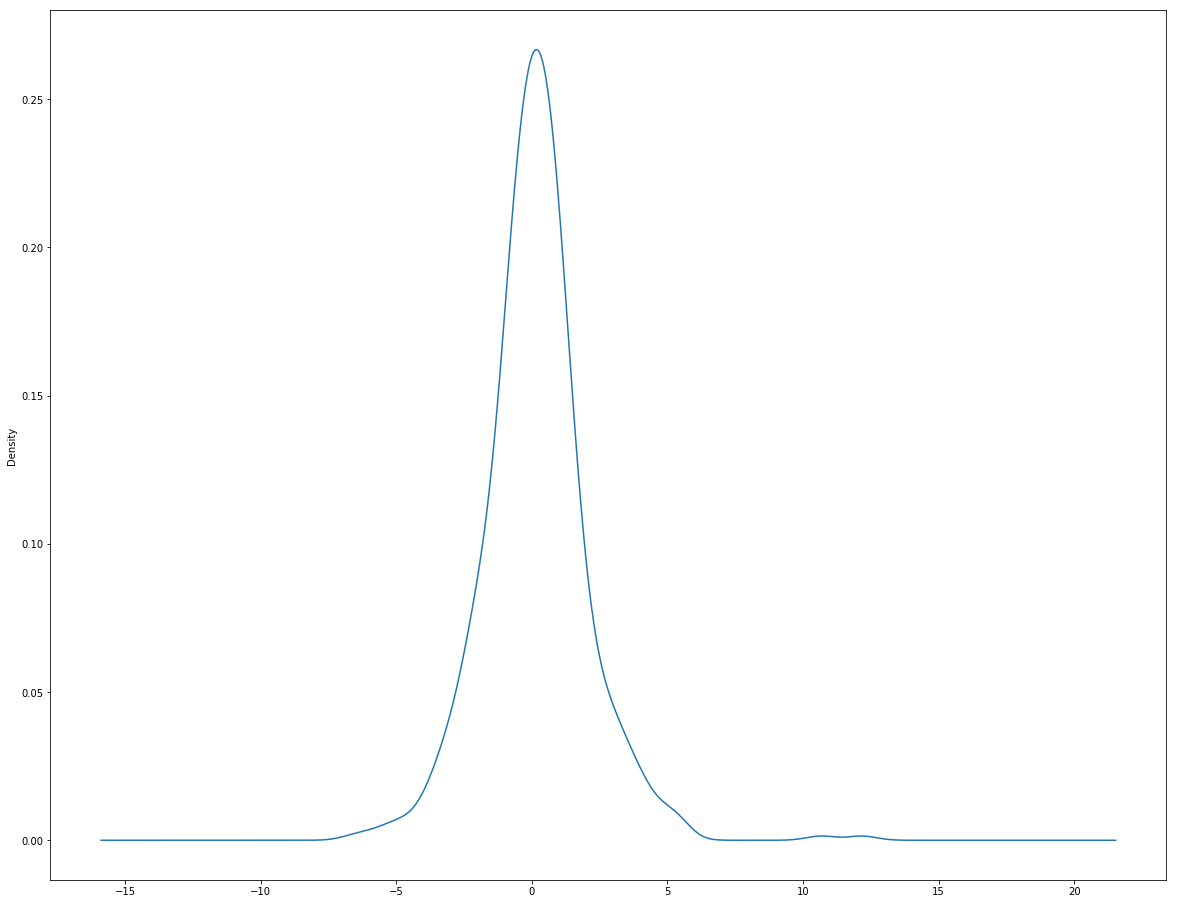


Рисунок 20 – Распределение остаточных ошибок

Воспользуемся plot\_diagnostics, чтобы сгенерировать диагностику модели и исследовать любое необычное поведение (Рисунок 21).

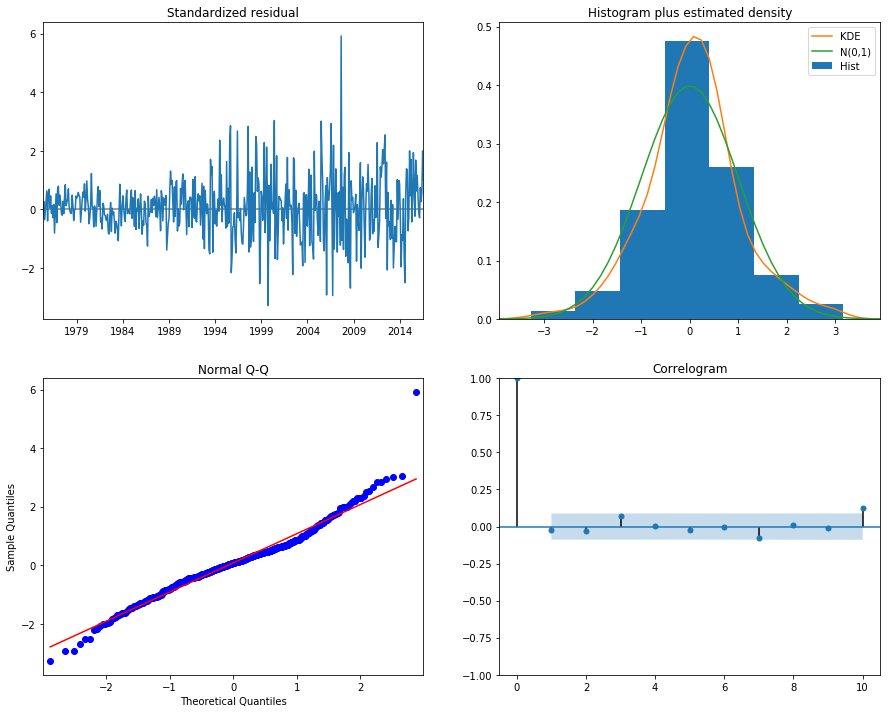


Рисунок 21 – Использование plot\_diagnostics

На верхнем правом графике линия KDE близко следует за линией N (0,1). Это означает, что остатки нормально распределены. Ошибки прогноза, однако, несколько отклоняются от прямой линии. Диаграмма Q-Q в левом нижнем углу показывает, что упорядоченное распределение остатков (синие точки) следует так же, как и нормальное распределение.

Остатки во времени (верхний левый график) не показывают явной сезонности и кажутся белым шумом. Это подтверждается графиком автокорреляции в правом нижнем углу, который показывает, что остатки временного ряда имеют низкую корреляцию с запаздывающими версиями самого себя.

Эти наблюдения позволяют нам сделать вывод, что наша модель обеспечивает удовлетворительное соответствие, которое может помочь нам понять данные нашего временного ряда и прогнозировать будущие значения.

Итак, теперь мы получили модель для нашего временного ряда, которую теперь можно использовать для составления прогнозов. Мы начнем с тестовых данных. Мы будем создавать прогнозы на один шаг вперед, что означает, что прогнозы в каждой точке генерируются с использованием полной истории до этой точки. Мы можем построить реальные и прогнозируемые значения временных рядов выбросов CO2, чтобы оценить, насколько хорошо подходит модель (Приложение 11) (Рисунок 22).

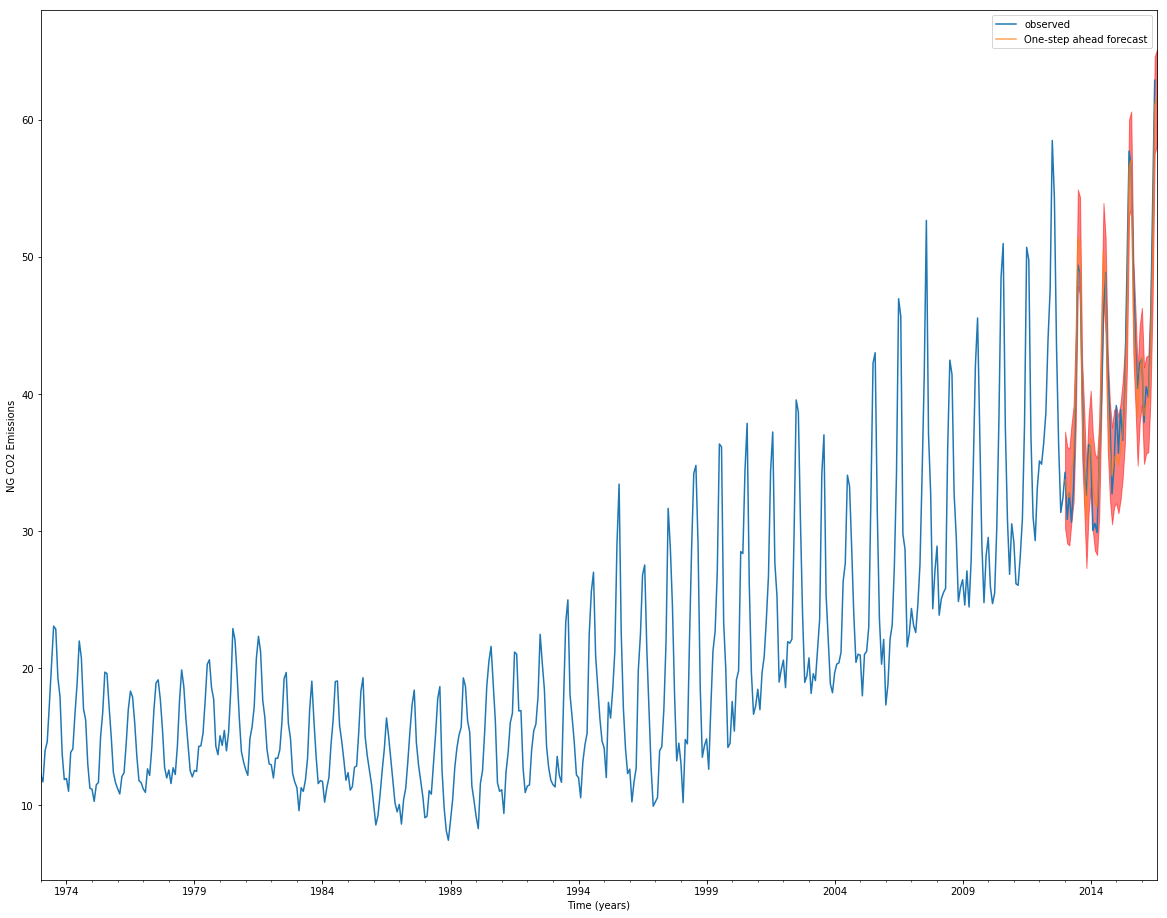


Рисунок 22 - Проверка прогноза на тестовых данных

В целом прогнозы очень похожи на истинные значения. Для оценки точности прогнозов используем среднеквадратичное отклонение (Приложение 12). Отклонение равно 4.09.

Исходя из графика наблюдаемых и прогнозируемых значений временного ряда, мы видим, что общие прогнозы точны, даже когда мы используем динамический прогноз. Все прогнозируемые значения (красная линия) близко соответствуют первоначальным наблюдаемым (синяя линия) данным и находятся в пределах доверительных интервалов нашего прогноза (Приложение 13) (Рисунок 23).

# C:\Users\Andrey\Desktop\CO2 Emission Forecast with Python (Seasonal ARIMA) _ Kaggle_files\__results___119_0.png

Рисунок 23 – Прогнозирование

Среднеквадратическая ошибка (MSE) прогноза составляет 14.39.

Корень от среднеквадратической ошибки (RMSE) прогноза: 3.7936.

# РАЗРАБОТКА СЕРВИСА

После анализа моделей и их улучшения и выбора модели был описан концепт сервиса, через который пользователь будет пользоваться аналитической моделью, в качестве языка был взят Python Django framework.

Сервис позволяет конечному пользователю выбирать параметр, а именно – значение количества выбросов на текущий год либо на некотором промежутке из нескольких лет. По этому входному параметру сервис сможет проанализировать и определить с высокой степенью надежности значение выбросов СО2 от электростанций, работающих на природном газе во всем мире на следующие 10 лет (Рисунок 24).

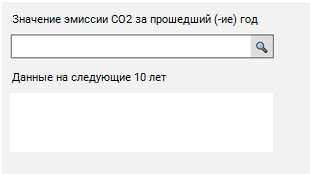


Рисунок 24 – Вид концептуального сервиса

После ввода параметров пользователь должен нажать клавишу Enter, после чего получит предсказание в окне ниже вида ГОД - ЗНАЧЕНИЕ. Значение будет измеряться в миллионах тонн.

# ЗАКЛЮЧЕНИЕ

В результате выполнения работы было рассмотрено, как извлечь набор данных CSV; как преобразовать набор данных во временные ряды; проверить, является ли временной ряд стационарным или нет с использованием графических методов и методов статистики теста Дикки-Фуллера; как преобразовать временной ряд в стационарный; как найти оптимальные параметры для построения модели SARIMA с использованием метода поиска по сетке, диагностики прогнозирования временных рядов, проверки прогнозирующей мощности, прогнозирования выбросов CO2 в течение 10 лет от выработки электроэнергии с использованием природного газа.

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Дубров А.М. Многомерные статистические методы /А.М. Дубров.– 2013. – 252 с.
2. Гришин А.Ф. Статистика: Учеб. пособие./ А.Ф. Гришин - 2003. – 240с.
3. Введение в многомерный анализ [Электронный ресурс]. Доступен по ссылке–https://habr.com/post/126810/ Последняя дата обращения: 18.01.19.
4. Репозиторий по курсу [Электронный ресурс]. Доступен по ссылке - https://github.com/SergeyMirvoda/MD-DA-2018/ Последняя дата обращения: 15.01.18.

# ПРИЛОЖЕНИЕ

1. **Импорт библиотек для Python**

%matplotlib inline

import numpy as np

import pandas as pd

import matplotlib.pylab

import matplotlib.pyplot as plt

from matplotlib.pylab import rcParams

rcParams['figure.figsize'] = 20, 16

import itertools

1. **Извлечение набора данных**

df = pd.read\_csv("../input/MER\_T12\_06.csv", parse\_dates=['YYYYMM'], index\_col='YYYYMM', date\_parser=dateparse)

1. **Группировка и визуализация данных по типу топлива**

Energy\_sources = ts.groupby('Description')

fig, ax = plt.subplots()

for desc, group in Energy\_sources:

group.plot(x = group.index, y='Value', label=desc,ax = ax, title='Carbon Emissions per Energy Source', fontsize = 20)

ax.set\_xlabel('Time(Monthly)')

ax.set\_ylabel('Carbon Emissions in MMT')

ax.xaxis.label.set\_size(20)

ax.yaxis.label.set\_size(20)

ax.legend(fontsize = 16)

1. **Построение** **гистограммы по всем источникам энергии**

fig = plt.figure(figsize = (16,9))

x\_label = cols

x\_tick = np.arange(len(cols))

plt.bar(x\_tick, CO2\_per\_source, align = 'center', alpha = 0.5)

fig.suptitle("CO2 Emissions by Electric Power Sector", fontsize= 25)

plt.xticks(x\_tick, x\_label, rotation = 70, fontsize = 20)

plt.yticks(fontsize = 20)

plt.xlabel('Carbon Emissions in MMT', fontsize = 20)

plt.show()

1. **Очистка данных от выбросов**

Emissions = ts.iloc[:,1:]

Emissions= Emissions.groupby(['Description', pd.TimeGrouper('M')])['Value'].sum().unstack(level = 0)

mte = Emissions['Natural Gas Electric Power Sector CO2 Emissions']

1. **Тест Дики -- Фуллера**

def TestStationaryPlot(ts):

rol\_mean = ts.rolling(window = 12, center = False).mean()

rol\_std = ts.rolling(window = 12, center = False).std()

plt.plot(ts, color = 'blue',label = 'Original Data')

plt.plot(rol\_mean, color = 'red', label = 'Rolling Mean')

plt.plot(rol\_std, color ='black', label = 'Rolling Std')

plt.xticks(fontsize = 25)

plt.yticks(fontsize = 25)

plt.xlabel('Time in Years', fontsize = 25)

plt.ylabel('Total Emissions', fontsize = 25)

plt.legend(loc='best', fontsize = 25)

plt.title('Rolling Mean & Standard Deviation', fontsize = 25)

plt.show(block= True )

def TestStationaryAdfuller(ts, cutoff = 0.01):

ts\_test = adfuller(ts, autolag = 'AIC')

ts\_test\_output = pd.Series(ts\_test[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations Used'])

for key,value in ts\_test[4].items():

ts\_test\_output['Critical Value (%s)'%key] = value

print(ts\_test\_output)

if ts\_test[1] <= cutoff:

print("Данные стационарны")

else:

print("Данные нестационарны")

1. **Агрегация данных**

moving\_avg = mte.rolling(12).mean()

plt.plot(mte)

plt.plot(moving\_avg, color='red')

plt.xticks(fontsize = 25)

plt.yticks(fontsize = 25)

plt.xlabel('Time (years)', fontsize = 25)

plt.ylabel('CO2 Emission (MMT)', fontsize = 25)

plt.title('CO2 emission from electric power generation', fontsize = 25)

plt.show()

1. **Сглаживание данных**

mte\_exp\_wighted\_avg = pd.ewma(mte, halflife=12)

plt.plot(mte)

plt.plot(mte\_exp\_wighted\_avg, color='red')

plt.xticks(fontsize = 25)

plt.yticks(fontsize = 25)

plt.xlabel('Time (years)', fontsize = 25)

plt.ylabel('CO2 Emission (MMT)', fontsize = 25)

plt.title('CO2 emission from electric power generation', fontsize = 25)

plt.show()

1. **Разложение данных**

from statsmodels.tsa.seasonal import seasonal\_decompose

decomposition = seasonal\_decompose(mte)

trend = decomposition.trend

seasonal = decomposition.seasonal

residual = decomposition.resid

plt.subplot(411)

plt.plot(mte, label='Original')

plt.legend(loc='best')

plt.subplot(412)

plt.plot(trend, label='Trend')

plt.legend(loc='best')

plt.subplot(413)

plt.plot(seasonal,label='Seasonality')

plt.legend(loc='best')

plt.subplot(414)

plt.plot(residual, label='Residuals')

plt.legend(loc='best')

plt.tight\_layout()

1. **Проверка стационарности оставшихся данных**

mte\_decompose = residual

mte\_decompose.dropna(inplace=True)

TestStationaryPlot(mte\_decompose)

TestStationaryAdfuller(mte\_decompose)

1. **Проверка прогноза на тестовых данных**

ax = mte['1973':].plot(label='observed')

pred.predicted\_mean.plot(ax=ax, label='One-step ahead forecast', alpha=.7)

ax.fill\_between(pred\_ci.index,

pred\_ci.iloc[:, 0],

pred\_ci.iloc[:, 1], color='r', alpha=.5)

ax.set\_xlabel('Time (years)')

ax.set\_ylabel('NG CO2 Emissions')

plt.legend()

plt.show()

1. **Расчет среднеквадратичного отклонения**

mte\_forecast = pred.predicted\_mean

mte\_truth = mte['2013-01-31':]

mse = ((mte\_forecast - mte\_truth) \*\* 2).mean()

print('Отклонение: {}'.format(round(mse, 2)))

1. **Прогнозирование**

ax = mte['1973':].plot(label='observed', figsize=(20, 15))

pred\_dynamic.predicted\_mean.plot(label='Dynamic Forecast', ax=ax)

ax.fill\_between(pred\_dynamic\_ci.index,

pred\_dynamic\_ci.iloc[:, 0],

pred\_dynamic\_ci.iloc[:, 1],

color='r',

alpha=.3)

ax.fill\_betweenx(ax.get\_ylim(),

pd.to\_datetime('2013-01-31'),

mte.index[-1],

alpha=.1, zorder=-1)

ax.set\_xlabel('Time (years)')

ax.set\_ylabel('CO2 Emissions')

plt.legend()

plt.show())

1. **Расчет среднеквадратичного отклонения 2**

mte\_forecast = pred\_dynamic.predicted\_mean

mte\_orginal = mte['2013-01-31':]

mse = ((mte\_forecast - mte\_orginal) \*\* 2).mean()

print('The Mean Squared Error (MSE) of the forecast is {}'.format(round(mse, 2)))

print('The Root Mean Square Error (RMSE) of the forcast: {:.4f}'

.format(np.sqrt(sum((mte\_forecast-mte\_orginal)\*\*2)/len(mte\_forecast))))