ТИТОВА Ирина Владимировна

Итоговая работа по курсу «Аналитик больших данных»
2 поток, 2022-2023г.г.

КЕЙС 40

https://drive.google.com/drive/folders/19AF-5dDhtH57mqSqDRHL0P28gLSvzKJ9?usp=share_link

1. Подготовка данных для загрузки в KNIME Analytics Platform

Для выполнения итоговой работы был выбран кейс №40. В нем содержатся данные о продажах моделей аэропланов и геликоптеров в США за 2017 – 2019 г.г.

В исходном файле excel 4 листа:

Лист Product (9 столбцов, 20 строк) – перечень и описание товаров.

ProductID – идентификатор продукта;

ProductSKU – артикул;

ProductName – наименование продукта;

ProductCategory – категория продукта;

ItemGroup – группа товаров;

KitType – тип комплекта;

Channels – каналы;

Demografic – уровень сложности;

RetailPrice – розничная цена.

Лист Region (2 столбца, 6 строк) – перечень регионов продаж.

RegionID – идентификатор региона;

RegionName – название региона.

Лист Sales (9 столбцов, 377741 строка) – данные о продажах.

OrderNumber – номер заказа;

OrderDate – дата заказа;

ShipDate – дата отправки заказа;

CustomerStateID – идентификатор штата покупателя;

ProductID – идентификатор продукта;

Quantity – количество товара в заказе;

UnitPrice – цена за единицу товара;

DiscountAmount – сумма скидки по промокоду;

PromotionCode – промокод.

Лист State (4 столбца, 51 строка) – перечень штатов, где проживают покупатели.

StateID – идентификатор штата;

StateCode – код штата;

StateName – название штата;

RegionID – идентификатор региона.

Цель работы - исследование и прогноз сумм продаж.

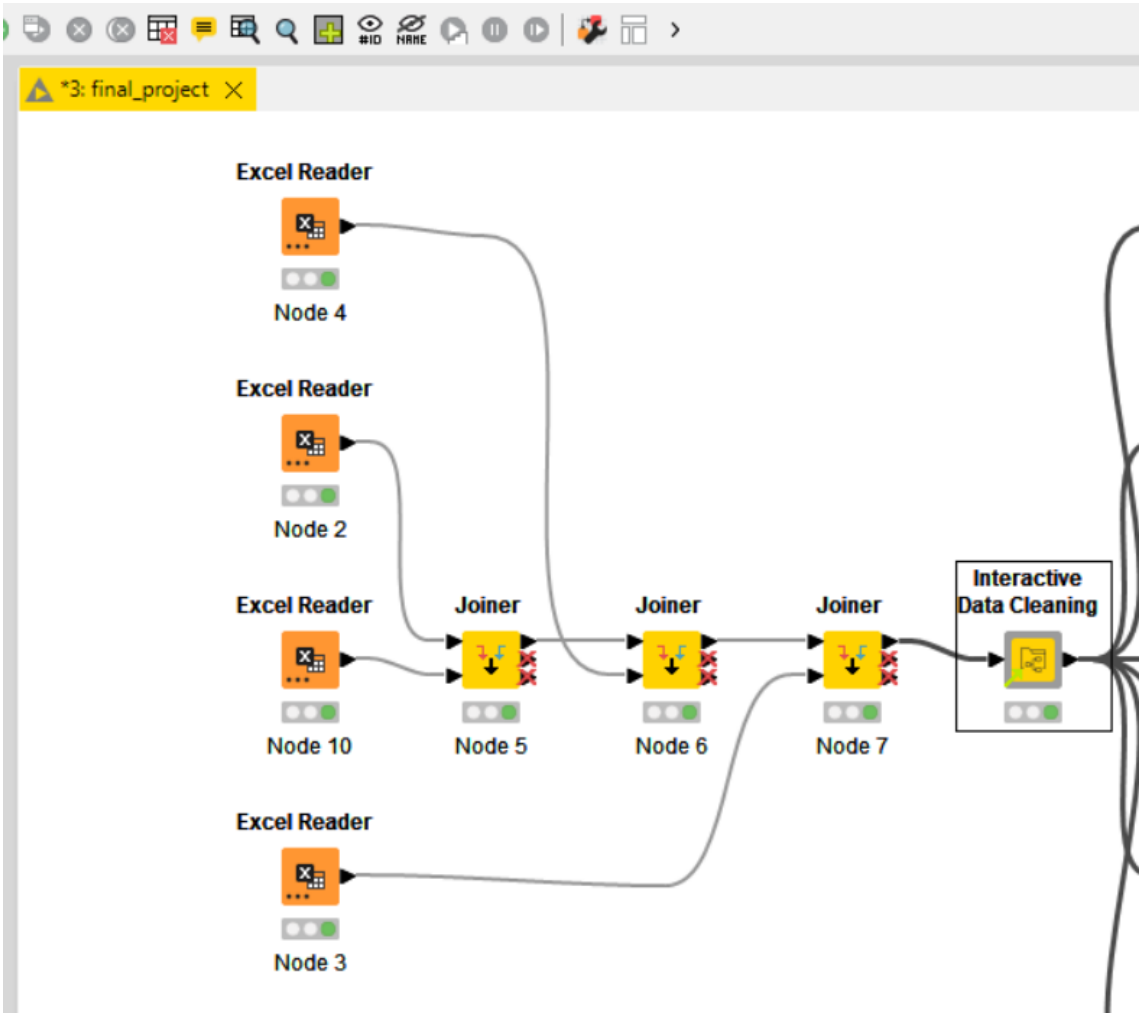
Для получения данных о суммах продаж в таблицу Sales добавлен столбец SaleAmount, содержащий сумму продаж по каждому заказу с учетом скидки.

2. KNIME Analytics Platform

2.1 Подготовка данных

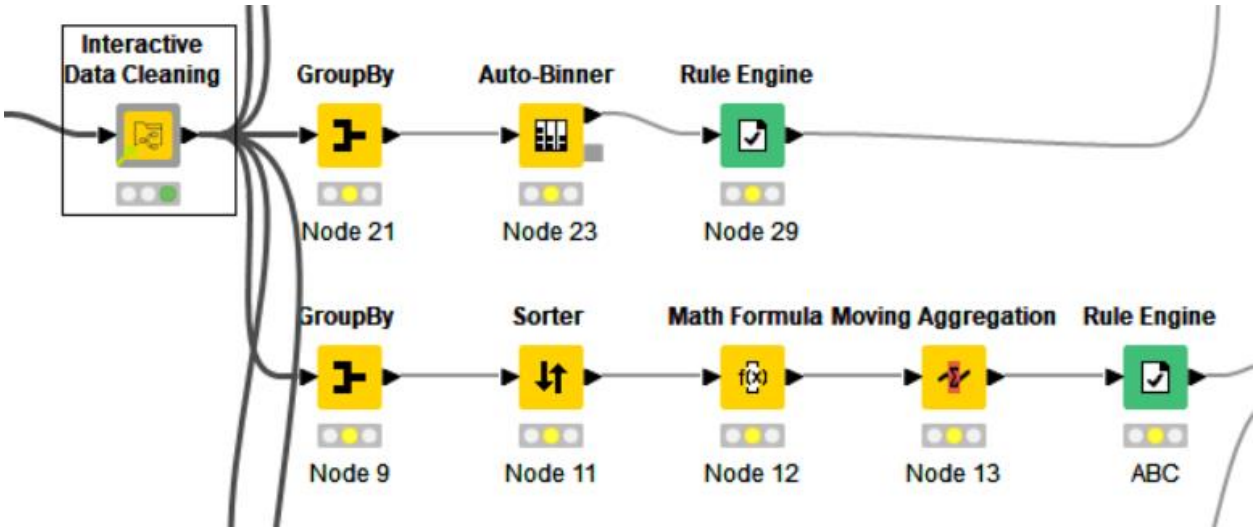
Проведена загрузка данных (4 таблицы). Таблицы объединены, в итоговой таблице получены 21 столбец, 377741 строка.

Посредством Interactive Data Cleaning произведена очистка данных: заполнены пропущенные значения в столбце PromotionCode и удален столбец ProductSKU как неинформативный.



2.2 ABC-XYZ, RFM-анализ

АВС-анализ



По результатам АВС-анализа товары классифицированы по степени важности.

16 товаров группы А (наиболее ценные), 3 товара группы В (промежуточные) и 1 товар группы С (наименее ценный).

File Edit Hilite Navigation View

Table "default" - Rows: 20 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	S ProductName	D Sum(Sa...	D Доля в...	D Sum(Д...	S ABC
Row15	Tailspin Heli - Max Pro Flight - 6ch	60,153,328.6	49.395	49.395	C
Row4	6CCP-A Helicopter	13,086,537.1	10.746	60.141	B
Row9	Piper Cub 4 Channel	10,873,882.4	8.929	69.07	B
Row6	P47 5 Channel	8,771,375.8	7.203	76.273	B
Row13	Tailspin Aviator Mk2-15	5,878,391.85	4.827	81.1	A
Row14	Tailspin Heli - Co-Ax Pro Mk I - 4ch	3,544,559.6	2.911	84.011	A
Row17	Tailspin Warbird BM32	3,468,572.8	2.848	86.859	A
Row12	Tailspin Aviator Mk2-12	3,303,140.7	2.712	89.571	A
Row7	P51	2,426,715.3	1.993	91.564	A
Row10	SkyTrainer	2,299,060	1.888	93.452	A
Row0	3CAX-B Helicopter	1,636,809.75	1.344	94.796	A
Row2	4CAX-B Helicopter	1,413,330.55	1.161	95.957	A
Row8	Piper Cub 3 Channel	1,229,610.85	1.01	96.966	A
Row5	P47 4 Channel	1,105,072.6	0.907	97.874	A
Row19	Trainer - Tailspin GL-155	883,205.85	0.725	98.599	A
Row11	Tailspin Aviator Mk2-11	754,207.6	0.619	99.218	A
Row18	Trainer - Tailspin GL-120	520,288.25	0.427	99.645	A
Row1	3CFP-I Helicopter	275,842.35	0.227	99.872	A
Row16	Tailspin Heli - Pro Mk III - 5ch	78,671.7	0.065	99.937	A
Row3	4CFP-I Helicopter	77,224	0.063	100	A

XYZ-анализ

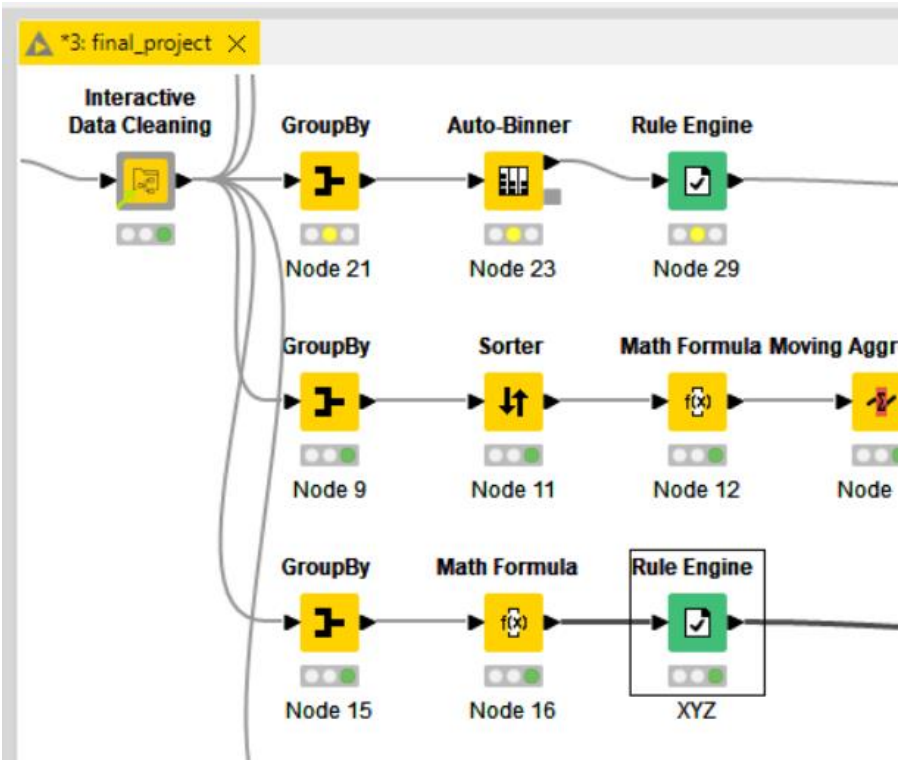


Table "default" - Rows: 20 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	ProductName	Mean(...	Standa...	variation	XYZ
Row0	3CAX-B Helicopter	1	0	0	X
Row1	3CFP-I Helicopter	1	0	0	X
Row2	4CAX-B Helicopter	1	0	0	X
Row3	4CFP-I Helicopter	1	0	0	X
Row4	6CCP-A Helicopter	1	0	0	X
Row5	P47 4 Channel	1	0	0	X
Row6	P47 5 Channel	1	0	0	X
Row7	P51	1	0	0	X
Row8	Piper Cub 3 Cha...	1	0	0	X
Row9	Piper Cub 4 Cha...	1	0	0	X
Row10	SkyTrainer	1	0	0	X
Row11	Tailspin Aviator ...	1	0	0	X
Row12	Tailspin Aviator ...	1	0	0	X
Row13	Tailspin Aviator ...	1	0	0	X
Row14	Tailspin Heli - Co...	1	0	0	X
Row15	Tailspin Heli - Ma...	1	0	0	X
Row16	Tailspin Heli - Pr...	1	0	0	X
Row17	Tailspin Warbird ...	1	0	0	X
Row18	Trainer - Tailspin...	1	0	0	X
Row19	Trainer - Tailspin...	1	0	0	X

Коэффициент вариации по всем товарам равен 0. В каждом из заказов присутствует от 1 до 3 товаров, причем подавляющее большинство заказов – с одним товаром. Профиль магазина – специфический, товары не относятся к товарам широкого потребления, поэтому, имея вышеуказанную картину, вполне можно говорить о стабильности спроса.

RFM-анализ

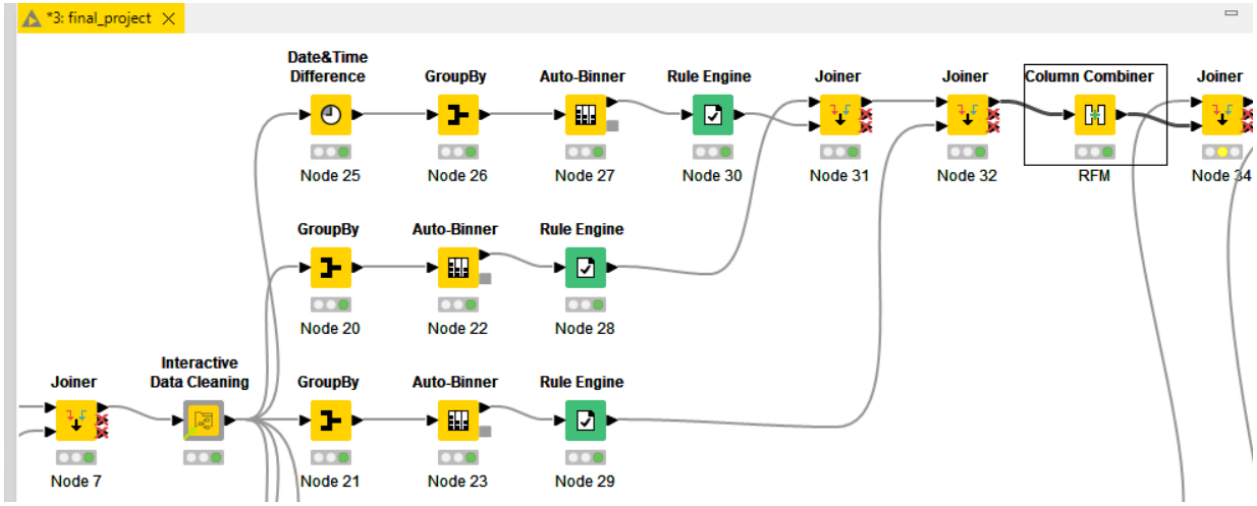
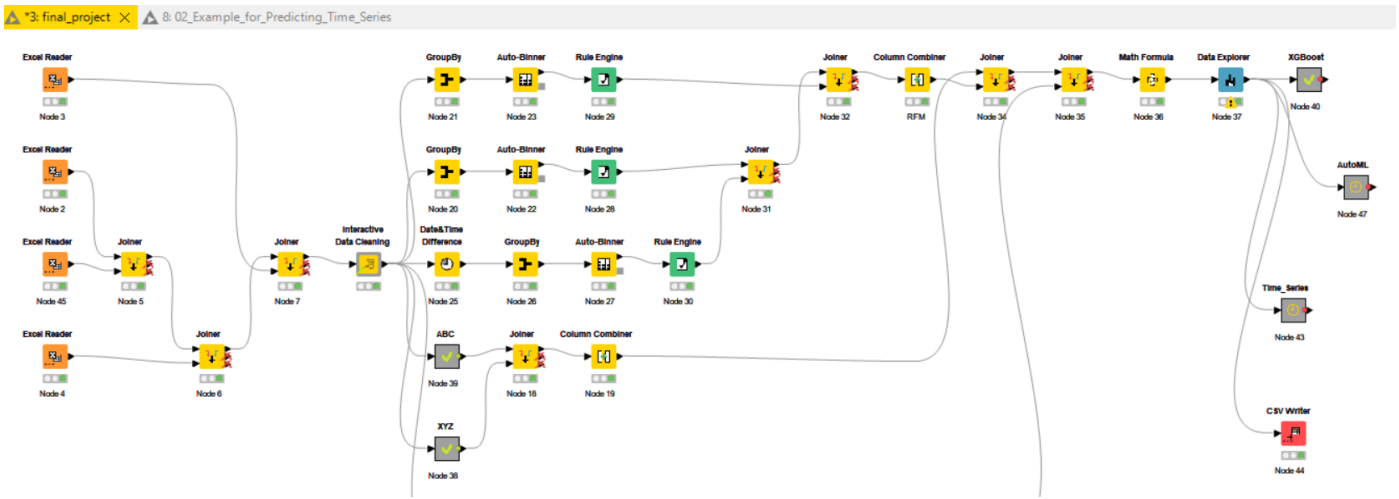


Table "default" - Rows: 20 | Spec - Columns: 3 | Properties | Flow Variables

Row ID	S ProductName	L Max*(d...	S RFM
Row0_Row0_...	3CAX-B Helicopter	2180	4,4,3
Row1_Row1_...	3CFP-I Helicopter	1874	1,5,1
Row2_Row2_...	4CAX-B Helicopter	2208	3,2,3
Row3_Row3_...	4CFP-I Helicopter	1784	1,5,1
Row4_Row4_...	6CCP-A Helicopter	1601	5,5,5
Row5_Row5_...	P47 4 Channel	2208	3,2,2
Row6_Row6_...	P47 5 Channel	2208	4,2,5
Row7_Row7_...	P51	2180	2,4,3
Row8_Row8_...	Piper Cub 3 Cha...	2239	4,1,2
Row9_Row9_...	Piper Cub 4 Cha...	2149	5,4,5
Row10_Row1...	SkyTrainer	2207	3,3,3
Row11_Row1...	Tailspin Aviator ...	2208	2,2,2
Row12_Row1...	Tailspin Aviator ...	2208	4,2,4
Row13_Row1...	Tailspin Aviator ...	2208	5,2,4
Row14_Row1...	Tailspin Heli - Co...	2229	3,1,4
Row15_Row1...	Tailspin Heli - Ma...	2149	5,4,5
Row16_Row1...	Tailspin Heli - Pr...	1996	1,5,1
Row17_Row1...	Tailspin Warbird ...	2208	2,2,4
Row18_Row1...	Trainer - Tailspin...	2180	2,4,1
Row19_Row1...	Trainer - Tailspin...	2208	1,2,2

В рамках XYZ-анализа проведена сегментация товаров по параметрам «давность – частота – деньги».

Таким образом, исходный кейс был обогащен расчетными данными и результатами ABC-XYZ-RFM-анализов.



Итоговый датасет (dataset.csv)

https://drive.google.com/file/d/1GoP5_ioEW0gG_zWIQ5JJjkXGuR1Erxyb/view?usp=share_link

Filtered Table - 3:37 - Data Explorer

File Edit Hilite Navigation View

Table "default" - Rows: 377741 Spec - Columns: 25 Properties Flow Variables

Row ID	ABC_XYZ	Max*(d...	RFM	ProductID	ProductName	ProductCate...	ItemGr...	KitType	Chann...	Demog...	RetailPrice	OrderN...	OrderD...	ShipDat
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017434	2017-01-01	2017-01-09
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017445	2017-01-02	2017-01-04
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017456	2017-01-06	2017-01-09
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017467	2017-01-08	2017-01-11
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017478	2017-01-08	2017-01-10
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017489	2017-01-09	2017-01-12
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017490	2017-01-09	2017-01-11
Row14...	AX	2229	3,1,4	13	Tailspin Heli - Co-A...	Co-Axial	Helicopter	KIT	4	Intermediate	389.95	TT00017501	2017-01-11	2017-01-14
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017512	2017-01-11	2017-01-13
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017523	2017-01-11	2017-01-13
Row8_...	AX	2239	4,1,2	3	Piper Cub 3 Channel	Trainer	Airplane	RTF	3	Beginner	84.65	TT00017534	2017-01-12	2017-01-15

CustomerSt...	Quantity	UnitPrice	Discoun...	Promotion...	SaleAm...	StateC...	StateN...	Regio...	Region...	Churn
14	1	69.95	0	Missing	69.95	IN	Indiana	1	Midwest	1
13	1	69.95	0	Missing	69.95	IL	Illinois	1	Midwest	1
35	1	69.95	0	Missing	69.95	OH	Ohio	1	Midwest	1
3	1	69.95	0	Missing	69.95	AZ	Arizona	5	Southern	1
5	1	69.95	0	Missing	69.95	CA	California	6	Southwest	1
40	1	69.95	0	SALE2016-01	62.95	SC	South Carolina	5	Southern	1
33	1	69.95	0	Missing	69.95	NC	North Carolina	5	Southern	1
23	1	308.95	0	Missing	308.95	MN	Minnesota	1	Midwest	1
14	1	69.95	0	Missing	69.95	IN	Indiana	1	Midwest	1
9	1	69.95	0	Missing	69.95	FL	Florida	5	Southern	1
5	1	69.95	0	Missing	69.95	CA	California	6	Southwest	1

2.3 Обучение и оценка моделей

2.3.1. В Knime выполнено обучение и прогнозирование сумм продаж в разрезе наименований товаров с применением модели **XGBoost** (регрессия).

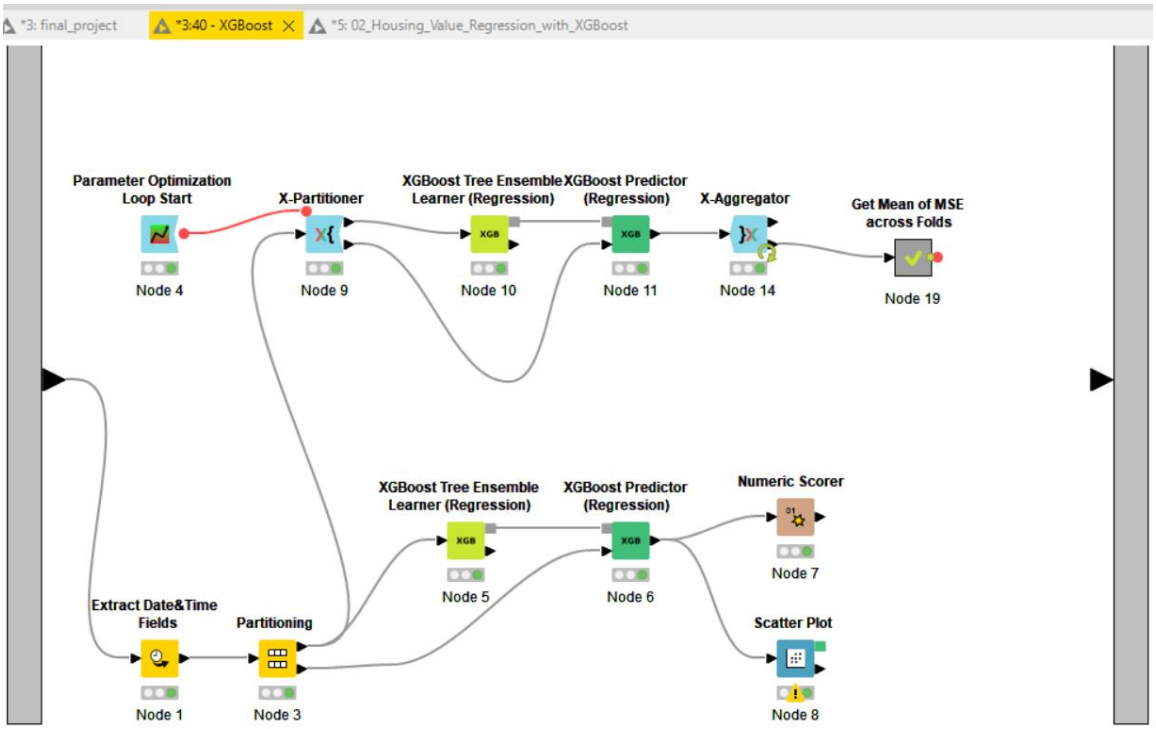
https://drive.google.com/file/d/1hwLwcrgCkiNHnIAQbu_i9bJEsMYuKiUa/view?usp=share_link

Для чистоты исследования из него были исключены столбцы, не имеющие прямого отношения к цели прогноза – сумме продаж.

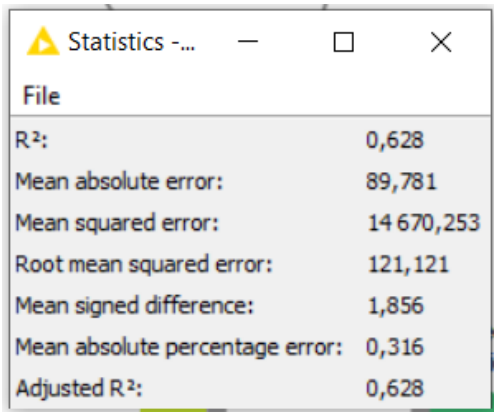
Данные разделены на обучающие и тестовые в пропорции 80% к 20%.

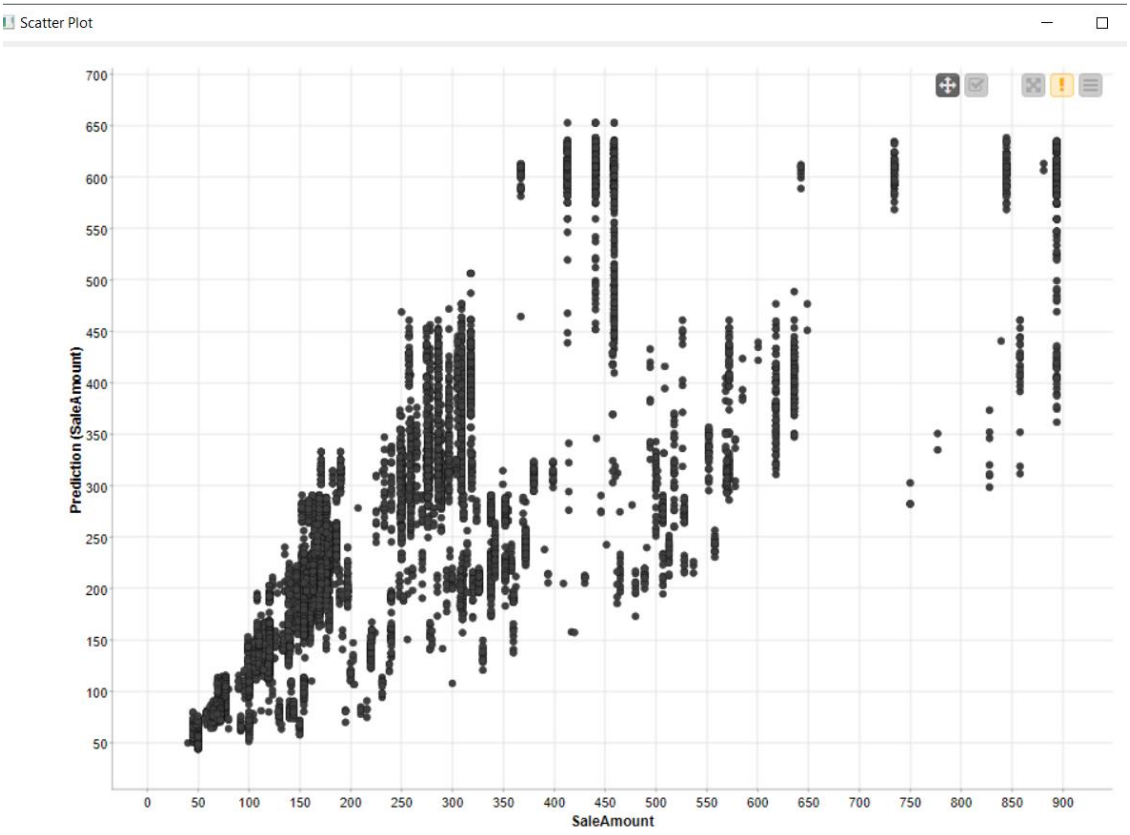
Столбцы, участвующие в исследовании:

Columns: 13
ProductName
ProductCategory
ItemGroup
KitType
Channels
Demographic
RetailPrice
OrderDate
ShipDate
Quantity
UnitPrice
DiscountAmount
SaleAmount



Результат:





Коэффициент детерминации R2 для данной модели равен 0.628, что считается неважным результатом. Остальные метрики также являются слабыми и говорят о невысокой точности модели.

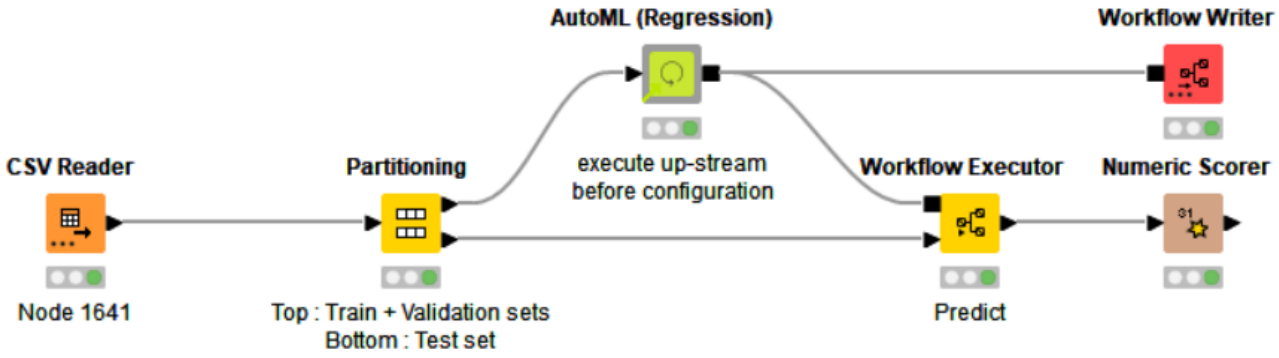
График разброса показывает корреляцию между фактическими и прогнозными значениями, но она имеет достаточно большой разброс, что также говорит о невысокой точности.

Таким образом, данные нуждаются в более внимательном изучении и подготовке, а модель – возможно, в дополнительной настройке параметров.

2.3.2. AutoML (XGBoost Linear Ensemble, регрессия)

Данные разделены на обучающие и тестовые в пропорции 80% к 20%.

https://drive.google.com/file/d/1ztm8Nm4Aydcv1RGXVsmSrzDKVVhGKGIP/view?usp=share_link



Statistics -...

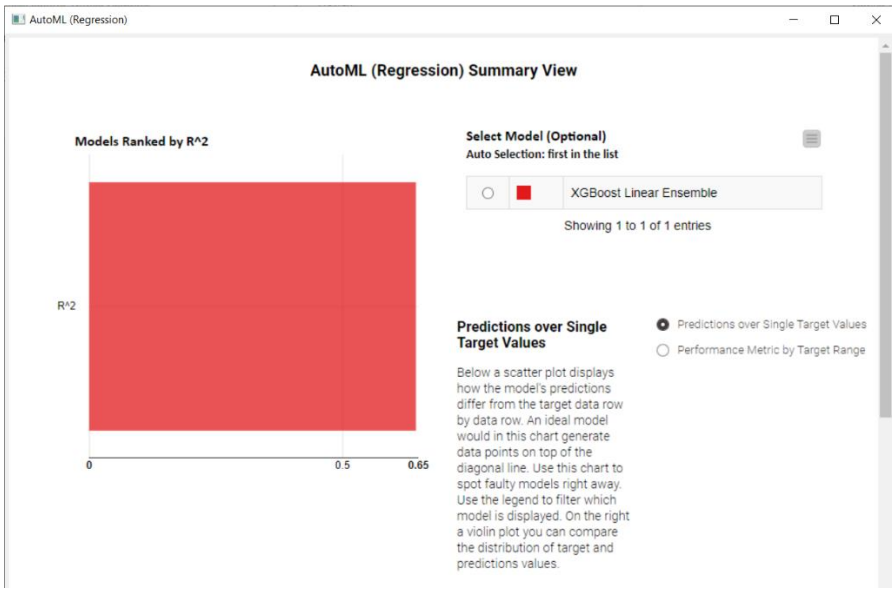
File

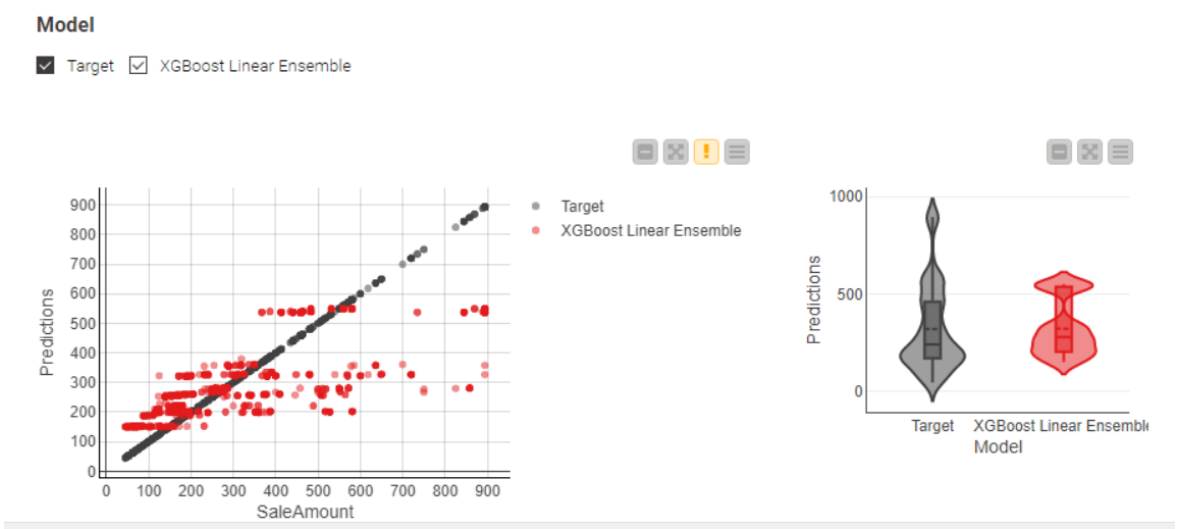
R^2:	0,646
Mean absolute error:	92,103
Mean squared error:	17 035,188
Root mean squared error:	130,519
Mean signed difference:	0,184
Mean absolute percentage error:	0,404
Adjusted R^2:	0,646

Прогнозы по отдельным целевым значениям

Ниже точечный график показывает, как прогнозы модели отличаются от целевых данных строка за строкой. Идеальная модель на этой диаграмме генерировала бы точки данных поверх диагональной линии. Мы видим, что в этой модели корреляция между целевыми и прогнозными значениями меньше, чем в модели XGBoost выше. При этом коэффициент детерминации немного выше (0.646), остальные метрики также слабые.

На графике скрипки можно сравнить распределение целевых значений и прогнозов.

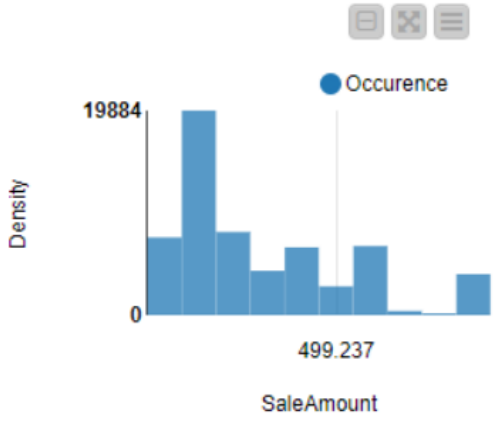




Показатели эффективности по целевым диапазонам

Гистограмма отображает распределение целевых столбцов.

Построение графика R ^ 2 по различным целевым диапазонам недоступно, поскольку R ^ 2 зависит от среднего значения общего целевого распределения.



2.3.3. Временные ряды

https://drive.google.com/file/d/1Iyw95WtpxcQQN_yDBs-OkBeZSIAZUg0j/view?usp=share_link

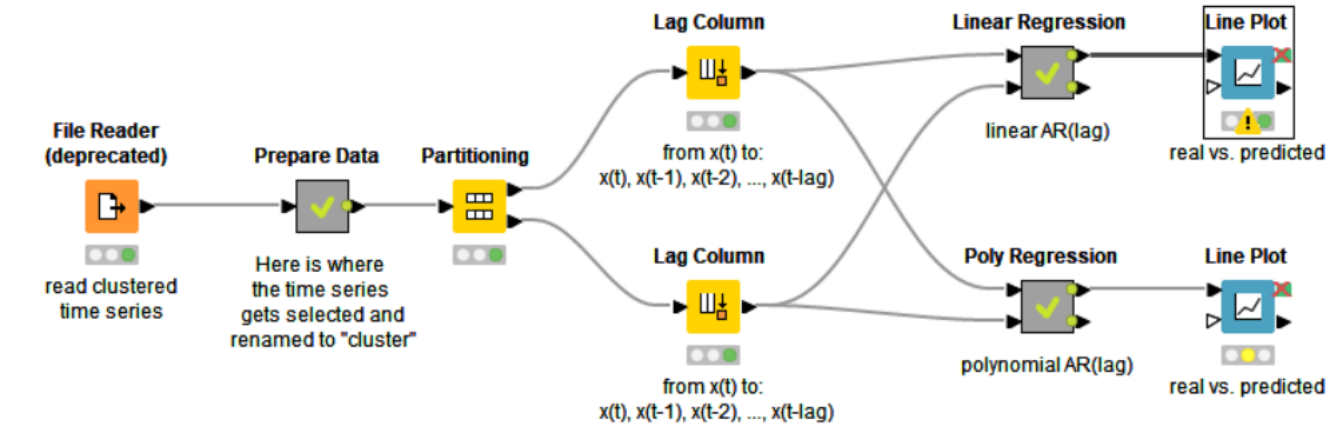
Использовалась простая авторегрессионная модель для прогнозирования временного ряда.

Простота означает только исходные данные: без коррекции на сезонность, допущение стационарности.

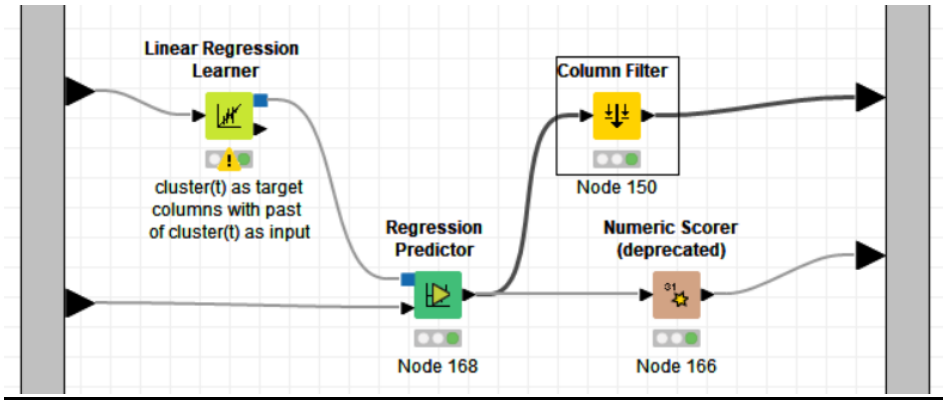
Авто означает использование прошлого того же временного ряда для прогнозирования. Никаких других временных рядов/данных не использовалось.

Модели: линейная и полиномиальная регрессия.

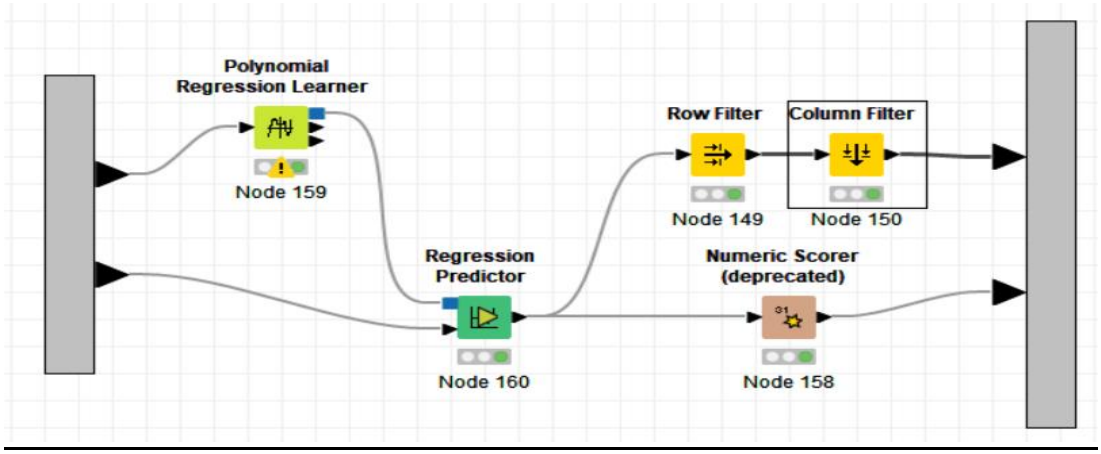
Общая схема



Линейная регрессия:

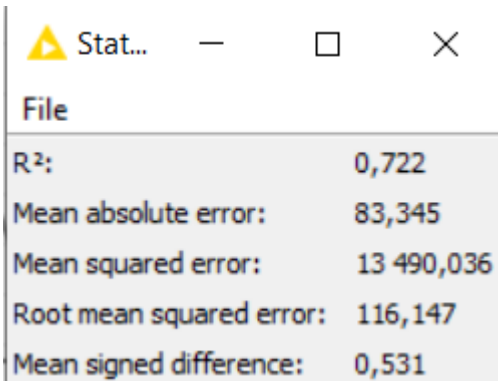


Полиноминальная регрессия

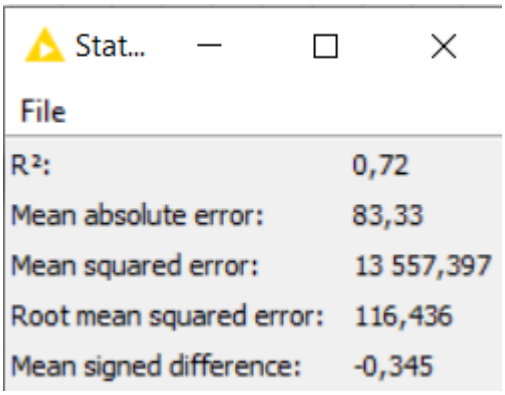


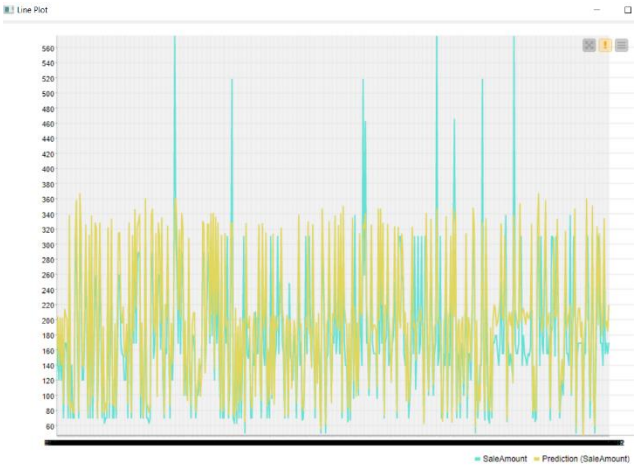
Статистика и графики прогнозов

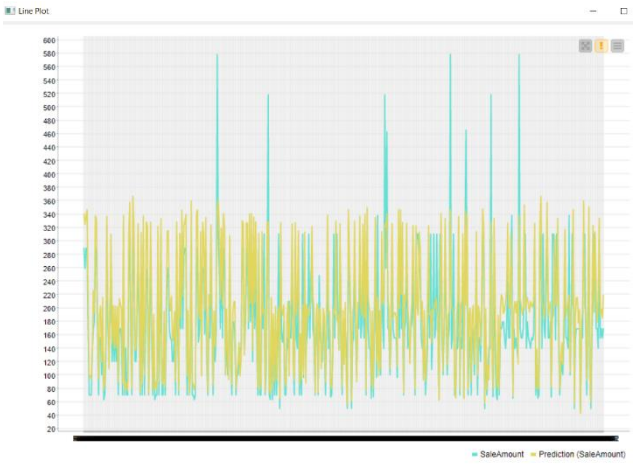
Линейная регрессия



Полиноминальная регрессия







Как видим, показатели линейной и полиноминальной регрессий практически идентичны. Графики также мало отличаются. Фактические показатели (синий цвет) имеют ярко выраженные всплески – сезонные отклонения, прогноз же этих отклонений не отражает. Как сказано выше, модель использовалась простая, без учета сезонности. Я пробовала разные варианты анализа временных рядов, которые предлагает Knime, но они более сложные и главное – долгие в выполнении, поэтому из-за дефицита времени остановилась на простом варианте.

3. EDA, обучение моделей регрессии и временных рядов с использованием библиотек Python

3.1. EDA и модели бустинга

https://colab.research.google.com/drive/1_WBFkZyMlcE0s_MysbFzIHNeLxNZdRzb?usp=sharing

Подготовленный в Knime датасет выгружен и размещен в Google Drive.

В блокноте Colab был проведен исследовательский анализ данных с использованием библиотеки dataprep.

Были удалены неинформативные для целевой переменной SaleAmount столбцы.

Пропущенные значения были в удаленных столбцах.

Категориальные типы данных преобразованы в числовые.

Для обучения были использованы модели CatBoost, LightBoost, XGBoost.

Данные были разделены на обучающие и тестовые сеты.

CatBoost

```
model=CatBoostRegressor(iterations=100, depth=3, learning_rate=0.1, loss_function='RMSE')
model.fit(X_train, y_train, cat_features=cat_feature_type, eval_set=(X_test, y_test), plot=True)
```

```
0:      learn: 189.5118795      test: 188.9079152       best: 188.9079152 (0)   total: 114ms    remaining: 11.3s
1:      learn: 178.1748625      test: 177.6304954       best: 177.6304954 (1)   total: 185ms    remaining: 9.08s
2:      learn: 168.2412898      test: 167.7507229       best: 167.7507229 (2)   total: 219ms    remaining: 7.08s
3:      learn: 159.5846719      test: 159.1374461       best: 159.1374461 (3)   total: 272ms    remaining: 6.53s
4:      learn: 152.3170874      test: 151.9235199       best: 151.9235199 (4)   total: 304ms    remaining: 5.78s
5:      learn: 146.0671989      test: 145.7167898       best: 145.7167898 (5)   total: 341ms    remaining: 5.34s
6:      learn: 140.7895421      test: 140.4766589       best: 140.4766589 (6)   total: 371ms    remaining: 4.93s
7:      learn: 136.3958486      test: 136.1183954       best: 136.1183954 (7)   total: 414ms    remaining: 4.76s
8:      learn: 132.5611721      test: 132.3153733       best: 132.3153733 (8)   total: 447ms    remaining: 4.52s
9:      learn: 129.3707320      test: 129.1524734       best: 129.1524734 (9)   total: 477ms    remaining: 4.29s
10:     learn: 126.7529016      test: 126.5676467       best: 126.5676467 (10)  total: 507ms    remaining: 4.11s
```

```
# print the R-squared train
from sklearn.metrics import r2_score
print("The R-squared value is: {0:0.4f} \n".format(r2_score(y_train,model.predict(X_train))))
```

```
The R-squared value is: 0.6869
```

```
# print the R-squared test
from sklearn.metrics import r2_score
print("The R-squared value is: {0:0.4f} \n".format(r2_score(y_test,model.predict(X_test))))
```

```
The R-squared value is: 0.6846
```

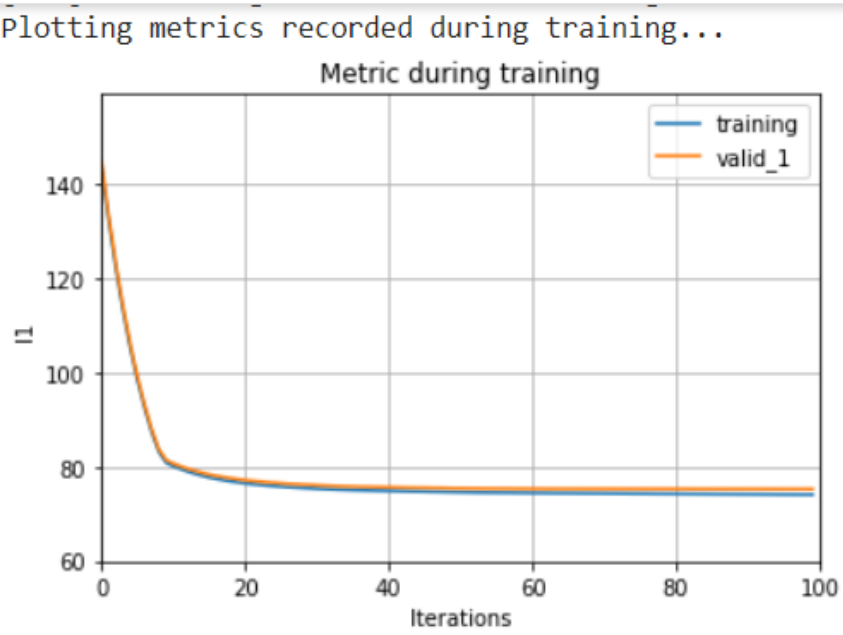
Метрики

```
The R-squared value is: 0.6846

Mean Absolute Error: 75.486360
Root Mean Squared Error: 113.460702
Relative Absolute Error: 0.481497
Relative Squared Error: 0.315368
```

LightBoost

```
# create dataset for lightgbm
lgb_train = lgb.Dataset(X_train_1,y_train_1)
lgb_test = lgb.Dataset(X_validation_1,y_validation_1, reference=lgb_train)

# specify your configurations as a dict
params = {
    'num_leaves': 25,
    'metric': ('l1', 'l2'),
    'verbose': 0
}

evals_result = {}  # to record eval results for plotting

print('Starting training...')
# train
gbm = lgb.train(params,
                lgb_train,
                num_boost_round=100,
                valid_sets=[lgb_train, lgb_test],
                feature_name=['f' + str(i + 1) for i in range(X_train.shape[-1])],
                categorical_feature=[21],
                evals_result=evals_result,
                verbose_eval=10)

print('Plotting metrics recorded during training...')
ax = lgb.plot_metric(evals_result, metric='l1')
plt.show()

print('Plotting feature importances...')
ax = lgb.plot_importance(gbm, max_num_features=10)
plt.show()
```

```
Plotting metrics recorded during training...
```

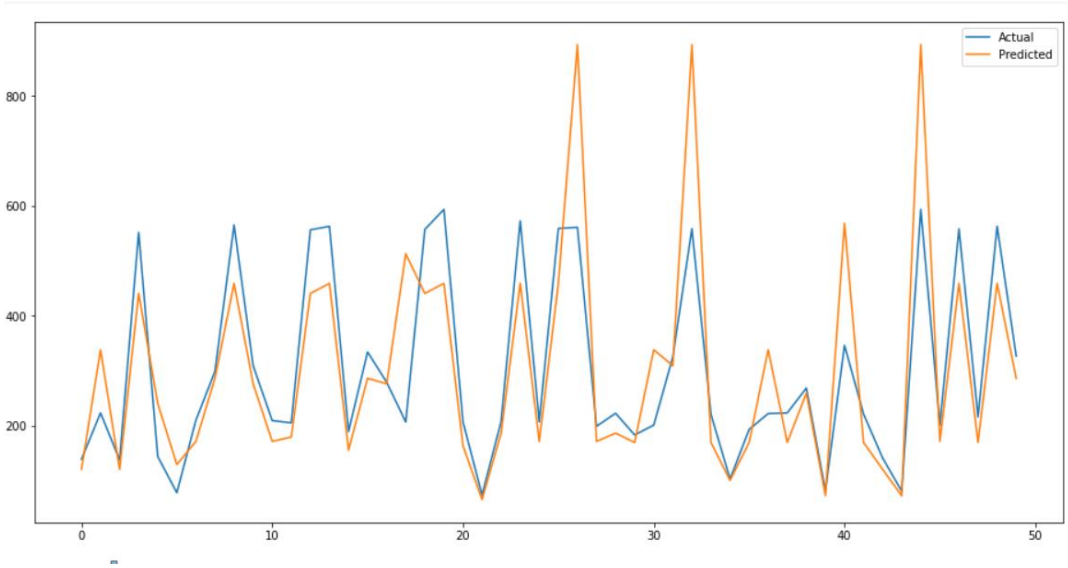


На графике видим, что метрики train и test практически полностью совпадают. Значит, модель одинаково работает и в обучении, и в прогнозе, что хорошо.

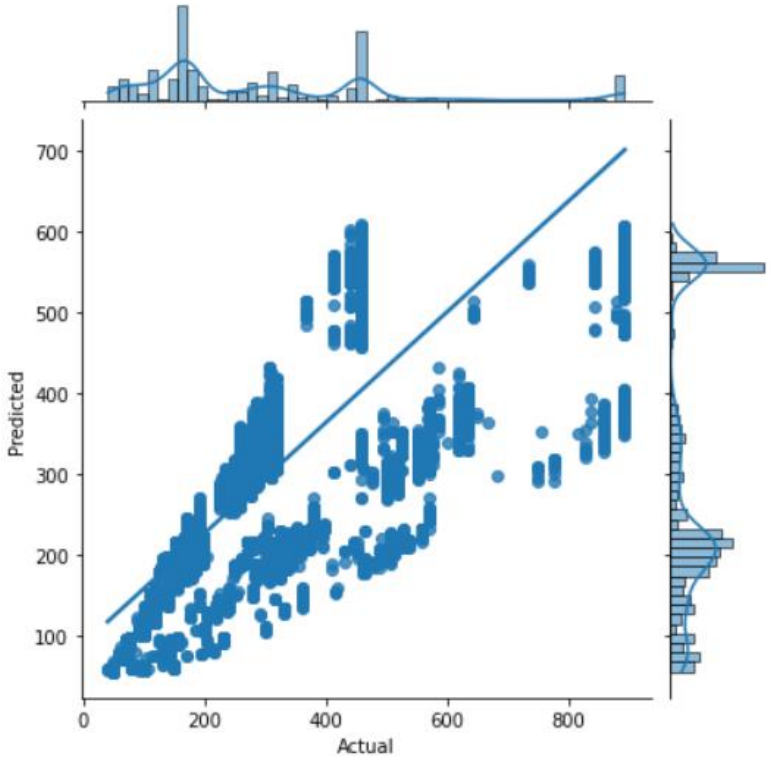
Значения R2 при этом:

Train – 0,6954

Test – 0,6878

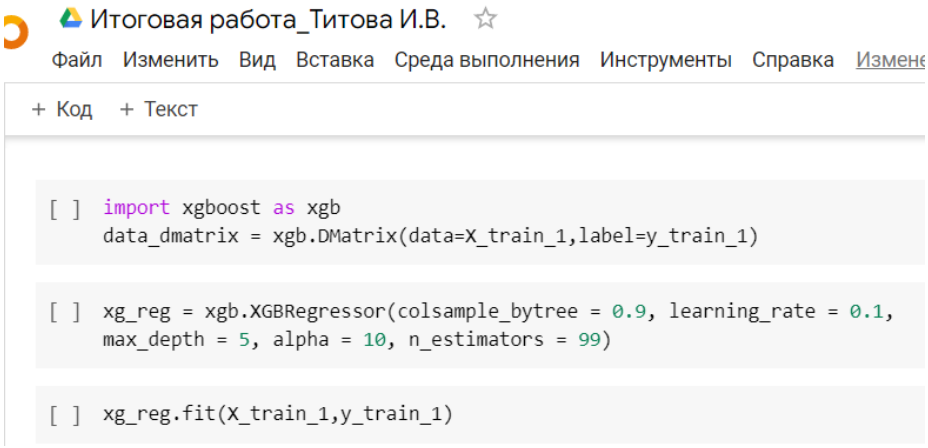


Прогноз неплохо коррелирует с фактическими значениями, но показывает больше выбросов.



Так же, как и в модели XGBoost Knime график разброса здесь показывает корреляцию между фактическими и прогнозными значениями, но также с большим разбросом.

XGBoost



Значения R2:

Train – 0,6945

Test – 0,6878

```
Mean Absolute Error: 75.255884
Root Mean Squared Error: 113.685533
Relative Absolute Error: 0.476569
Relative Squared Error: 0.312191
```

Сравнения метрик различных моделей будет сделано в п.5.

3.2. Временные ряды

https://colab.research.google.com/drive/1J2BwhMAKnzi7CaKVaQGGGwjjLlc0GORS?usp=sharing

Для анализа с помощью модели временных рядов из исходного датасета сформирована локальная выборка с двумя столбцами – временным рядом и целевым признаком.

```
data_time = data[['OrderDate', 'SaleAmount']]
data_time
```

	OrderDate	SaleAmount
0	2017-01-01	69.95
1	2017-01-02	69.95
2	2017-01-06	69.95
3	2017-01-08	69.95
4	2017-01-08	69.95
...	...	...
377736	2019-12-31	63.55
377737	2019-12-31	193.55
377738	2019-12-31	560.25
377739	2019-12-31	579.55

Столбец OrderDate имеет тип «объект», поэтому его потребовалось перевести в формат даты.

AutoTS

Длина прогноза установлена на 90 дней.

```
from autots import AutoTS
model = AutoTS(forecast_length=90, frequency='infer',
               ensemble='simple')
model = model.fit(data_time, date_col='OrderDate', value_col='SaleAmount', id_col=None)
```

```
10/10 [==============================] - 0s 9ms/step - loss: 0.3757
Epoch 23/50
10/10 [==============================] - 0s 9ms/step - loss: 0.3758
Epoch 24/50
```

```
DEBUG:cmdstanpy:idx 0
DEBUG:cmdstanpy:running CmdStan, num_threads: None
DEBUG:cmdstanpy:CmdStan args: ['/usr/local/lib/python3.8/dist-packages/prophet/stan_model/prophet_model.bin', 'random', 'seed=43148', 'data', 'file=/tmp/tmpes03_6b3/w6mwua3n.json', '
16:44:36 - cmdstanpy - INFO - Chain [1] start processing
INFO:cmdstanpy:Chain [1] start processing
16:44:36 - cmdstanpy - INFO - Chain [1] done processing
INFO:cmdstanpy:Chain [1] done processing
SaleAmount Forecast
            SaleAmount
2020-01-01  576.932999
2020-01-02  600.059908
2020-01-03  612.345663
2020-01-04  585.960486
2020-01-05  592.694484
...                ...
2020-03-26  634.215758
2020-03-27  634.841342
2020-03-28  635.089722
2020-03-29  636.181525
2020-03-30  636.755961

[90 rows x 1 columns]
```

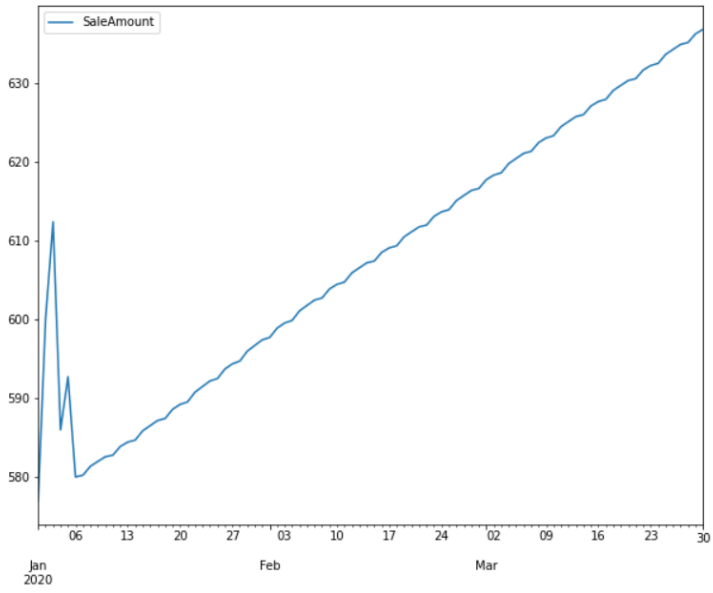
Итоговая работа_временные ряды

Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения сохр

+ Код + Текст

```
forecast.plot(subplots=True, figsize=(10,8))
```

```
array([<AxesSubplot:>], dtype=object)
```



GreyKite

Итоговая работа_временные ряды

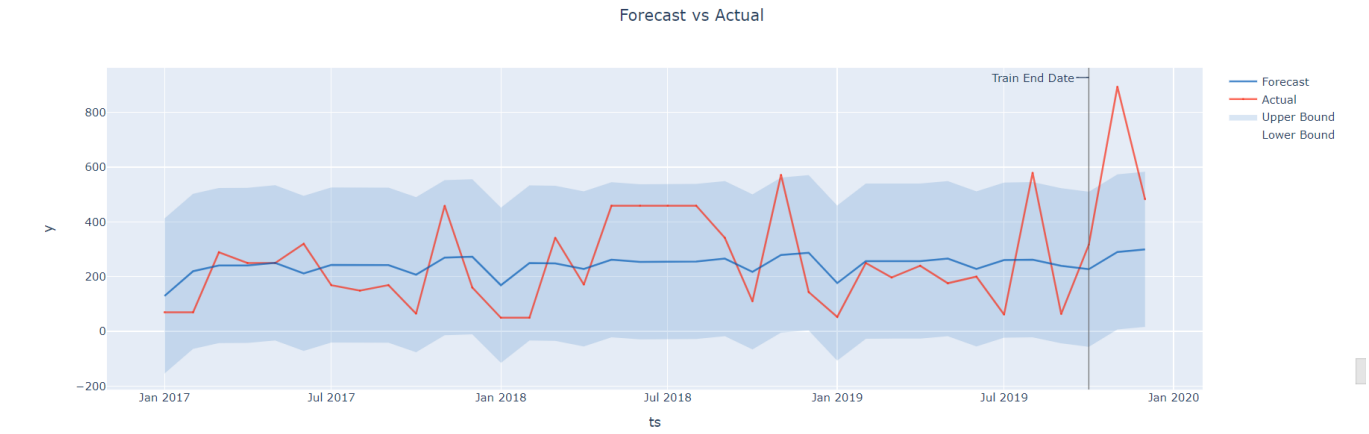
Файл Изменить Вид Вставка Среда выполнения Инструменты Справка Изменения сохранены

+ Код + Текст

```
from greykite.algo.changepoint.adalasso.changepoint_detector import ChangepointDetector
from greykite.algo.forecast.silverkite.constants.silverkite_holiday import SilverkiteHoliday
from greykite.algo.forecast.silverkite.constants.silverkite_seasonality import SilverkiteSeasonalityEnum
from greykite.algo.forecast.silverkite.forecast_simple_silverkite_helper import cols_interact
from greykite.common import constants as cst
from greykite.common.features.timeseries_features import build_time_features_df
from greykite.common.features.timeseries_features import convert_date_to_continuous_time
from greykite.framework.benchmark.data_loader_ts import DataLoaderTS
from greykite.framework.templates.autogen.forecast_config import EvaluationPeriodParam
from greykite.framework.templates.autogen.forecast_config import ForecastConfig
from greykite.framework.templates.autogen.forecast_config import MetadataParam
from greykite.framework.templates.autogen.forecast_config import ModelComponentsParam
from greykite.framework.templates.forecaster import Forecaster
from greykite.framework.templates.model_templates import ModelTemplateEnum
from greykite.framework.utils.result_summary import summarize_grid_search_results
```

```
# Specifies dataset information
metadata = MetadataParam(
    time_col="ts",  # name of the time column
    value_col="y",  # name of the value column
    freq="MS"  # "H" for hourly, "D" for daily, "W" for weekly, etc.
)

forecaster = Forecaster()
result = forecaster.run_forecast_config(
    df=df,
    config=ForecastConfig(
        model_template=ModelTemplateEnum.SILVERKITE.name,
        forecast_horizon = 2,
        coverage=0.95,  # 95% prediction intervals
        metadata_param=metadata
```

```
backtest = result.backtest
fig = backtest.plot()
plotly.io.show(fig)
```



4. Tableau Public

https://public.tableau.com/views/FinalFU/Dashboard?:language=en-US&:display_count=n&:origin=viz_share_link

Файл https://drive.google.com/file/d/1rStgYtrcqiTfsxA3XesXJe1B1qIIthRE/view?usp=share_link

Датасет, подготовленный в Knime, был загружен в дескстопную версию Tableau Public.

Дашборд отражает состояние продаж и прибыли компании.

На дашборде размещены фильтры, позволяющие фильтровать данные по дате, категории товара и штату продаж.

Сами графики являются интерактивными и могут использоваться как фильтры.

Были созданы вычисляемые поля: Profit, Profit per Order, Profit per Product.

В географических признаках выделены города.

Построено и собрано в дашборд 6 графических представлений датасета:

1. Показатели KPI (общая прибыль, прибыль за заказ, прибыль за продукт, количество, сумма продаж).

2. Динамика во времени суммы продаж и общей прибыли (line-chart).

Тенденции общего объема продаж и общей прибыли по неделям с датой заказа. Цветом показаны сведения об объеме продаж и общей прибыли. Данные фильтруются по названию штата, категории продукта, и дате заказа.

График показывает выраженные всплески продаж – в основном, в апреле и в ноябре. Видимо, в апреле продажи увеличиваются из-за приближающегося лета (возможность использовать товары на свежем воздухе), а в ноябре – из-за предновогодних распродаж и покупок подарков к Новому году.

3. Средняя прибыль по категориям товаров (bar-chart).

Среднее значение прибыли по каждой товарной категории. Цвет показывает подробную информацию о категории продукта. Данные фильтруются по названию штата, категории продукта и дате заказа.

Из графика следует, что наибольшие средние продажи – у категории товара collective pitch, а минимальные – у fixed pitch и glider. Соответственно, продавцу надо активнее предлагать эти товары, возможно, увеличить бюджет на их рекламу.

4. Распределение продаж и прибыли в разрезе регионов и категорий товаров (map).

На графике представлена карта США, т.к. датасет содержит информацию о продажах на территории США. Подробная информация приведена для названия региона и штата. Цвет штатов показывает % от общей прибыли. Размер круглых значков, которыми обозначены города, показывает % от общей суммы продаж. Подробная информация указана для названия региона, штата и города. Данные фильтруются по категории товара, названию штата и дате заказа.

Максимальную прибыль с большим отрывом от других штатов приносит Калифорния (13,14%). Минимальная прибыль – в штате Вайоминг (0,22%). В целом прибрежные штаты приносят больше прибыли, чем центральные. Видимо, это связано с климатом и образом жизни.

5. График, отражающий корреляцию продаж и прибыли (scatter plot).

Отражена зависимость прибыли и сумм продаж. Цвет и размер круглых значков показывают информацию о категории товара. Данные фильтруются по названию штата, категории товара и дате заказа.

На этом графике видим интересный инсайт: товар категории collective pitch, приносящий максимальную среднюю прибыль, также имеет и самые убыточные продажи.

6. Количество заказов по категориям товаров (circle).

Цвет значков показывает информацию о категории продукта, размер - суммарное количество заказов. Данные фильтруются по названию штата, дате заказа и категории товара.

Здесь почти та же картина, что и в графике распределения средней прибыли (п.3), кроме товаров Trainer и Warbird: меньшее количество заказов категории Warbird приносит большую прибыль, чем Trainer.

Ссылка на профиль в Табло Паблик:
https://public.tableau.com/app/profile/irina.titova8839

5. Сравнение моделей машинного обучения

Итак, сравним по метрикам построенные в результате исследования модели.

Основной метрикой в исследованиях был коэффициент детерминации (R-квадрат).

Коэффициент детерминации для модели с константой принимает значения от 0 до 1. Чем ближе значение коэффициента к 1, тем сильнее зависимость. При оценке регрессионных моделей это интерпретируется как соответствие модели данным. Для приемлемых моделей предполагается, что коэффициент детерминации должен быть хотя бы не меньше 50 % (в этом случае коэффициент множественной корреляции превышает по модулю 70 %). Модели с коэффициентом детерминации выше 80 % можно признать достаточно хорошими (коэффициент корреляции превышает 90 %). Значение коэффициента детерминации 1 означает функциональную зависимость между переменными.

	KNIME				Библиотеки ML			
	XGBoost	**AutoML (XGBoost Linear Ensemble)**	**Временные ряды (линейная регрессия)**	**Временные ряды (полиноми-нальная регрессия)**	**CatBoost**	**LightBoost**	**XGBoost**	**Врем. ряды (AutoTS, GreyKite)**
R-квадрат	0.628	0.646	0,722	0,72	0.6846	0,6954	0,6945	-
MAE	89.781	92.103	83,345	83,33	75.48636	-	75,255884	-
MSE	14670.253	17035.188	13490,036	13557,397	-	-		-
RMSE	121.121	130.519	116,147	16,436	113,460702	-	113,685533	-
MSD	1.856	0.184	0,531	-0,345	-	-		-
MAPE	0.316	0.646	-	-	-	-		-

R-квадрат. У всех моделей эта метрика больше 50%, что неплохо, но меньше 80%, что не позволяет считать наши модели достаточно хорошими. Лучший показатель R2 – у моделей временных рядов Knime, но, к сожалению, в результатах исследования невозможно определить, какая именно из моделей временных рядов дала лучший результат. Будем считать, что лучшей по показателю R-квадрат являются модели **LightBoost и XGBoost библиотек ML**.

MAE. Среднеквадратичная ошибка подходит для сравнения двух моделей или для контроля качества во время обучения, но не позволяет сделать выводов о том, насколько хорошо данная модель решает задачу. Например, MAE = 10 является очень плохим показателем, если целевая переменная принимает значения от 0 до 1, и очень хорошим, если целевая переменная лежит в интервале (10000, 100000). Среднее значение нашего целевого показателя SaleAmount равно **322,38**. Соответственно, MAE всех моделей в таблице является плохим показателем. Лучшие из худших – **модели CatBoost и XGBoost библиотек ML**.

MSE применяется в ситуациях, когда нам надо подчеркнуть большие ошибки и выбрать модель, которая дает меньше больших ошибок прогноза. Грубые ошибки становятся заметнее за счет того, что ошибку прогноза мы возводим в квадрат. И модель, которая дает нам меньшее значение среднеквадратической ошибки, считается лучшей, т.к. у нее меньше грубых ошибок. Меньшие значения MSE у временных рядов Knime, но опять же, неизвестно, какие именно модели дали результаты, поэтому выберем наименьшее значение у других моделей. Здесь лучшая **XGBoost Knime**.

RMSE – получается из ошибки MSE путем извлечения корня. Если не брать временные ряды Knime, лучшими моделями здесь будут **CatBoost и XGBoost библиотек ML**.

По ошибкам **MSD и MAPE** есть данные не у всех моделей, поэтому ограничимся перечисленными четырьмя.
Как видно из написанного, чаще всего признавалась лучшей (в 3х случаях из 4х возможных) модель XGBoost библиотек ML.

Вывод: лучшей моделью по результатам описанного здесь исследования является модель XGBoost библиотек ML.