Каргашина Раиса Валентиновна

Итоговая аттестационная работа

Тема для исследования:

«Анализ посещаемости магазинов и прогнозирование временных рядов с использование данных о количестве чеков по сети магазинов «товары для дома»»

1. Знакомство с данными

- Для работы были использованы данные сети магазинов товары для дома
- Данные представляют собой количество чеков разрезе типов магазинов и регионов (выгружено с использованием Microsoft SQL Server Management Studio с базы данных магазинов Retail Analytics) *(Рис.1)*

- *Рис. 1*

```
select
t1.store_cod,t.[Дата чека],t.[Количество чеков], t1.store_square, t1.square_format,
case
when RegionalOfficeID like '%Москва%' then 'Москва'
else RegionalOfficeID
end as RegionalOffice


from [dmnd].[v_report_receipt_count] as t
join [dmnd].[v_dct_stores] as t1
on t.Магазин = t1.store_cod
where --t.[Дата чека]>='20200101' and t.[Дата чека]<='20201231'
t.[Дата чека] between '2019-01-01' and '2020-12-31'
and t1.[square_format] = 'Гипермаркет'
order by [Дата чека],[RegionalOffice]
```

100 %

Результаты | Сообщения

	store_cod	Дата чека	Количество чеков	store_square	square_format	RegionalOffice
1	714	2019-01-01	2	7995.0000000000000000	Гипермаркет	Дон
2	715	2019-01-01	1	13438.0000000000000000	Гипермаркет	Дон
3	715	2019-01-01	1	13438.0000000000000000	Гипермаркет	Дон
4	714	2019-01-01	1	7995.0000000000000000	Гипермаркет	Дон
5	733	2019-01-01	1	7107.0000000000000000	Гипермаркет	Москва
6	732	2019-01-01	2	8100.0000000000000000	Гипермаркет	Москва
7	720	2019-01-01	1	10936.0000000000000000	Гипермаркет	Москва
8	725	2019-01-01	1	8951.0000000000000000	Гипермаркет	Москва
9	728	2019-01-01	1	6853.0000000000000000	Гипермаркет	Москва
10	732	2019-01-01	1	8100.0000000000000000	Гипермаркет	Москва
11	711	2019-01-01	1	11309.0000000000000000	Гипермаркет	Москва
12	724	2019-01-01	1	9822.0000000000000000	Гипермаркет	Москва
13	725	2019-01-01	1	8951.0000000000000000	Гипермаркет	Москва
14	724	2019-01-01	2	9822.0000000000000000	Гипермаркет	Москва

Рис.2

Целью работы является проведение анализа временного ряда и дальнейшее его прогнозирование с помощью моделей SARIMA, ETS Model (Exponential Smoothing).

Для этих целей был выбран формат магазинов гипермаркет и сформирован дата-фрейм для дальнейшего анализа временного ряда и построения соответствующихх графиков.(Рис. 2)

date_recipt	BU	quantity_recipt	store_square	square_format	RegionalOffice	year	month	day
2019-01-01	714	2	7995.0	Гипермаркет	Дон	2019	1	1
2019-01-01	715	1	13438.0	Гипермаркет	Дон	2019	1	1
2019-01-01	715	1	13438.0	Гипермаркет	Дон	2019	1	1
2019-01-01	714	1	7995.0	Гипермаркет	Дон	2019	1	1
2019-01-01	733	1	7107.0	Гипермаркет	Москва	2019	1	1
...	...	...	...	...	...	...	...	...
2020-12-31	713	110	13035.0	Гипермаркет	Юг	2020	12	31
2020-12-31	713	117	13035.0	Гипермаркет	Юг	2020	12	31
2020-12-31	803	80	9571.0	Гипермаркет	Юг	2020	12	31
2020-12-31	803	85	9571.0	Гипермаркет	Юг	2020	12	31
2020-12-31	803	57	9571.0	Гипермаркет	Юг	2020	12	31

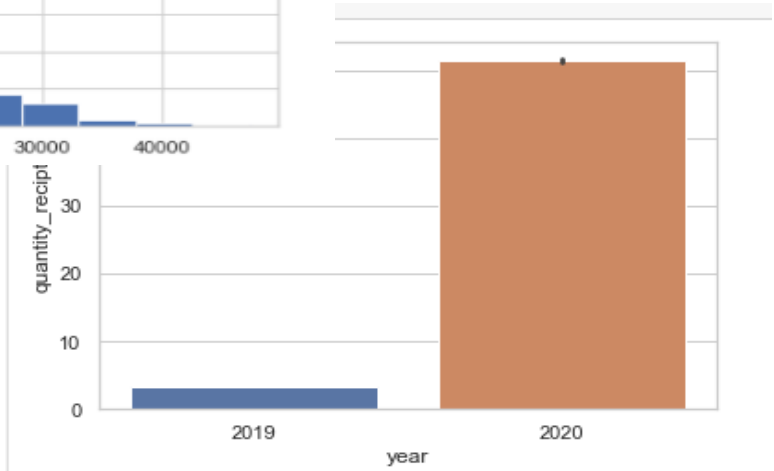
196249 rows × 8 columns

Исследование данных

- Основные статистики датафрема
- Count – количество значений в датасете.
- Mean – среднее значение по ряду.
- Std – стандартная ошибка.
- Min – минимальное значение.
- 25%, 50%, 75% - значения границ квартилей. 50% - это не что иное, как медиана. В нормально распределенных не смещенных данных, как правило, медиана и среднее значение близки друг

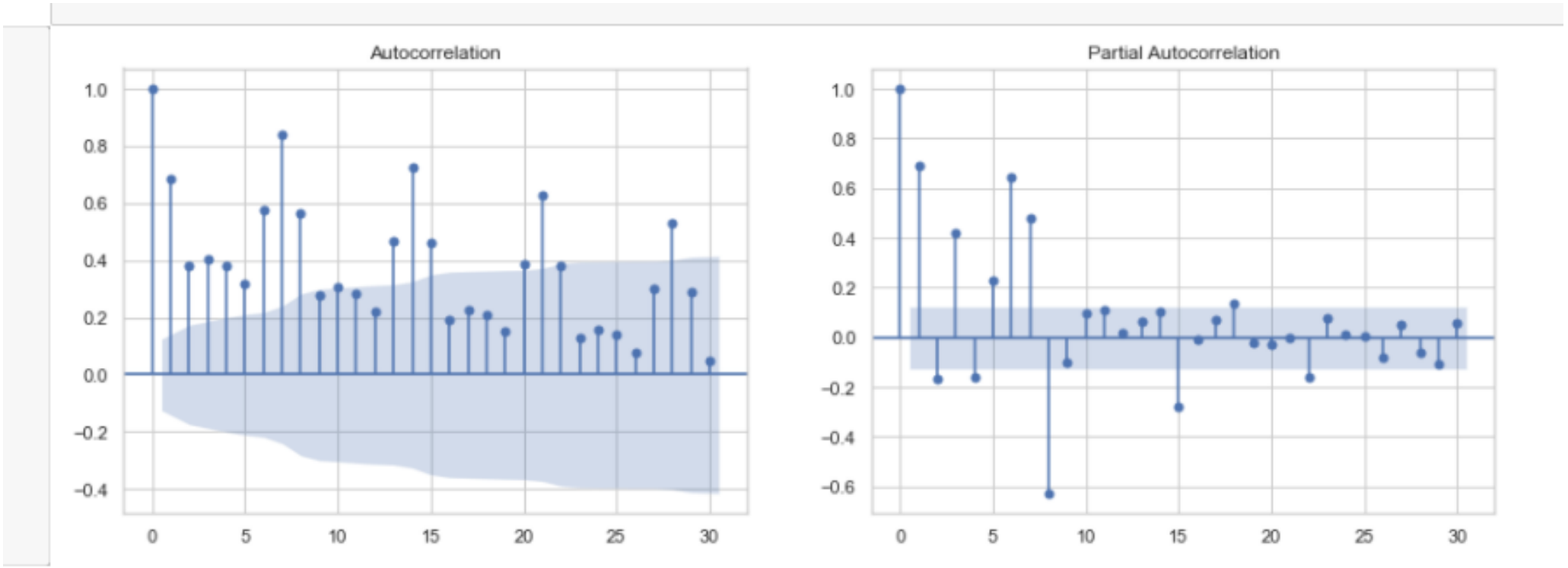
	BU	quantity_recipt	store_square	year	month	day
count	196249.000000	196249.000000	196249.000000	196249.000000	196249.000000	196249.000000
mean	733.680870	31.157519	9685.898308	2019.578296	6.697991	15.799637
std	39.926486	36.549022	2820.466529	0.493833	3.590786	8.775565
min	201.000000	1.000000	0.000000	2019.000000	1.000000	1.000000
25%	716.000000	3.000000	7483.000000	2019.000000	3.000000	8.000000
50%	727.000000	17.000000	9571.000000	2020.000000	7.000000	16.000000
75%	737.000000	49.000000	11697.000000	2020.000000	10.000000	23.000000
max	811.000000	769.000000	16746.000000	2020.000000	12.000000	31.000000

- Строим гистограмму и выполняем визуализацию
- В данном случае кол-во чеков по годам говорит нам о том что либо в компании был рост магазинов в 2020 году либо у нас не хватает данных по 2019 г.

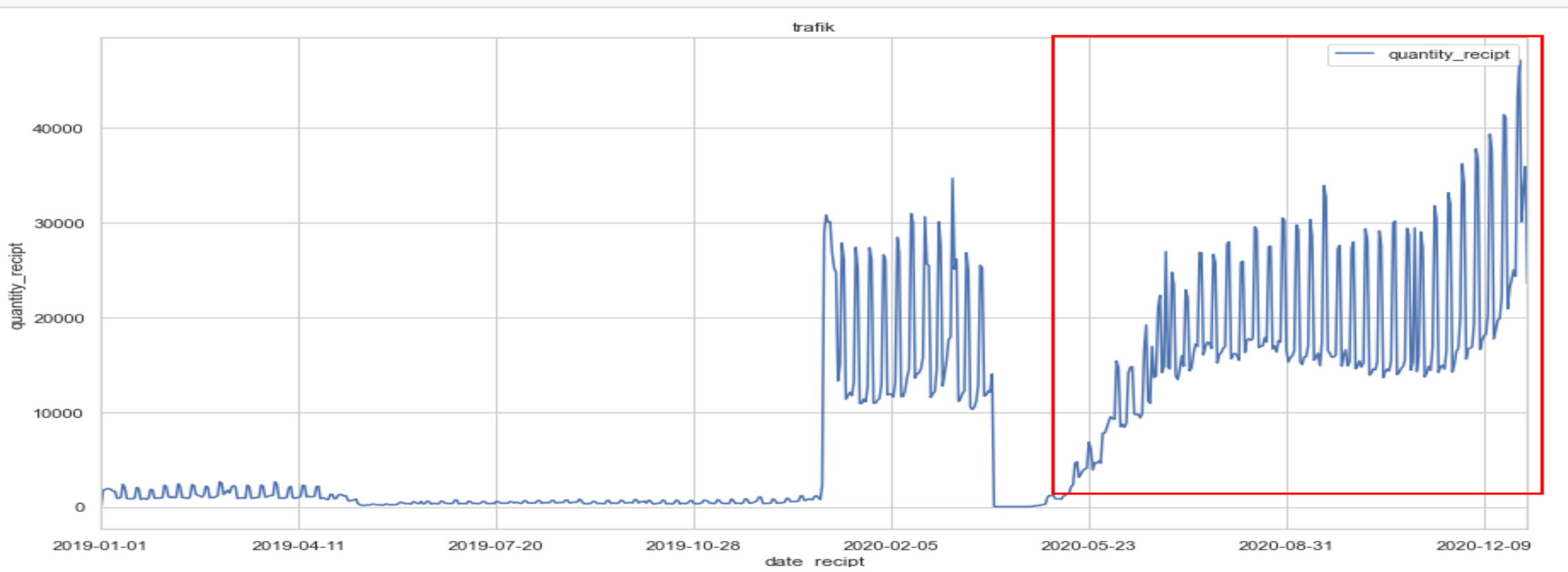


Строим графики автокорреляции

из данных графиков видно ряд нестационарен и есть зависимость от тренда и сезонности



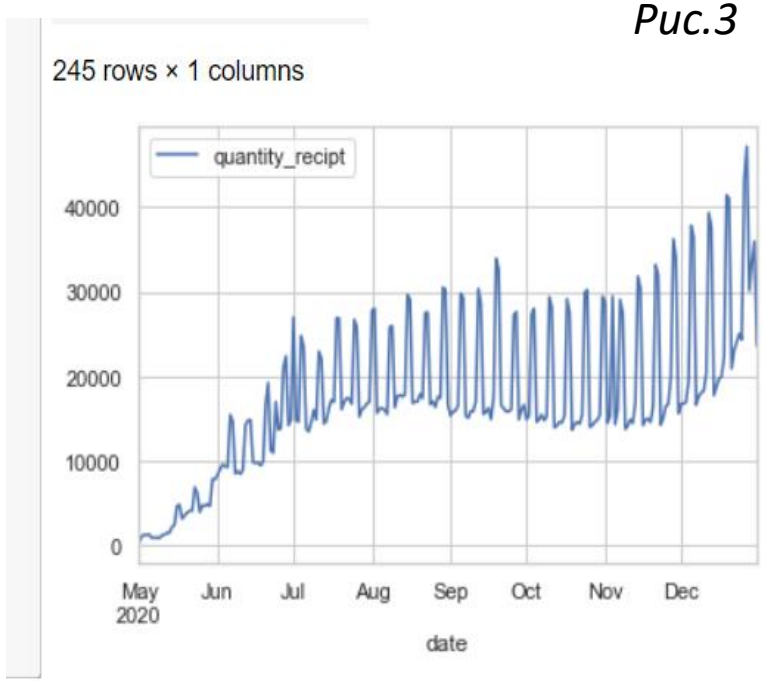
Строим временной ряд по DatetimeIndex



В результате данного временного ряда понимаем что для исследования мы можем взять только данные после 23.05.2020

Берем данные после 05.2020 *(рис.3)* и работаем с ними
Выполняем ETS декомпозицию *(рис4.)*

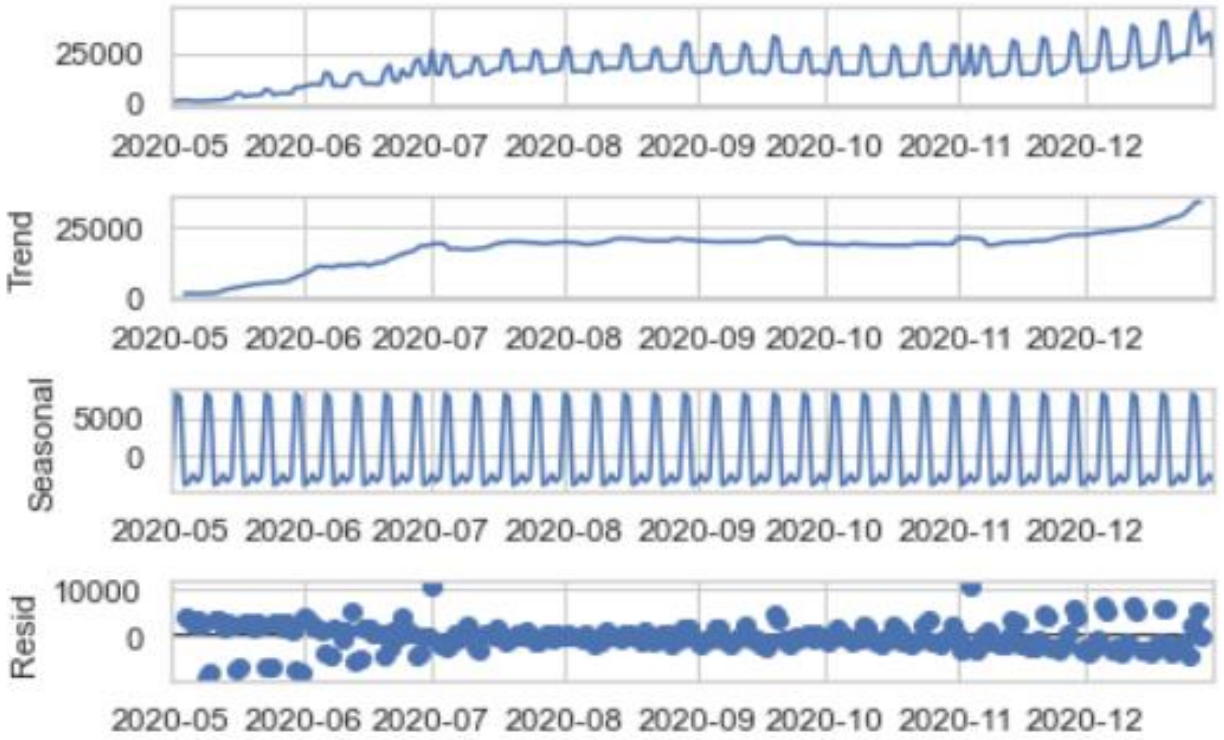
Рис.3



Составляющие сезонной декомпозиции временного ряда
Мы видим что имеется

- выраженный тренд и
- сезонность
- остаток составляет белый шум

Рис.4



SARIMA

Для подбора параметров к данной модели была использована функция auto_arima(). В результате была подобрана модель вида SARIMAX(0, 1, 2)x(0, 0, 1, 12).

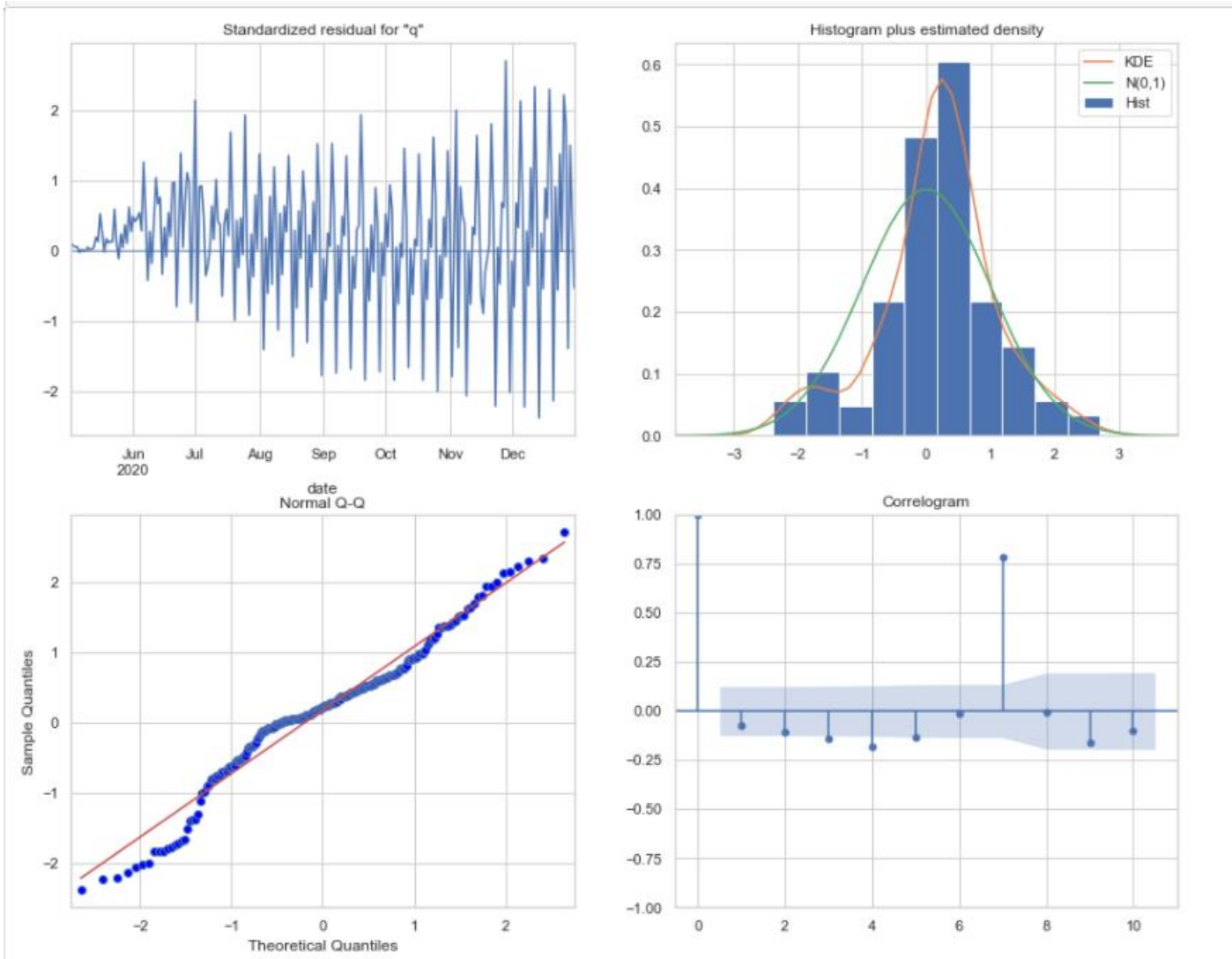
SARIMAX Results

Dep. Variable:	y	**No. Observations:**	245
Model:	SARIMAX(0, 1, 2)x(0, 0, [1], 12)	**Log Likelihood**	-2418.836
Date:	Sat, 26 Mar 2022	**AIC**	4847.673
Time:	23:14:54	**BIC**	4865.159
Sample:	0	**HQIC**	4854.715
	- 245		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	103.7245	31.109	3.334	0.001	42.753	164.696
ma.L1	-0.3653	0.078	-4.703	0.000	-0.518	-0.213
ma.L2	-0.4382	0.068	-6.438	0.000	-0.572	-0.305
ma.S.L12	-0.5749	0.053	-10.778	0.000	-0.679	-0.470
sigma2	2.335e+07	0.000	2.2e+11	0.000	2.34e+07	2.34e+07

Ljung-Box (L1) (Q):	0.31	**Jarque-Bera (JB):**	15.59
Prob(Q):	0.58	**Prob(JB):**	0.00
Heteroskedasticity (H):	4.19	**Skew:**	-0.47
Prob(H) (two-sided):	0.00	**Kurtosis:**	3.80

Построим диагностические графики для полученной модели.



Тренировочные и тестовые наборы

Чтобы иметь возможность более качественно оценивать модель, разделим данные на тренировочные и тестовые.

```
train = df.iloc[:len(df)-12]
train.head()
```

date	quantity_recipt
2020-05-01	262
2020-05-02	976
2020-05-03	1174
2020-05-04	1156
2020-05-05	1220

```
test = df.iloc[len(df)-12:]
test.head()
```

date	quantity_recipt
2020-12-20	41038
2020-12-21	20881
2020-12-22	23060
2020-12-23	23971
2020-12-24	24992

Обучаем SARIMA на тренировочных данных и сравниваем полученные результаты

```
#Обучаем SARIMA(0, 1, 2)(0, 0, 1, 12)
model = SARIMAX(train['quantity_recipt'], order=(0, 1, 2), seasona
results = model.fit()
results.summary()
```

SARIMAX Results

Dep. Variable:	quantity_recipt	No. Observations:	233
Model:	SARIMAX(0, 1, 2)x(0, 0, [1], 12)	Log Likelihood	-2299.210
Date:	Sun, 27 Mar 2022	AIC	4606.420
Time:	11:26:17	BIC	4620.207
Sample:	05-01-2020	HQIC	4611.980
	- 12-19-2020		
Covariance Type:	opg		

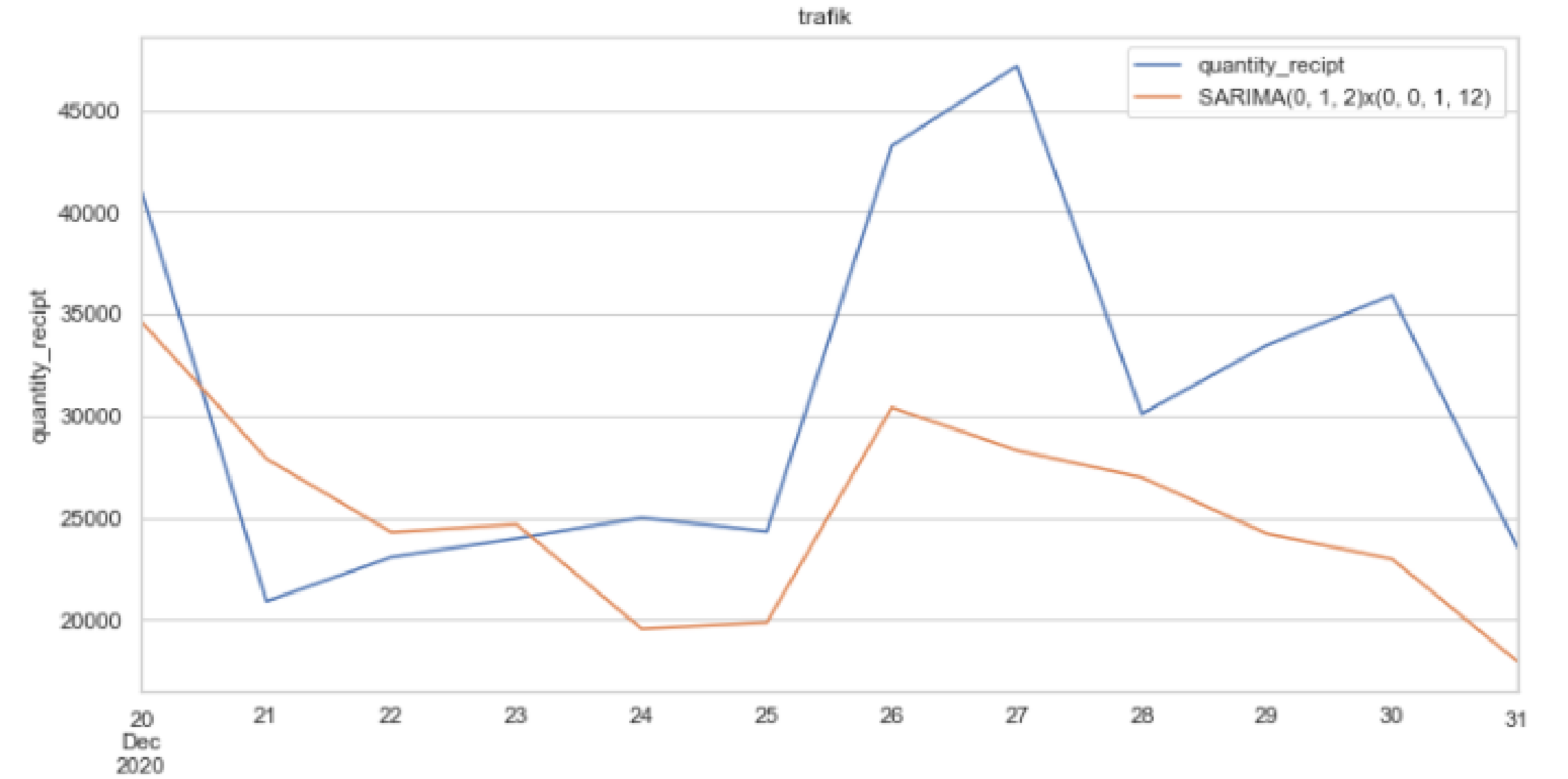
	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.3138	0.099	-3.158	0.002	-0.509	-0.119
ma.L2	-0.4496	0.080	-5.650	0.000	-0.606	-0.294
ma.S.L12	-0.5434	0.059	-9.143	0.000	-0.660	-0.427
sigma2	2.724e+07	7.02e-10	3.88e+16	0.000	2.72e+07	2.72e+07

Ljung-Box (L1) (Q):	0.74	Jarque-Bera (JB):	15.04
Prob(Q):	0.39	Prob(JB):	0.00
Heteroskedasticity (H):	3.71	Skew:	-0.39
Prob(H) (two-sided):	0.00	Kurtosis:	3.97

```
# Сравниваем результаты
for i in range(len(predictions)):
    print(f"predicted={predictions[i]:<11.10
```

```
predicted=34611.83103, expected=41038
predicted=27887.79746, expected=20881
predicted=24273.91392, expected=23060
predicted=24662.96819, expected=23971
predicted=19535.98468, expected=24992
predicted=19843.51329, expected=24314
predicted=30405.12772, expected=43255
predicted=28294.25664, expected=47171
predicted=26969.94826, expected=30098
predicted=24204.01055, expected=33477
predicted=22963.62082, expected=35916
predicted=17931.25068, expected=23552
```

Графики прогнозируемых и известных значений



Оценка качества и точности математических моделей с использованием стандартных метрик (ошибок)

Среднее абсолютное отклонение

$$MAD = \frac{1}{n}\sum_{1}^{n} |Yt - \hat{Yt}|$$

Среднеквадратическая ошибка

$$MSE = \frac{1}{n}\sum_{1}^{n} (Yt - \hat{Yt})^2$$

Средняя абсолютная ошибка в процентах

Наиболее важная. Именно на нее обращают внимание в первую очередь, когда сравнивают точность методов между собой. Уменьшение этой ошибки приводит к уменьшению всех остальных ошибок.

$$MAPE = \frac{1}{n}\sum_{1}^{n} \frac{|Yt - \hat{Yt}|}{Yt}$$

Средняя процентная ошибка

$$MPE = \frac{1}{n}\sum_{1}^{n} \frac{(Yt - \hat{Yt})}{Yt}$$

```
#Оцениваем качество модели с помощью MSE и RMSE
from sklearn.metrics import mean_squared_error

error = mean_squared_error(test['quantity_recipt'], predictions)
print(f'SARIMA(0, 1, 2)x(0, 0, [1], 12) MSE Error: {error:11.10}')
```

```
SARIMA(0, 1, 2)x(0, 0, [1], 12) MSE Error: 79889772.29
```

```
from statsmodels.tools.eval_measures import rmse
# rmse = root mse
error = rmse(test['quantity_recipt'], predictions)
print(f'SARIMA(0, 1, 2)x(0, 0, 1, 12) RMSE Error: {error:11.10}')
```

```
SARIMA(0, 1, 2)x(0, 0, 1, 12) RMSE Error:  8938.10787
```

```
# MAPE
mape = np.mean(np.abs(predictions - test['quantity_recipt'])/np.abs(test['quantity_recipt']))
mape
```

```
0.22110745657231254
```

Обучаем модель на полных данных и прогнозируем будущее

График прогнозируемых данных на 2021 г

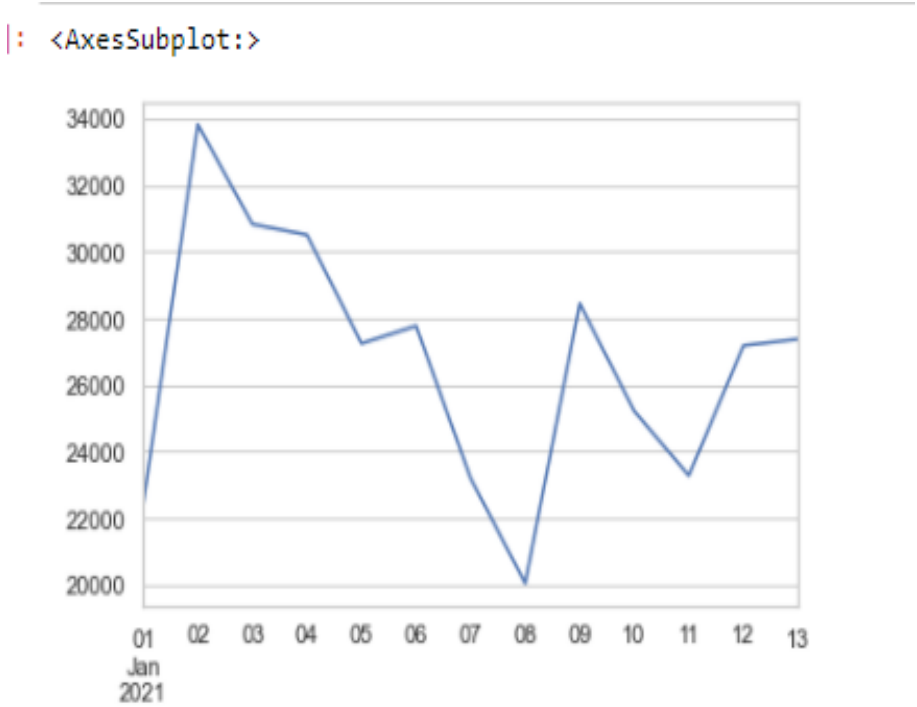
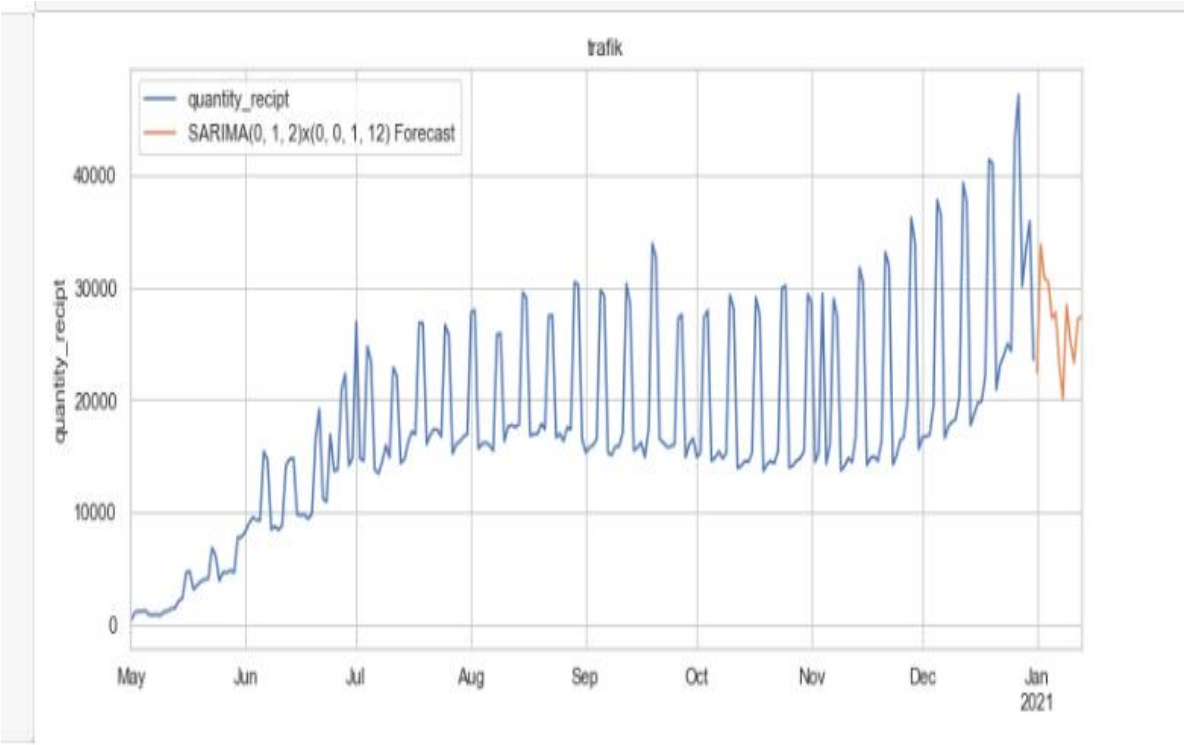


График предсказаний и график исходных данных



Вывод

- Провели тест на стационарность получили результат, что ряд стационарен, но есть зависимость от тренда
- Построили график функции автокорреляции которая также указывает на нестационарность и необходимость дальнейших преобразований данных исходного ряда
- Запустили autoarima на полном наборе данных для подбора параметров модели и построили
- диагностические графики из графиков видно, что показание графиков диагностических моделей отличается от требуемых.
- Построили sarima обучили ее на тренировочных данных и построили прогноз на полных данных на год вперед
- Получили ошибки
- MSE Error: 79889772.29
- RMSE Error: 8938.10787
- Mape 0.22110745657231254
- Исходя из Mape получаем что % точности при прогнозировании на год составляет 78%