Альбрандт М.Д. Итоговое задание по курсу Анализ больших данных (ЦП 2021)

КЕЙС 10.

1. Описание кейса.

Для данной работы взят кейс №10 из предложенного списка наборов данных.

Выборка данных хранит информацию о продажах радиоуправляемых моделей воздухоплавательной техники за три года в США.

Данные хранятся в excel-файле на 4 листах:

- Лист 1 – Product – содержит 9 столбцов и 20 строк, хранит перечень и описание продукции, содержит следующие данные:

- ✓ Product ID - идентификатор продукта;
- ✓ Product SKU - артикул товара;
- ✓ Product Name - название продукта;
- ✓ Product Category- категория продукта;
- ✓ Item Group - группа товаров;
- ✓ Kit Type- тип комплекта;
- ✓ Channels – количество каналов:
- ✓ Demographic – уровень сложности;
- ✓ Retail Price – актуальная розничная цена

- Лист 2 – Region - содержит 2 столбца и 6 строк, хранит перечень регионов. Содержит следующие данные:

- ✓ Region ID - идентификатор региона;
- ✓ Region Name -наименование региона.

- Лист 3 – Sales - содержит 9 столбцов и 377741 строк, хранит данные о продажах за несколько лет. Содержит следующие данные:

- ✓ Order Number - номер заказа;
- ✓ Order Date - дата заказа;
- ✓ Ship Date - дата отправки;
- ✓ Customer State ID – идентификатор штата отправки;
- ✓ Product ID - идентификатор продукта;
- ✓ Quantity – количество;
- ✓ Unit Price - цена за единицу продукции;
- ✓ Discount Amount – сумма скидки по промокоду;
- ✓ Promotion Code – промокод.

- Лист 4 – State - содержит 4 столбца и 51 строк, хранит перечень штатов. Содержит следующие данные:

- ✓ State ID – идентификатор штата;
- ✓ State Code - код штата;
- ✓ State Name - наименование штата;
- ✓ Region ID - идентификатор региона.

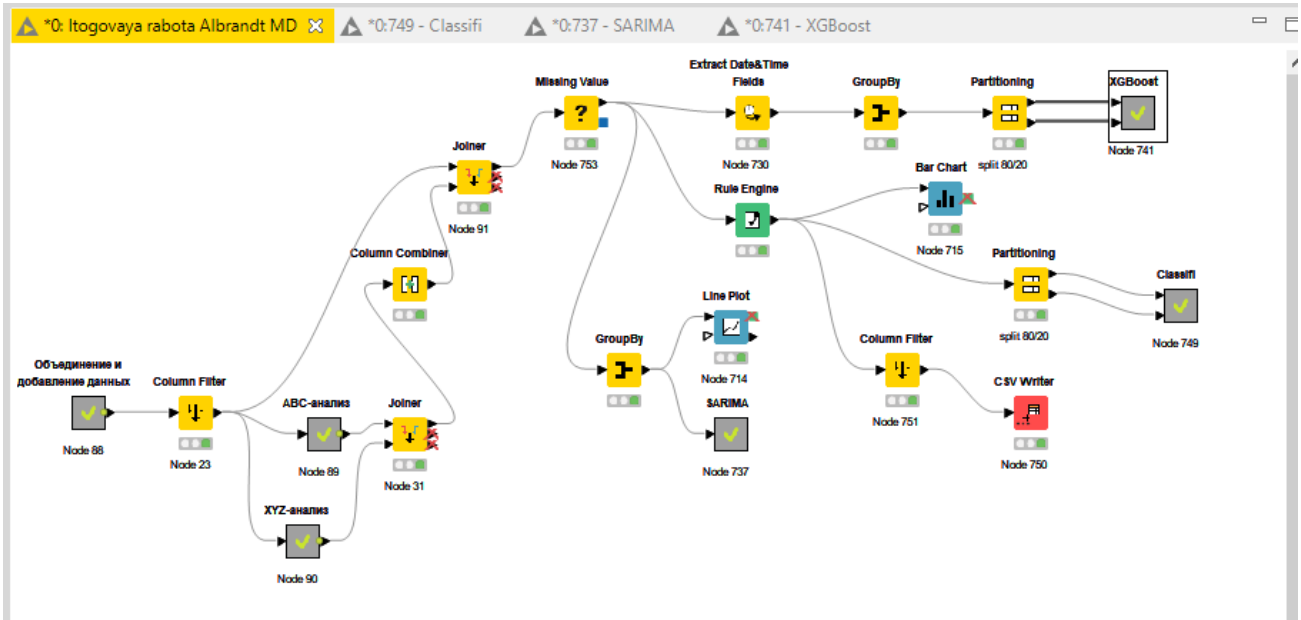
2. Первичная загрузка, обработка и анализ данных с использованием системы Knime Analytics Platform.

Ссылки:

https://drive.google.com/file/d/1rOxUane8qTTeY1SZXqLKBS9aXDjyTkwT/view?usp=sharing - Workflows Knime

С помощью системы Knime Analytics Platform были проведены работы по объединению, первичной обработке и первичному анализу данных.

Кейс был обогащен расчетными данными, также результатами ABC-XYZ-анализа, проведена классификация строк по срокам исполнения, проведено обучение нескольких моделей машинного обучения. Сформирован CSV-файл для загрузки и обработки в *Colab и* BI-платформах.



Рассмотрим детальнее выполненные работы в рамках данного пункта.

1. Первым шагом была произведена загрузка данных.

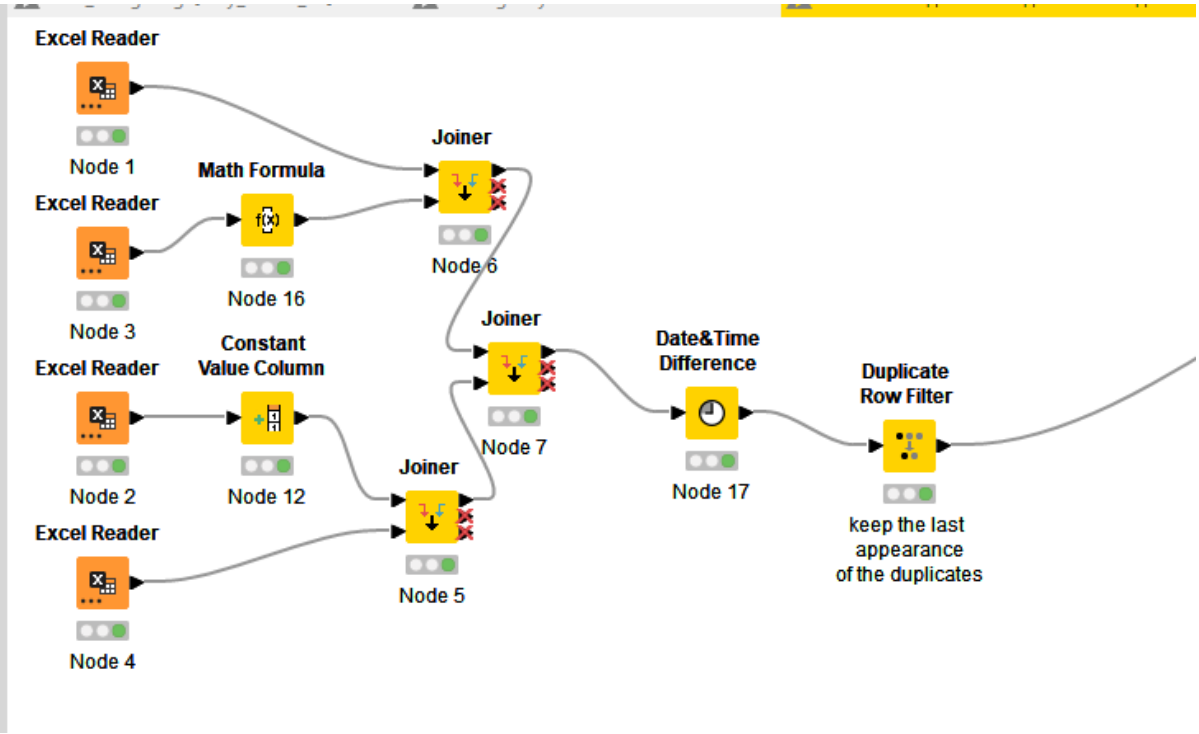
В таблицу Region был добавлен столбец Country, хранящий название страны. Данный столбец добавлен для дальнейшего корректного выстраивание иерархии для геолокации.

В таблицу Sales был добавлен столбец SaleAmount, хранящий информацию о сумме продажи по каждому заказу с учетом скидки.

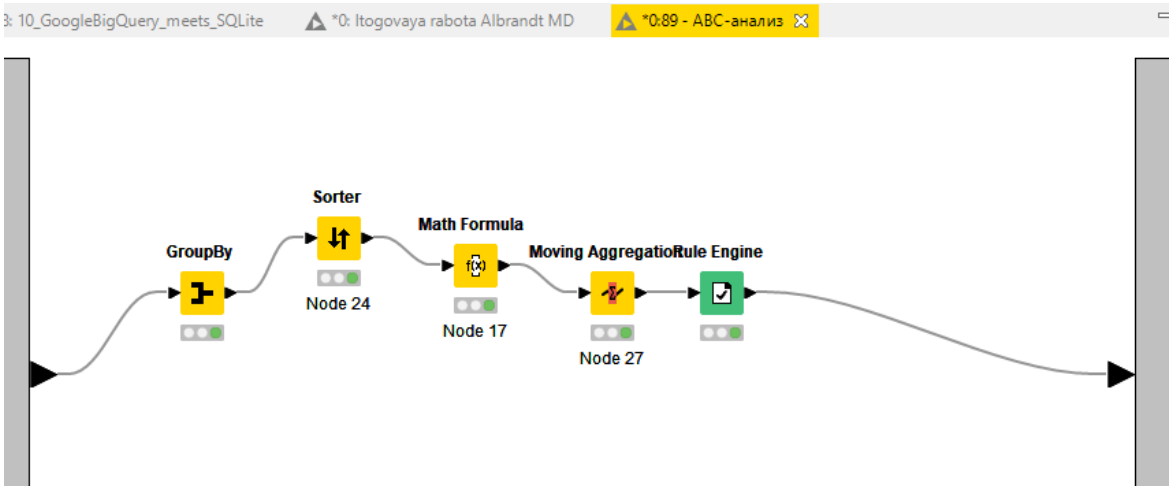
Было произведено объединение таблиц. При последнем объединении из итогового набора данных были исключены столбцы с индексами, которые дублируют информацию из присоединенных столбцов.

Также в набор данных был добавлен столбец OrderProcessing – обработка заказа, хранящий срок обработки каждого заказа до отправки в днях.

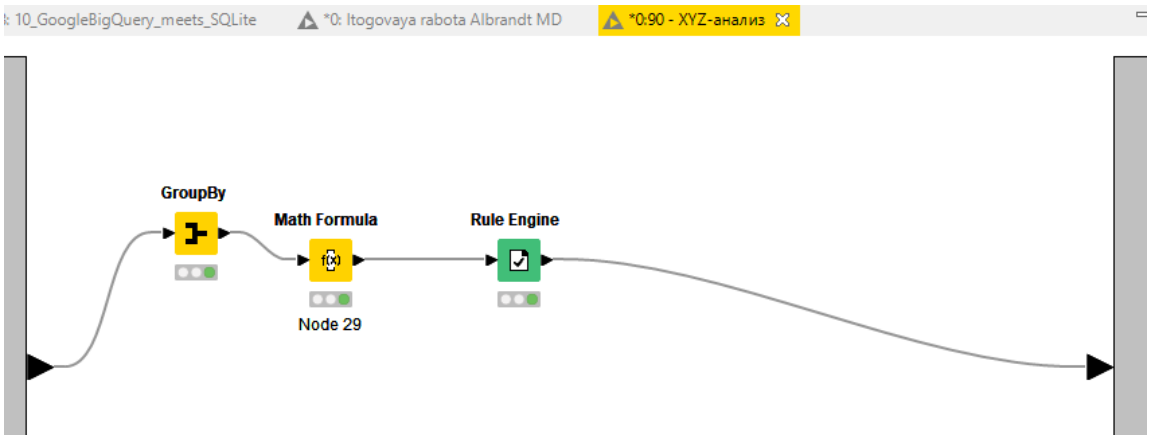
Была проведена проверка на наличие дубликатов в кейсе.



2. После формирование обобщенного набора данных, были исключены неинформативные столбцы содержащие коды. Проведен ABC – анализ продукции.



Также проведен XYZ- анализ продукции и проведено обогащение кейса.

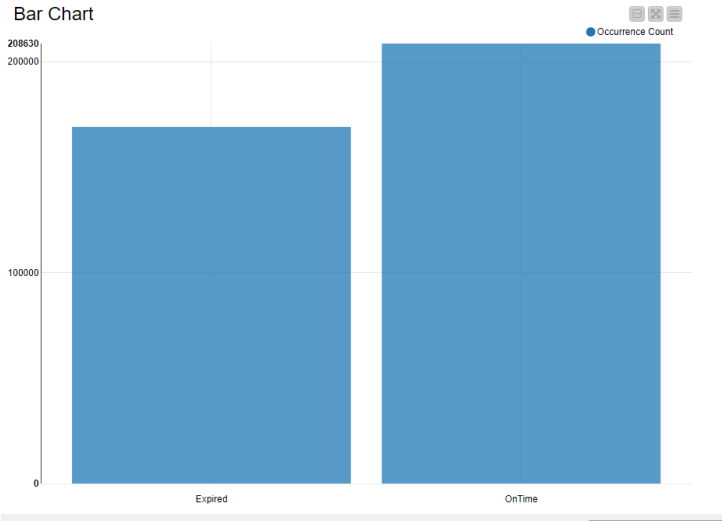


3. Данные имеют пропуски по полю Ship Date - дата отправки, процент пропусков составляет около 1% от всех данных, что привело к такому же проценту пропусков по добавленному полю OrderProcessing. Заполнить пропуски не представляется возможным, в связи с чем была были принято решение отказаться от поля Ship Date, т.к. данных о возвратах или потерях заказа у нас нет, а построение прогнозных моделей будет строится по дате заказа. Пропуски по добавленному полю OrderProcessin были заполненные, максимальным нормативным значений, в которые среднестатистический интернет магазин должен обработать заказ (14 дней), таким образом мы максимально сохраняем выборку данных.

Так как в наборе данных, предположительно, рассматривается работа интернет магазина радиоуправляемых моделей, то можно предположить, что срок обработки заказа от получения до момента отправки должен составлять не более 2 дней. Тогда по срокам обработки заказа данные можно классифицировать следующим образом:

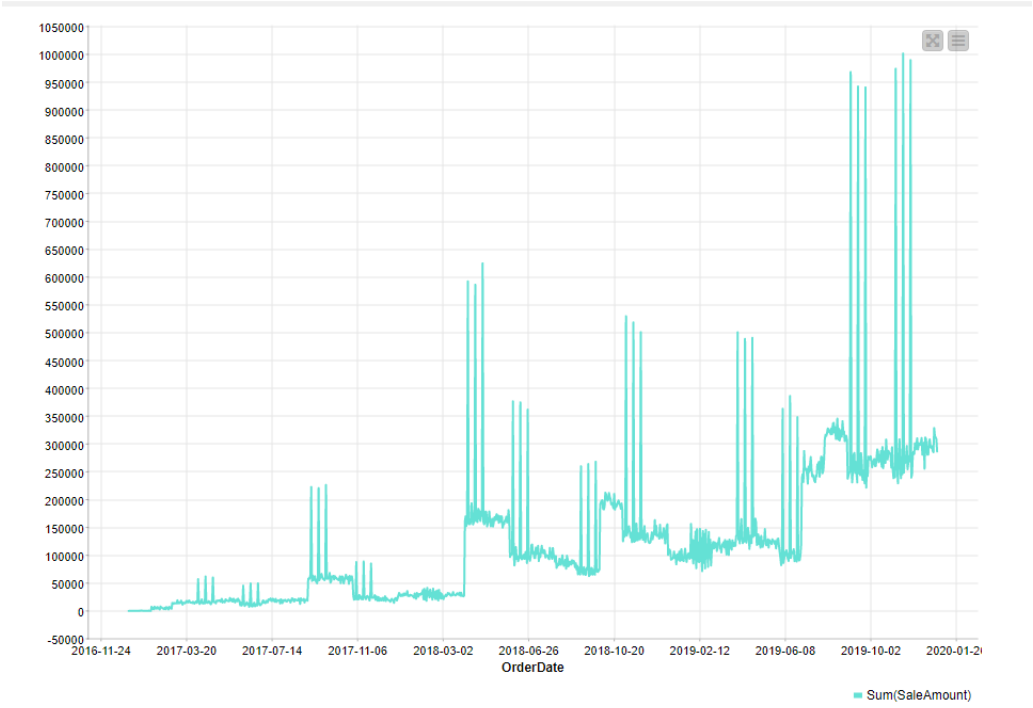
- 0-2 дня – OnTime - заказ обработан в рамках нормативных сроков;
- 3 дня и более - Expired - заказ просрочен.

По результату мы получаем набор данных с бинарной классификацией при незначительном дисбалансе классов.

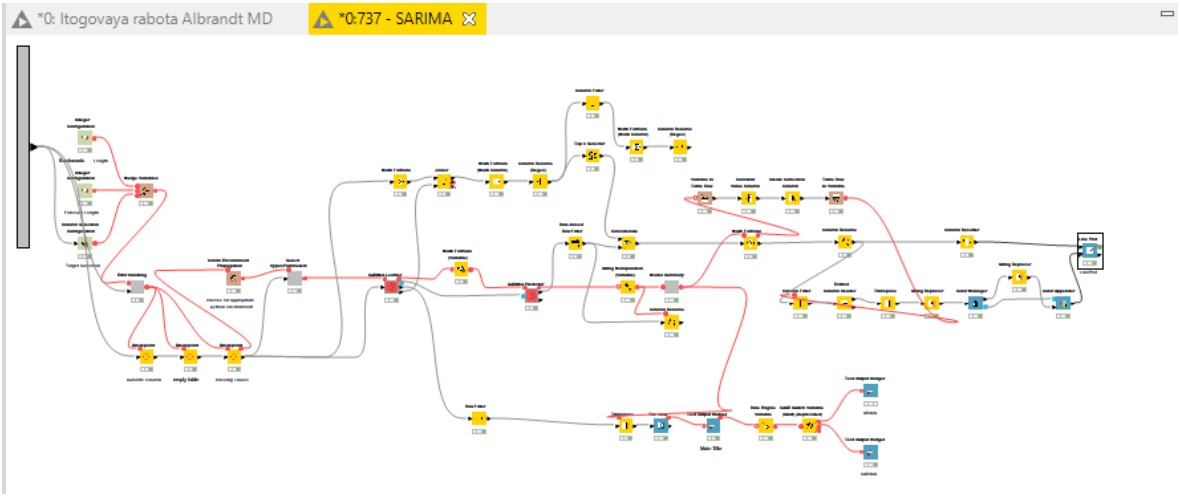


4. В рамках работы с данным в Knime Analytics Platform было обучено несколько моделей:

   4.1. Прогнозирование суммы продаж с использованием временных рядов. Для прогнозирования была выбрана модель AUTO-SARIMA, т.к. изменение данных имеет ярко выраженный сезонный характер.



   Для построения модели был сформирован одномерный временной ряд, содержащий Сумму продаж за каждый день, объем выборки составил 1085 строк, далее были заданы показатели сезонности и произведена авто настройка параметров.



На текущей выборке модель дала метрики не очень хорошего качества, при этом график прогноза будущих периодов показывает усредненные сезонные колебания без скачков до минимальных позиций. Согласно данному прогнозу, сумма продаж будет достаточно стабильной на протяжении года.

Insample Metrics

SARIMA(1,1,1)(0,1,1)2

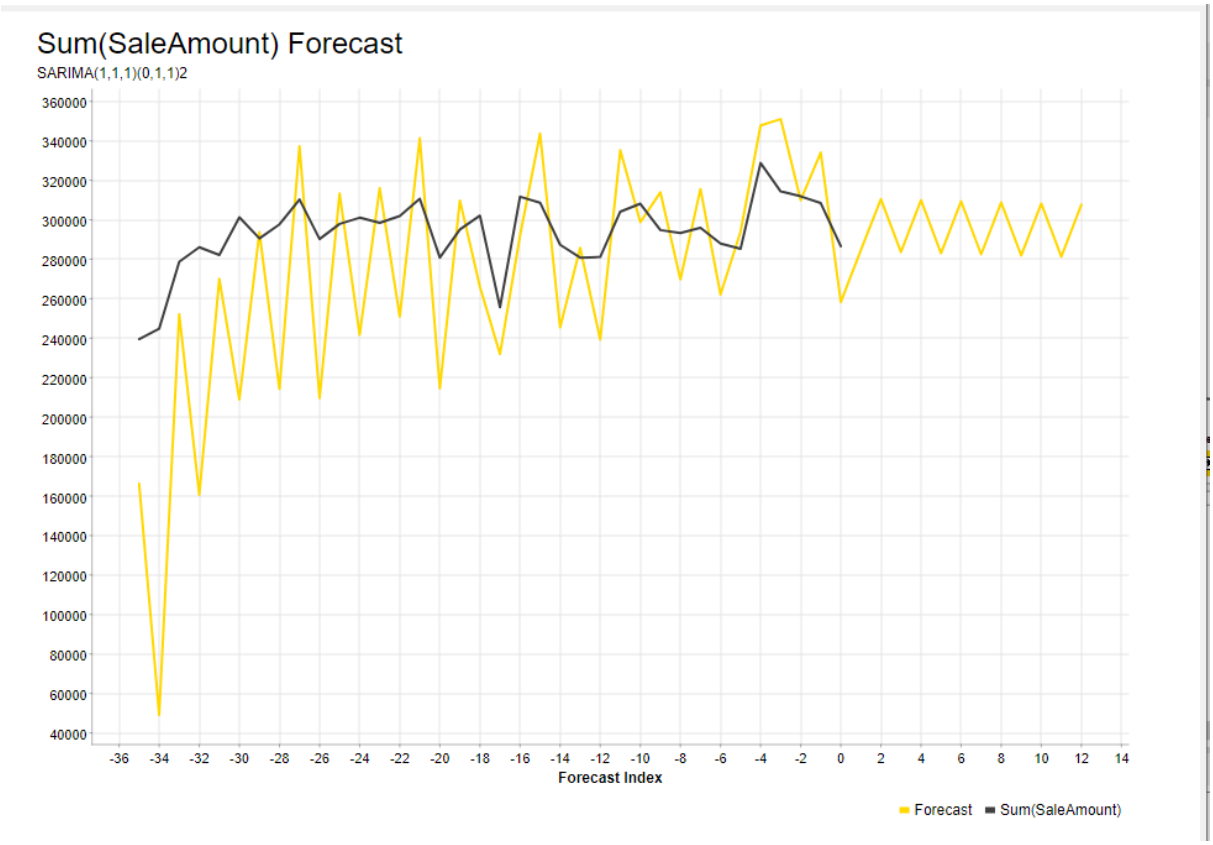
RMSE: 72031.79

MAE: 28841.08

MAPE: 0.26

R2: 0.64

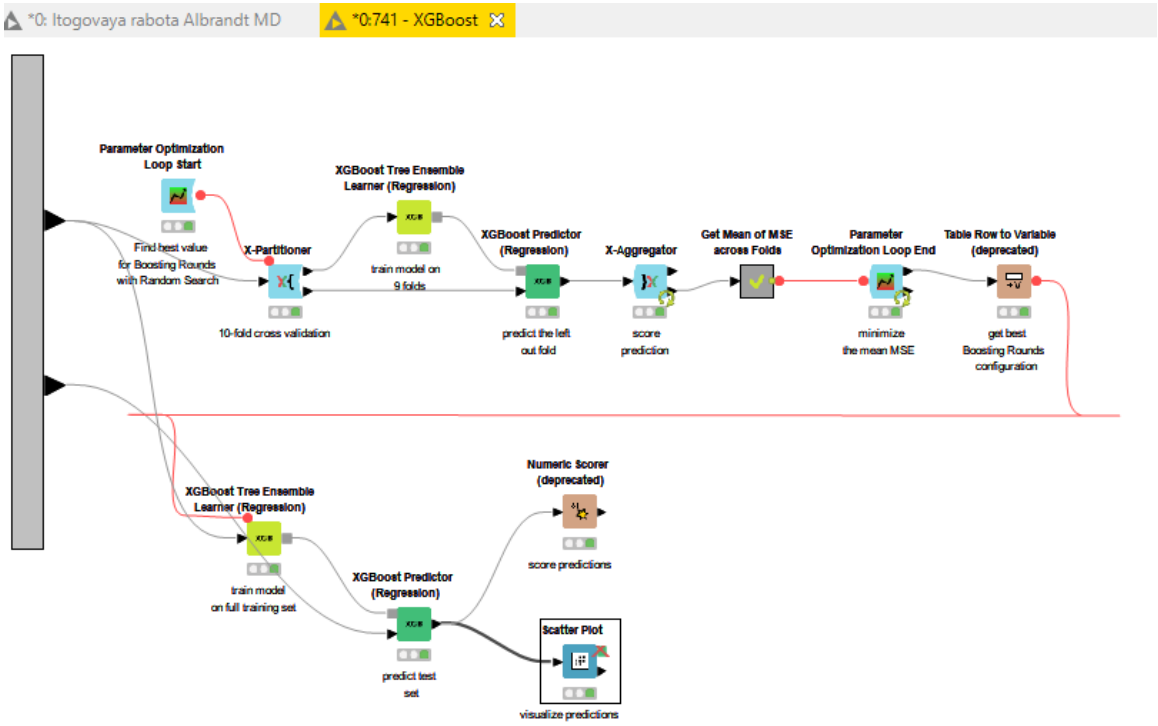
Showing 1 to 1 of 1 entries



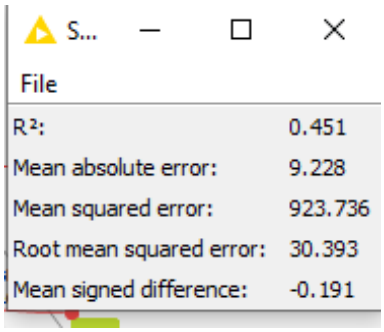
Следует учитывать, что данные требуют дополнительного изучения, а модель настройки параметров, и применения альтернативных методов прогнозирования для получения уточненного прогноза.

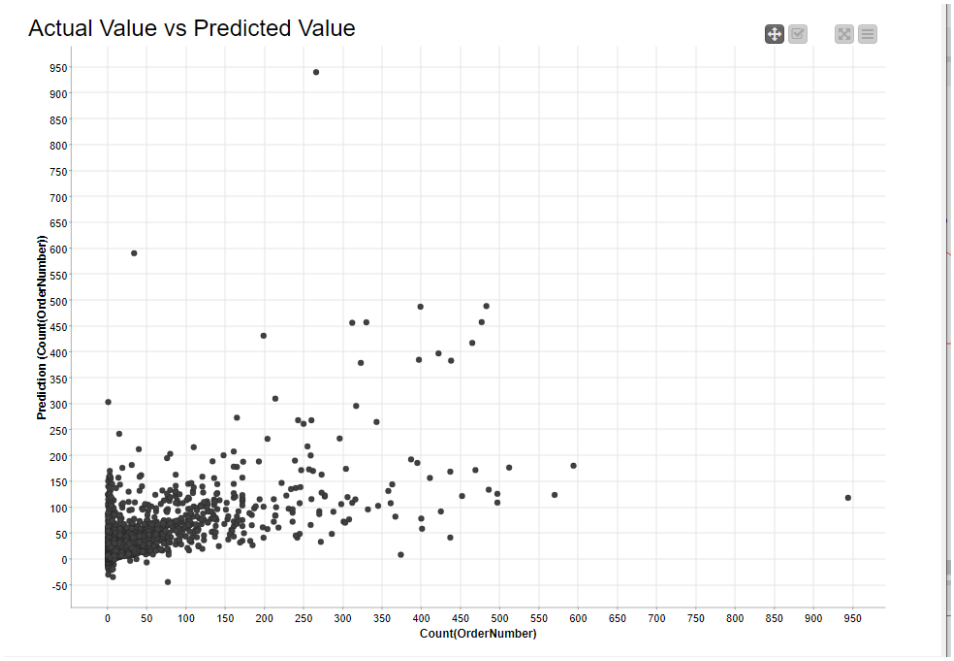
При дальнейшем изучении набора данных также построим модель прогноза продаж при использовании BI-платформ и библиотек Python.

4.2 Прогнозирование объема заказов исходя из характеристик товара, использования скидок и периода моделью XGBoost. Для частоты эксперимента из выборки были исключены данных о сумме продаж, и количестве проданных товаров в рамках заказов.

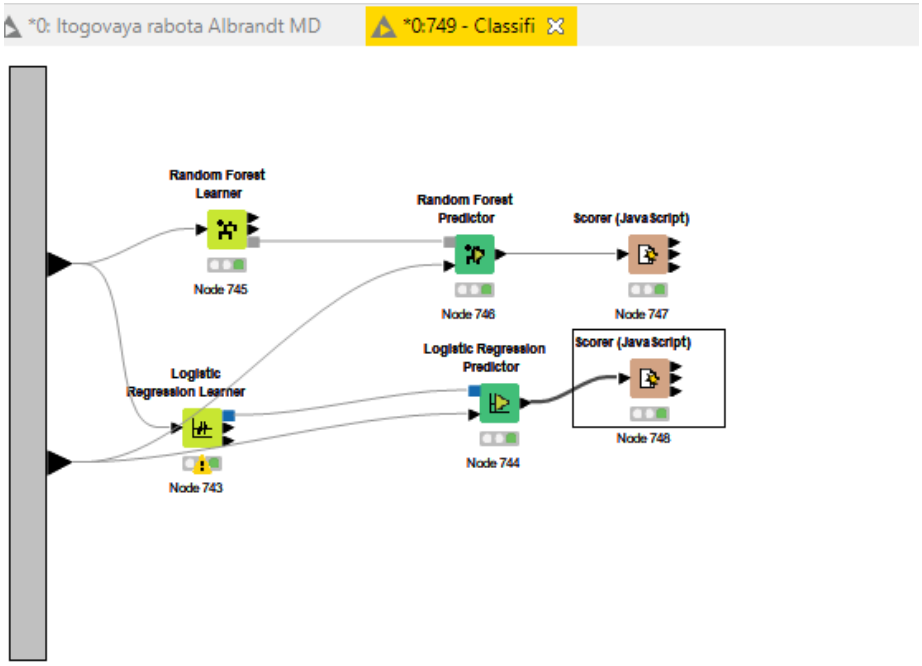


К сожалению метрики данной модели, так говорят о низкой точности, что еще раз подтверждает теорию, что данные требуют дополнительного анализа и изучения.





4.3. Так как у нас была введена классификация по нормативным срокам обработки заказов, были обучены модели бинарной классификации Логистическая регресия и Случайный лес.



Сначала обучение было проведено на максимальном наборе данных, исключено было только поле Номер заказа, т.к. обладало слишком большим набором уникальных значений и срок обработки заказа, так как оказывает прямое влияние на результирующий показатель. Метрики моделей были крайне слабыми.

Random forest

Confusion Matrix

	Expired (Predicted)	OnTime (Predicted)	
Expired (Actual)	548	50186	1.08%
OnTime (Actual)	688	61901	98.90%
	44.34%	55.23%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
55.11%	44.89%	-0.000	62449	50874

Logistic regression

Confusion Matrix

	Expired (Predicted)	OnTime (Predicted)	
Expired (Actual)	50734	0	100.00%
OnTime (Actual)	62589	0	0.00%
	44.77%	undefined	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
44.77%	55.23%	0.000	50734	62589

Затем набор атрибутов был сокращен, были убраны данные добавленные при обогащении набора, также данные о цене товара и скидках.

Random forest

Confusion Matrix

	Expired (Predicted)	OnTime (Predicted)	
Expired (Actual)	3	50731	0.01%
OnTime (Actual)	4	62585	99.99%
	42.86%	55.23%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
55.23%	44.77%	-0.000	62588	50735

Logistic regression

Confusion Matrix

	Expired (Predicted)	OnTime (Predicted)	
Expired (Actual)	0	50734	0.00%
OnTime (Actual)	0	62589	100.00%
	undefined	55.23%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
55.23%	44.77%	0.000	62589	50734

Метрики моделей улучшилий, но крайне не значительно. Что также говорит о том, что набор данных требует детального изучения и выявления связей, для выборки наилучших атрибутов для обучения моделей.
Дальнейший анализ набора будет проводится в *Colab и* BI-платформах.

3. EDA, расширенный анализ данных, обучение моделей регрессии и классификации с использованием библиотек Python.

Ссылки:

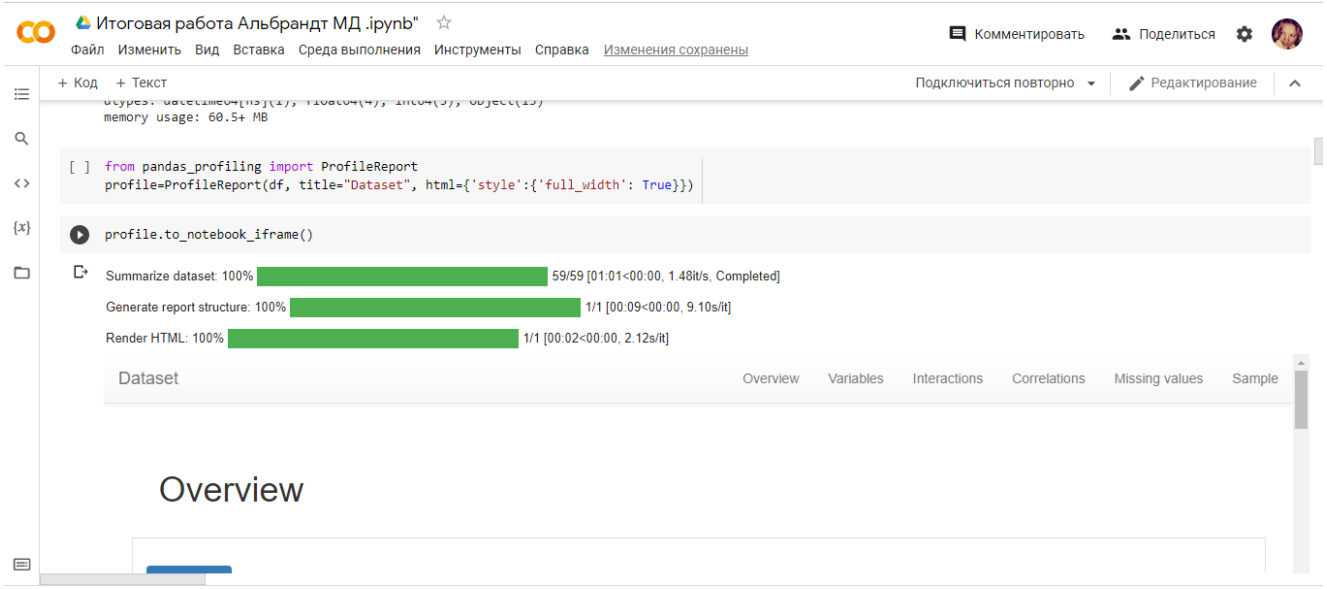
https://drive.google.com/file/d/1hq57DH-W2UxQW3tUaoM2lq55ilpWWdVr/view?usp=sharing – итоговый CSV-файл

https://colab.research.google.com/drive/1vxEPU9UaNXWNZFtUkXyS1KpubUU-HuUG?usp=sharing – блокнот Colab

Для дальнейшего анализа и моделирования данные были выгружены из Knime в csv-файл и размещены в Google Drive.

Был создан блокнот Colab для манипуляций с данными с использованием библиотек Python, так же был проведен анализ данных с использованием BI-платформ, подробнее в п.4.

В рамках работ в блокноте Colab был проведен исследовательский анализ данных с использованием библиотеки pandas-profiling.



https://drive.google.com/file/d/1qQwFLFKJJFtAyQJW3C9SNMyBS3It6Yxg/view?usp=sharing – отчет анализа данных.

Изучив анализ данных, были сделаны следующие выводы:

1. Признак **Order Number** - Имеет является уникальным как номер заказа, т.е. из 374741 строк набора данных поле имеет столько же уникальных значений. Также это нам говорит о том, что в рамках одного заказа была заявка только на одно наименование продукции. Для моделирования данный признак не информативен, поэтому при формировании под выборки он будет заменен на новый признак **Quantity Order** - количество заказав, по которому будет возможна агрегация для построения моделей регрессии.
2. Признак **Country**, добавленный нами для формирования корректной гео-локации в bi-платформах, состоит из константы и не представляет для нас ценности при моделировании, поэтому не будет включатся в локальные наборы данных;
3. Признак **Retail Price** является актуальной розничной ценой товаров, и соответствует значениям поля **Unit Price** на конец 2019 года, для построения моделей для нас данный показатель не актуален, поэтому можем им пренебречь.
4. Выявлена высокая корреляция между атрибутами Product **Category**, Item **Group**, **Kit Type**, **Channels**, **Demographic** на **Unit Price**, а, следовательно, и на **Sale Amount**, поэтому при построении моделей регрессии по оценки объема продаж, параметры **Unit Price** и **Sale Amount** следует исключить.

В рамках работ с данными проведены работы по замене показателя **Order Number** на новый признак **Quantity Order,** равный на начальном этапе 1 и имеющий целочисленное значение.

```
[ ]  df.insert(loc=len(df.columns), column='QuantityOrder', value=1)
```

```
df = df.drop(['OrderNumber'], axis=1)
df.info()
```

Итоговая работа Альбрандт МД .ipynb"

Файл Изменить Вид Вставка Среда выполнения Инструменты Справ

+ Код + Текст

```
Data columns (total 21 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   ProductName           377741 non-null  object
 1   ProductCategory       377741 non-null  object
 2   ItemGroup             377741 non-null  object
 3   KitType               377741 non-null  object
 4   Channels              377741 non-null  int64
 5   Demographic           377741 non-null  object
 6   RetailPrice           377741 non-null  float64
 7   OrderDate             377741 non-null  datetime64[ns]
 8   Quantity              377741 non-null  int64
 9   UnitPrice             377741 non-null  float64
 10  DiscountAmount        377741 non-null  float64
 11  SaleAmount            377741 non-null  float64
 12  RegionName            377741 non-null  object
 13  Country               377741 non-null  object
 14  StateName             377741 non-null  object
 15  OrderProcessing       377741 non-null  int64
 16  ABC                   377741 non-null  object
 17  XYZ                   377741 non-null  object
 18  ABC_XYZ               377741 non-null  object
 19  OrderProcessingSpeed  377741 non-null  object
 20  QuantityOrder         377741 non-null  int64
dtypes: datetime64[ns](1), float64(4), int64(4), object(12)
memory usage: 60.5+ MB
```

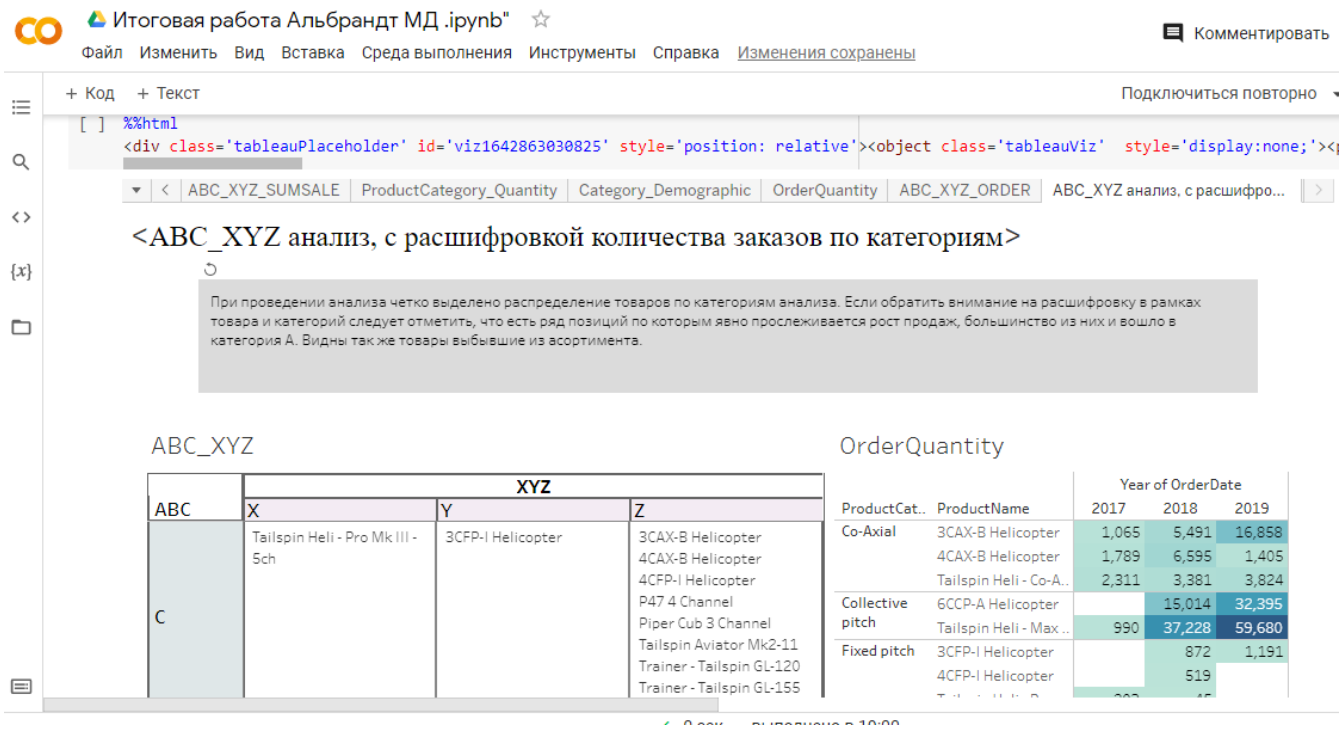
По набору данных были сформированы сводные таблицы и визуализации средствами Python по, внедрены интерактивные отчеты, сформированные на BI-платформах. Сформированы локальные выборки данных для обучения моделей. Для обучения была использована библиотека Pycaret и рассмотрено прогнозирование суммы продаж на основе временного ряда (подвыборка prsm - prognoz sum sale), модель регрессии при оценке количества продаж (подвыборка qo - Quantity Order), модель бинарной классификации для прогнозирования просрочки обработки заказов (подвыборка ops - OrderProcessingSpeed).

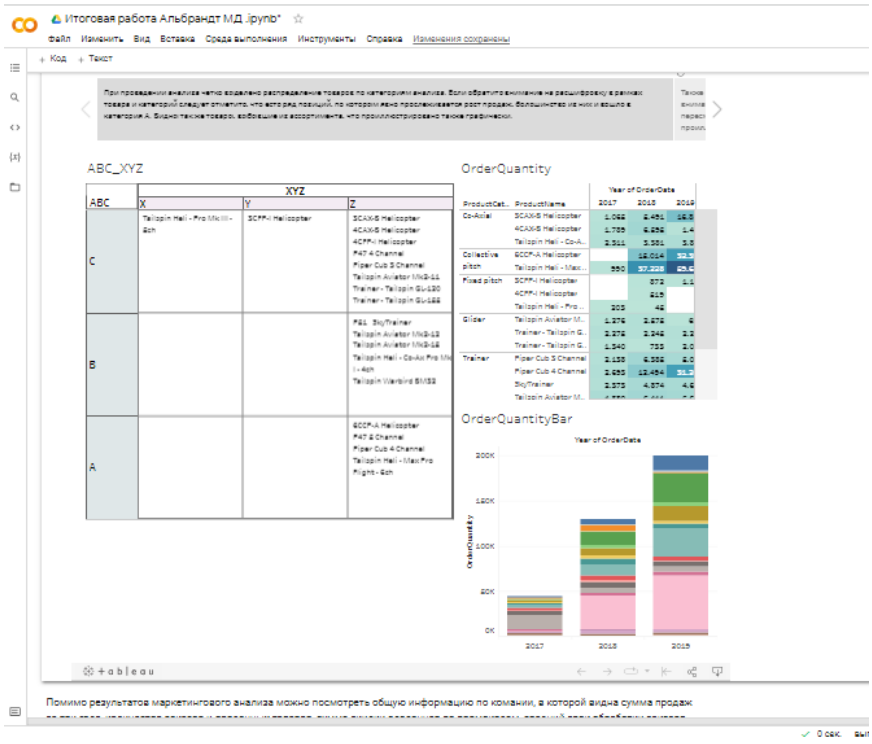
Была сформирована сводная таблица по распределению количества заказов между категориями ABC-XYZ – анализа.

```
pd.crosstab(df['ABC'],df['XYZ'], normalize=
```

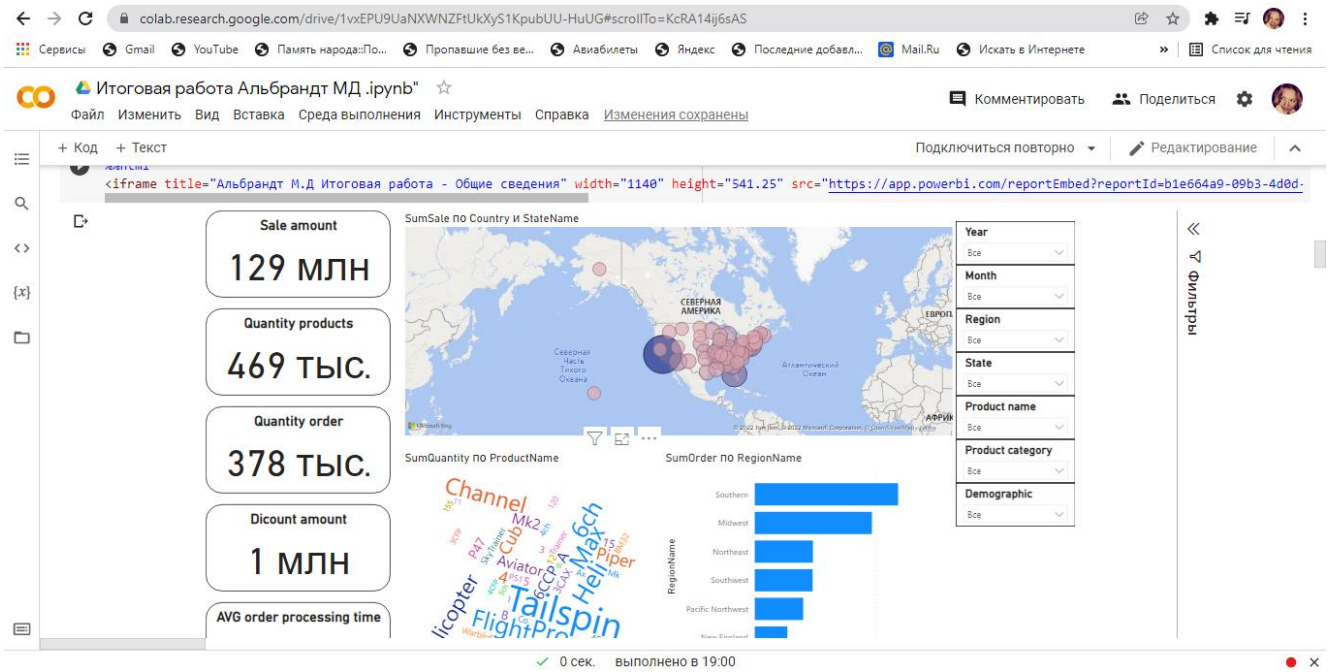
ABC	X	Y	Z
A	0	0	220381
B	0	0	82395
C	248	2080	72637

С точки зрения анализа продаваемой продукции, данная таблица не достаточна информативна, поэтому она была сопровождена внедрением интерактивного отчета из Tableau показывающего распределение продукции по категориям с разбивкой количества продаж по каждой позиции (отдельно ссылки на интерактивные отчеты bi-платформ будут в п.4.).

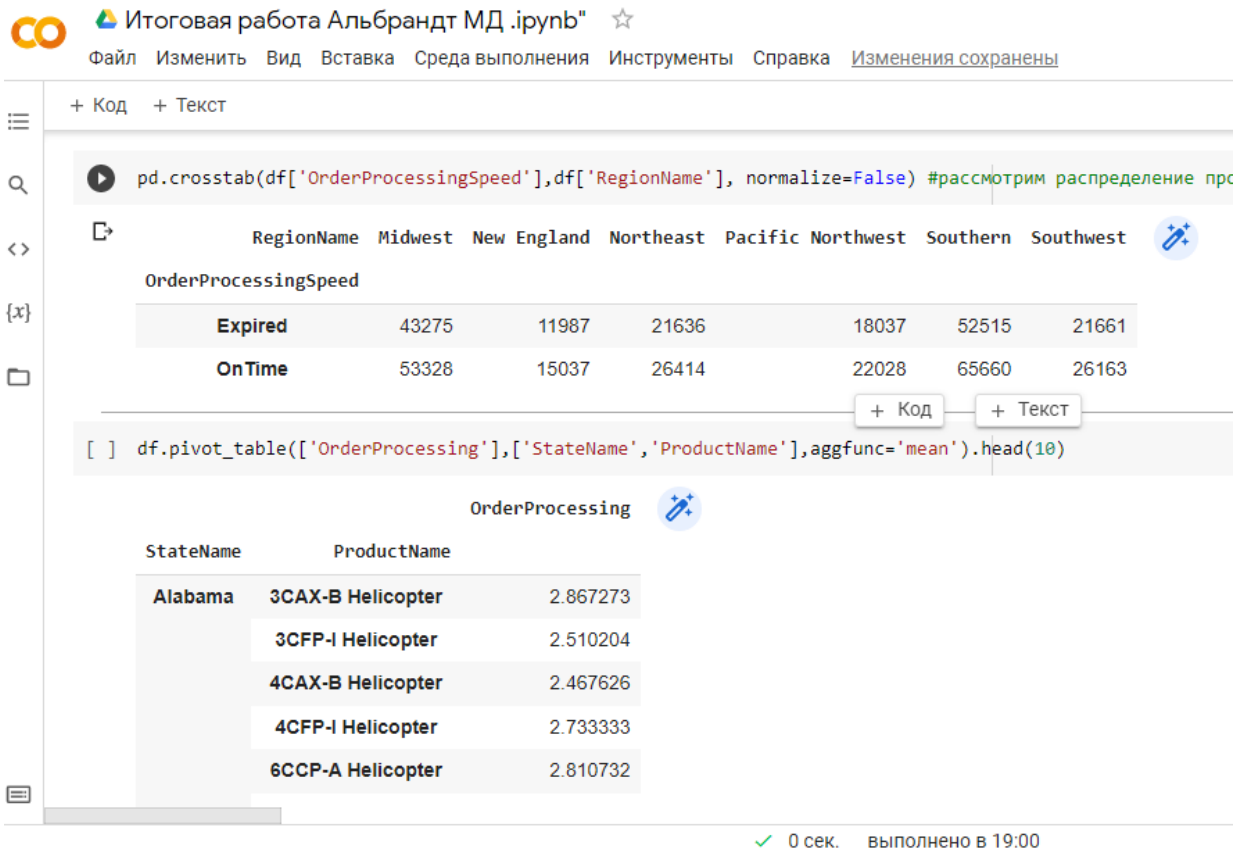


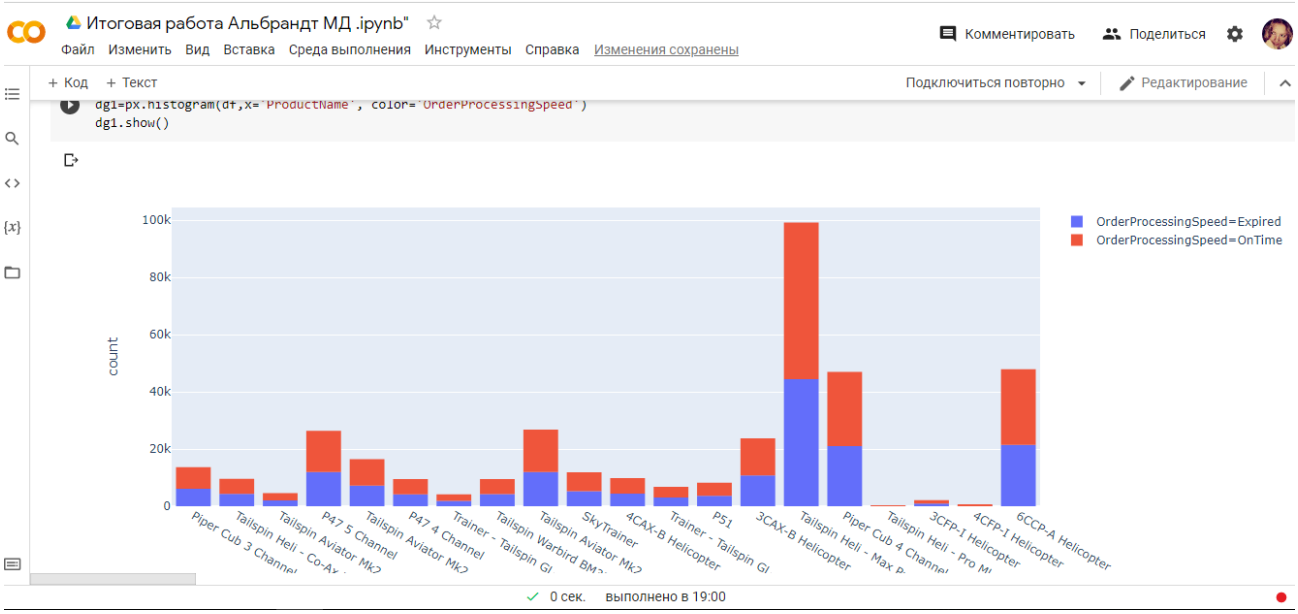


Помимо результатов маркетингового анализа был внедрен отчет, показывающий общую информацию по компании, в которой видна сумма продаж за три года, количество заказов и проданных товаров, сумма скидки по промокодам, средний срок обработки заказов, география распространения заказов. Данные в отчете интерактивны и позволяют выставить отборы для получения среза по любой комбинации. Отчет также содержит страницы с диаграммами по прогнозированию.

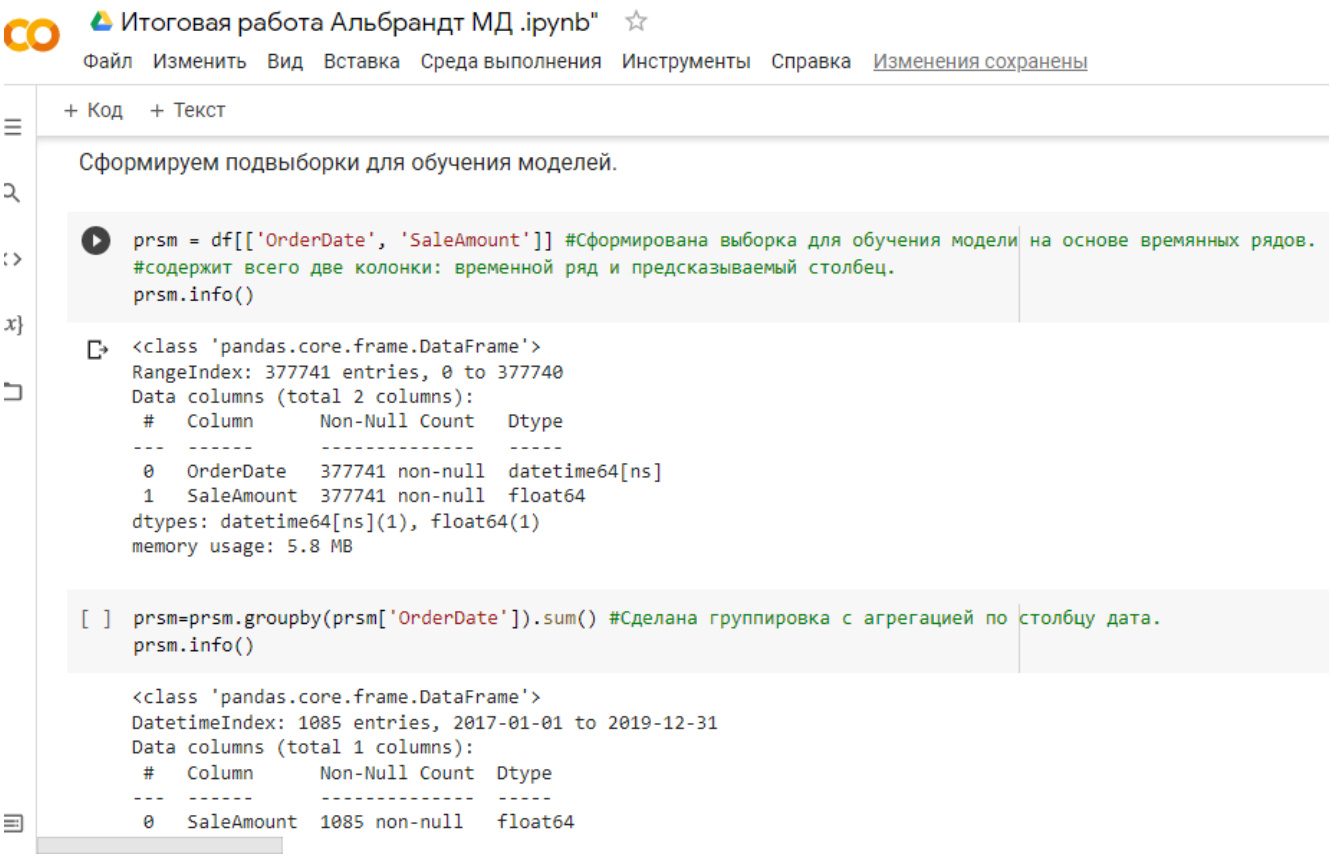


Также были сделаны сводные таблицы, для выявления среднего времени обработки заказов в разбивке регион, продукт и процентного распределение просрочек в обработке заказов в данном направлении с внедрением соответствующих визуализаций.



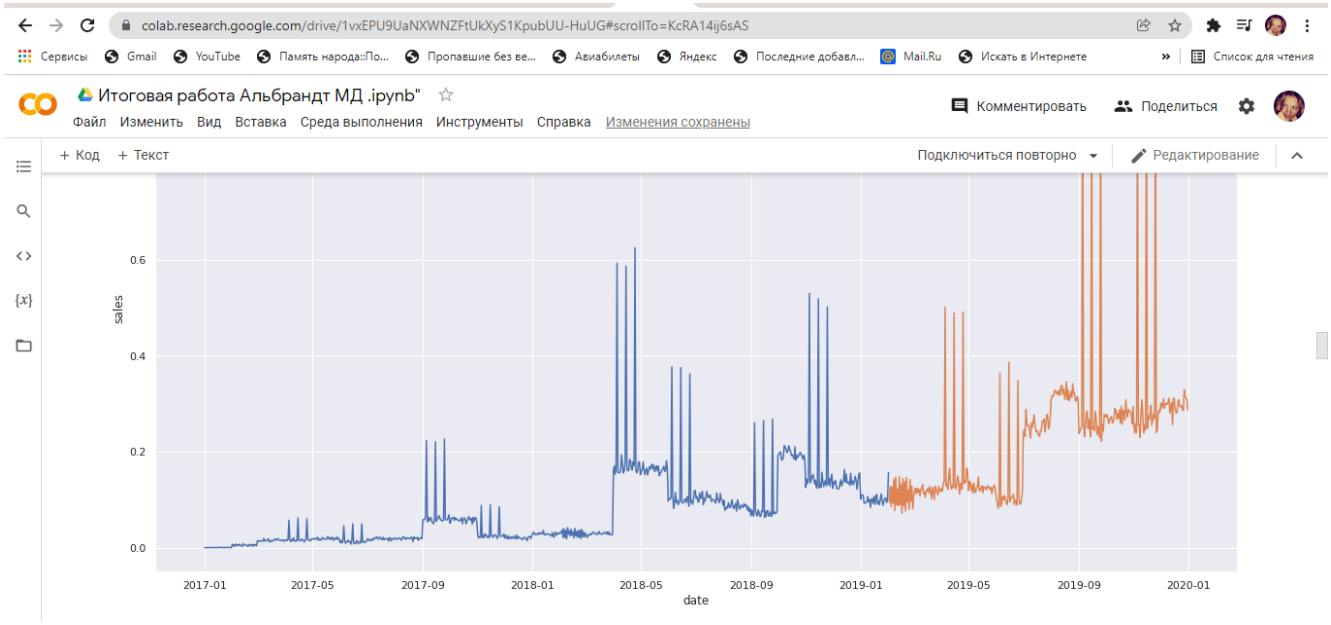


Сформированы подвыборки для моделирования.

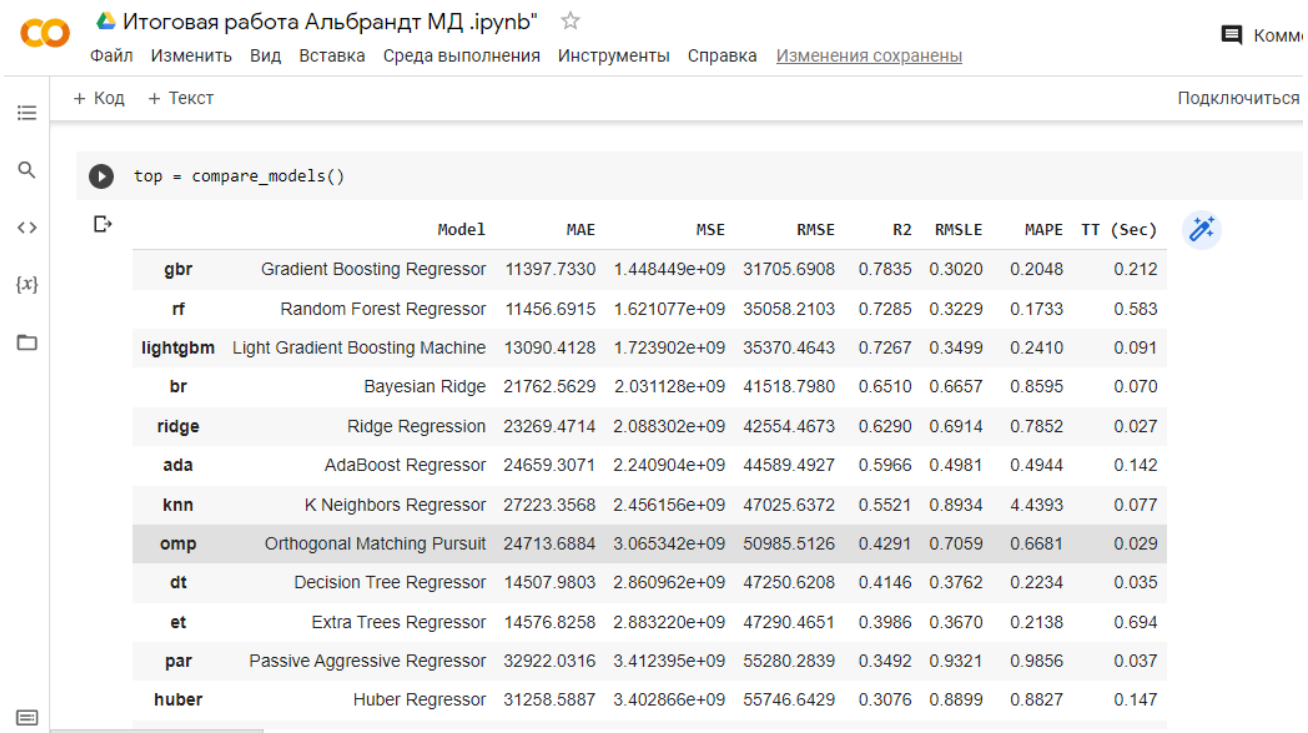


Для обучения модели прогнозирования суммы продаж на основе временных рядов была сформирована выборка, содержащая информацию о сумме продаж и дате заказа, с агрегацией по дате. Выборка аналогичная обучению модели SARIMA в Knime. Проведено выделение дополнительных показателей и разделения на обучающее и тестовое множество (Код прописан в блокноте раздел Временные ряды).

Особенность данных была описана в п.2., Временной ряд имеет ярко выраженную сезонность продаж.

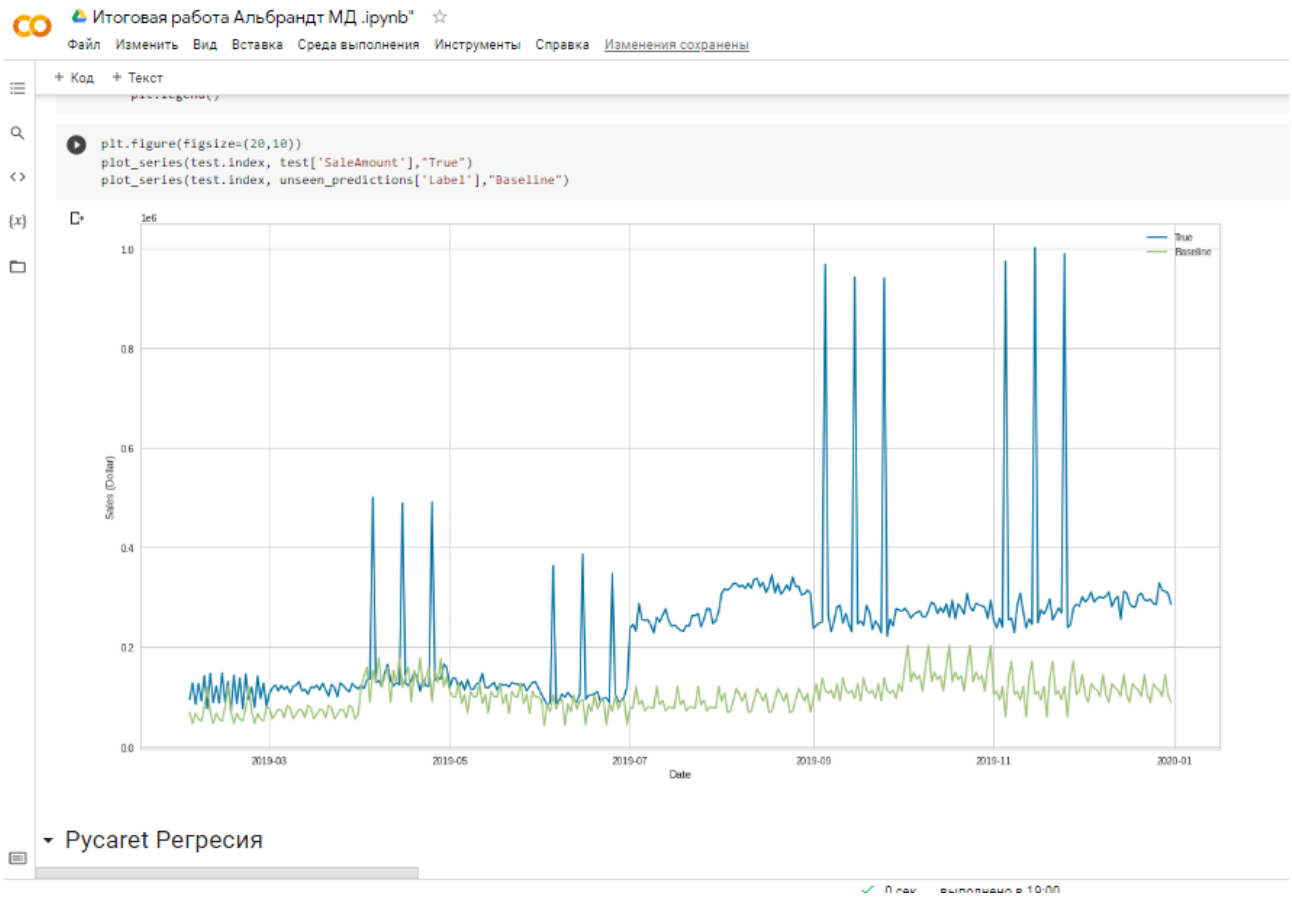


В обучении использовались регрессионные модели, наилучшие метрики были показаны моделью Gradient Boosting Regressor.



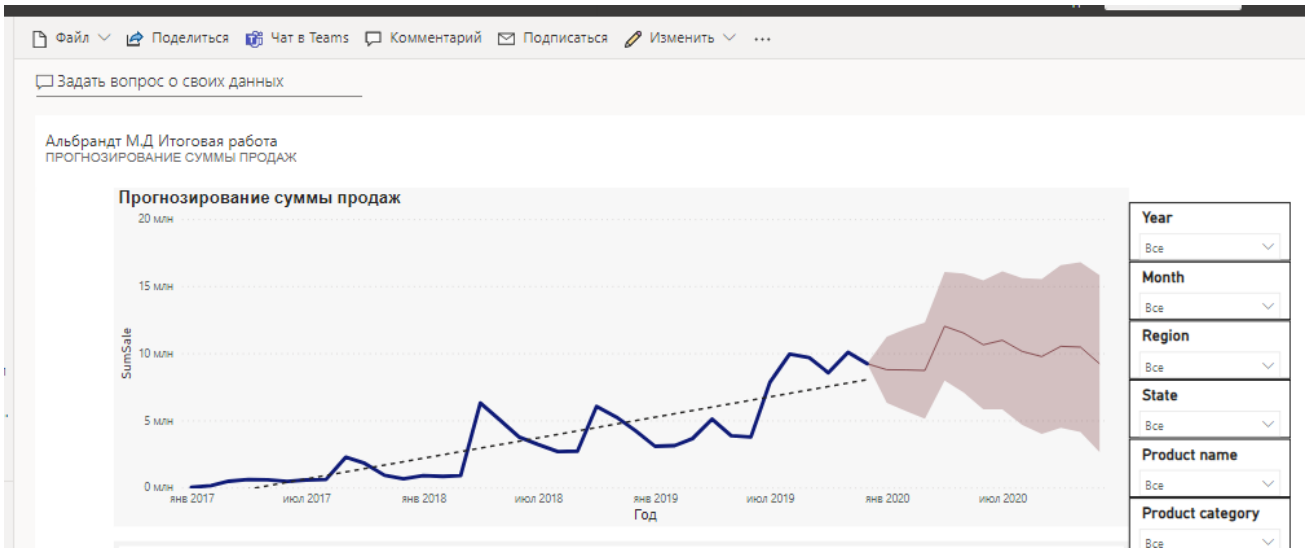
	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
gbr	Gradient Boosting Regressor	11397.7330	1.448449e+09	31705.6908	0.7835	0.3020	0.2048	0.212
rf	Random Forest Regressor	11456.6915	1.621077e+09	35058.2103	0.7285	0.3229	0.1733	0.583
lightgbm	Light Gradient Boosting Machine	13090.4128	1.723902e+09	35370.4643	0.7267	0.3499	0.2410	0.091
br	Bayesian Ridge	21762.5629	2.031128e+09	41518.7980	0.6510	0.6657	0.8595	0.070
ridge	Ridge Regression	23269.4714	2.088302e+09	42554.4673	0.6290	0.6914	0.7852	0.027
ada	AdaBoost Regressor	24659.3071	2.240904e+09	44589.4927	0.5966	0.4981	0.4944	0.142
knn	K Neighbors Regressor	27223.3568	2.456156e+09	47025.6372	0.5521	0.8934	4.4393	0.077
omp	Orthogonal Matching Pursuit	24713.6884	3.065342e+09	50985.5126	0.4291	0.7059	0.6681	0.029
dt	Decision Tree Regressor	14507.9803	2.860962e+09	47250.6208	0.4146	0.3762	0.2234	0.035
et	Extra Trees Regressor	14576.8258	2.883220e+09	47290.4651	0.3986	0.3670	0.2138	0.694
par	Passive Aggressive Regressor	32922.0316	3.412395e+09	55280.2839	0.3492	0.9321	0.9856	0.037
huber	Huber Regressor	31258.5887	3.402866e+09	55746.6429	0.3076	0.8899	0.8827	0.147

Она же была выбрана как конечная модель.



Уже при обучении модель показала, усредненные показатели, так же как Sarima.

Наиболее оптимистичный вид при моделировании показали визуализации прогнозов в Power Bi.



Для обучения модели регрессии была выбран показатель объема заказов и сформирована выборка, содержащая информацию о продукции и географии

распределения заказов. Для отражения показателя сезонности атрибут Дата заказа был заменен отдельно выделенными атрибутами Год, Месяц. Проведена агрегация по ключевому столбцу.

```
qo = df[['OrderDate','ProductName','ProductCategory','KitType','Channels','Demographic', 'DiscountAmount', 'RegionName', 'StateName','Quant
#Сформирована выборка для прогнозирования объема продаж, в зависимости от продуктовых и географических составляющих выборки
qo.info()
```

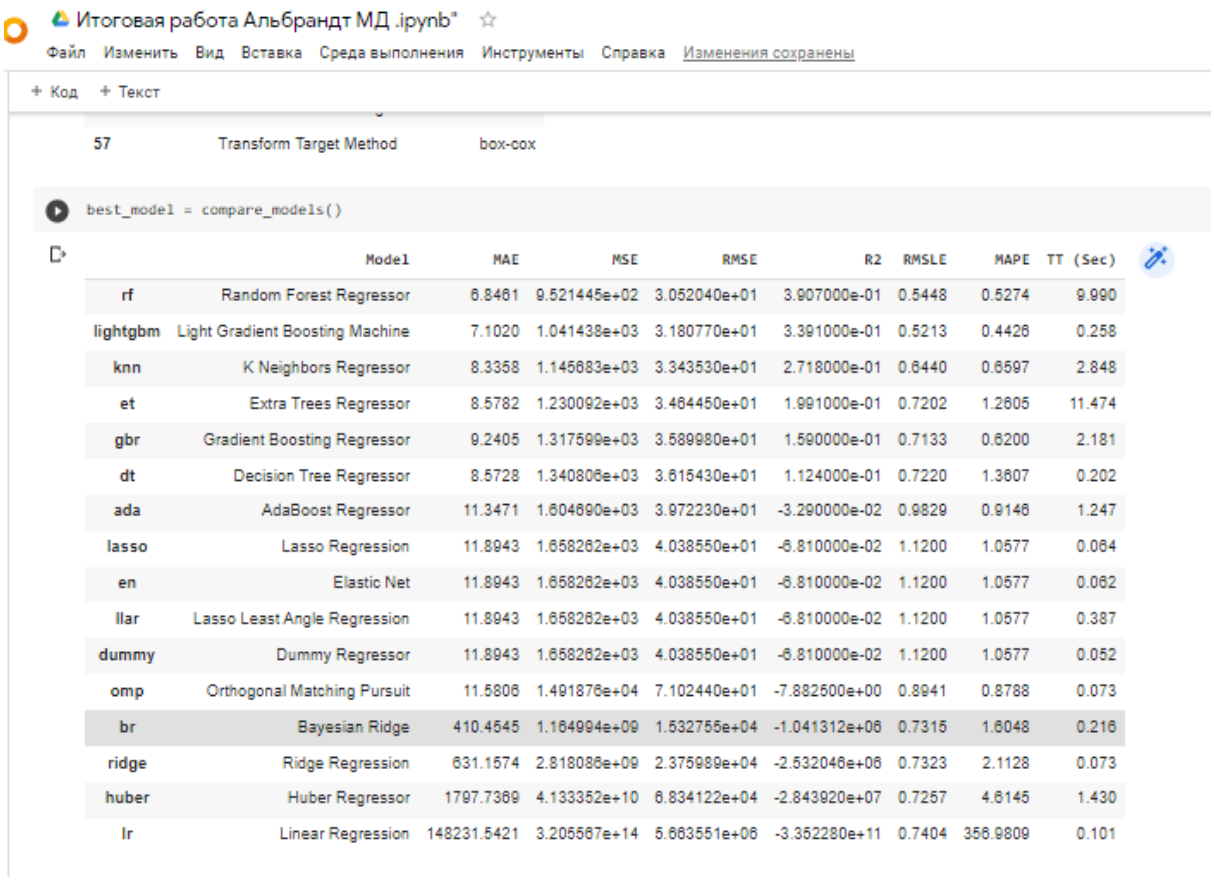
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 377741 entries, 0 to 377740
Data columns (total 10 columns):
 #   Column           Non-Null Count   Dtype
---  ------           --------------   -----
 0   OrderDate        377741 non-null  datetime64[ns]
 1   ProductName      377741 non-null  object
 2   ProductCategory  377741 non-null  object
 3   KitType          377741 non-null  object
 4   Channels         377741 non-null  int64
 5   Demographic      377741 non-null  object
 6   DiscountAmount   377741 non-null  float64
 7   RegionName       377741 non-null  object
 8   StateName        377741 non-null  object
 9   QuantityOrder    377741 non-null  int64
dtypes: datetime64[ns](1), float64(1), int64(2), object(6)
memory usage: 28.8+ MB
```

```
qo['Year']=qo['OrderDate'].dt.year
qo['Month']=qo['OrderDate'].dt.month
qo.info()
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
```

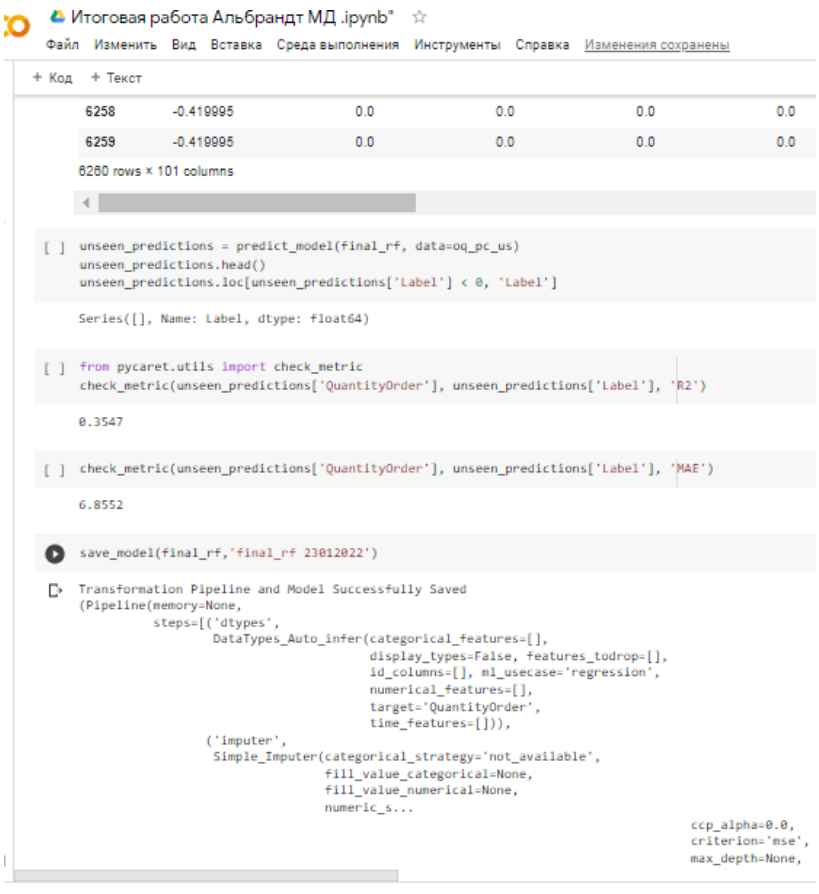
Проведено разделения на обучающее и тестовое множество (Код прописан в блокноте раздел Регрессия).

В обучении наилучшие метрики были показаны моделью Random Forest Regressor. Она же была выбрана как конечная модель. При тюнинге модель показала ухудшение метрик, поэтому к результату бралась модель до тюнинга показателей.

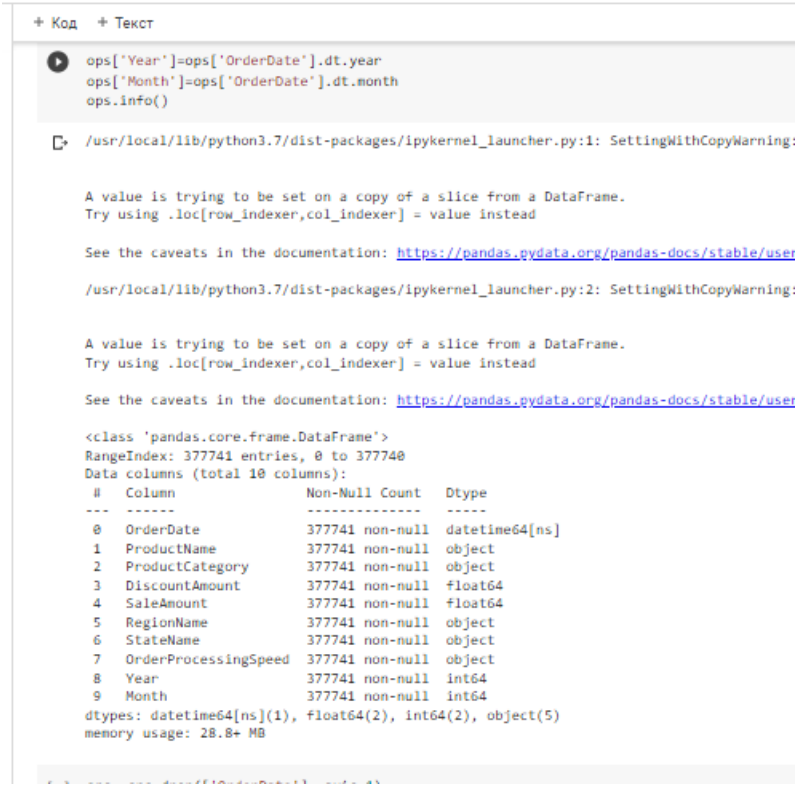


	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
rf	Random Forest Regressor	6.8461	9.521445e+02	3.052040e+01	3.907000e-01	0.5448	0.5274	9.990
lightgbm	Light Gradient Boosting Machine	7.1020	1.041438e+03	3.180770e+01	3.391000e-01	0.5213	0.4426	0.258
knn	K Neighbors Regressor	8.3358	1.145683e+03	3.343530e+01	2.718000e-01	0.6440	0.6597	2.848
et	Extra Trees Regressor	8.5782	1.230092e+03	3.464450e+01	1.991000e-01	0.7202	1.2605	11.474
gbr	Gradient Boosting Regressor	9.2405	1.317599e+03	3.589980e+01	1.590000e-01	0.7133	0.6200	2.181
dt	Decision Tree Regressor	8.5728	1.340806e+03	3.615430e+01	1.124000e-01	0.7220	1.3607	0.202
ada	AdaBoost Regressor	11.3471	1.604690e+03	3.972230e+01	-3.290000e-02	0.9829	0.9146	1.247
lasso	Lasso Regression	11.8943	1.658262e+03	4.038550e+01	-6.810000e-02	1.1200	1.0577	0.064
en	Elastic Net	11.8943	1.658262e+03	4.038550e+01	-6.810000e-02	1.1200	1.0577	0.062
llar	Lasso Least Angle Regression	11.8943	1.658262e+03	4.038550e+01	-6.810000e-02	1.1200	1.0577	0.387
dummy	Dummy Regressor	11.8943	1.658262e+03	4.038550e+01	-6.810000e-02	1.1200	1.0577	0.052
omp	Orthogonal Matching Pursuit	11.5806	1.491876e+04	7.102440e+01	-7.882500e+00	0.8941	0.8788	0.073
br	Bayesian Ridge	410.4545	1.164994e+09	1.532755e+04	-1.041312e+06	0.7315	1.6048	0.216
ridge	Ridge Regression	631.1574	2.818086e+09	2.375989e+04	-2.532046e+06	0.7323	2.1128	0.073
huber	Huber Regressor	1797.7369	4.133352e+10	6.834122e+04	-2.843920e+07	0.7257	4.6145	1.430
lr	Linear Regression	148231.5421	3.205567e+14	5.663551e+06	-3.352280e+11	0.7404	356.9809	0.101

При обучении модель показала метрики хуже, чем XGBoost, которую обучали в Knime.

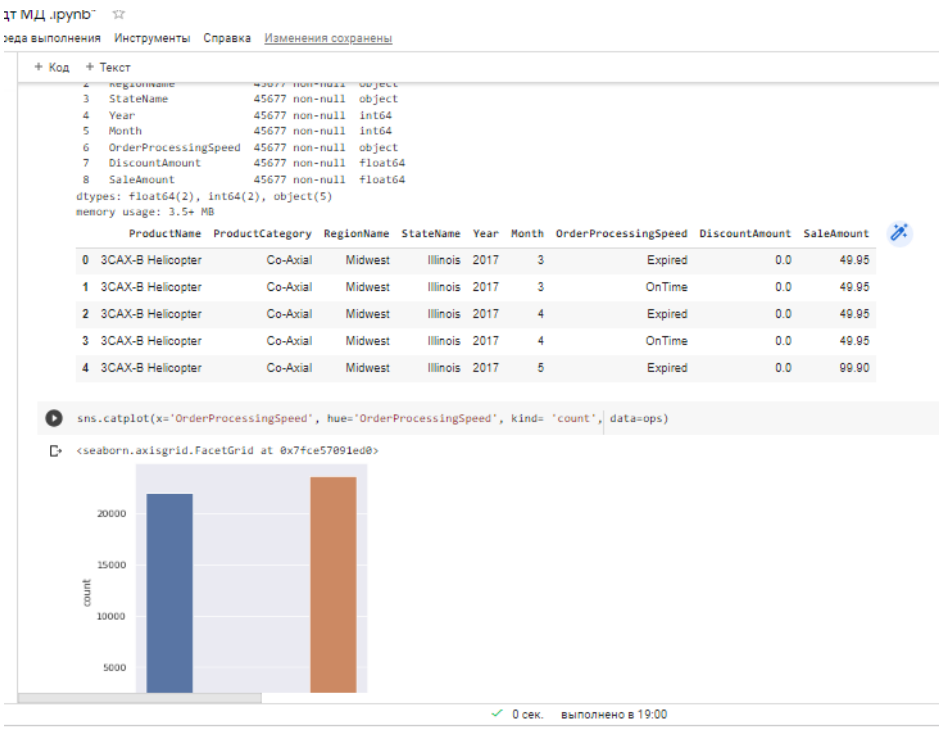


Для обучения модели классификации был выбран классификационный показатель срока обработки заказа (просрочен заказ или нет) и сформирована выборка, содержащая информацию о наименовании и категории продукции, географии распределения заказов, сумме продажи и скидок. Для отражения показателя сезонности атрибут Дата заказа был заменен отдельно выделенными атрибутами Год, Месяц. Проведена агрегация финансовых показателей по категориальным. Такой выбор атрибутов обусловлен анализом, проведенным в BI-системах.

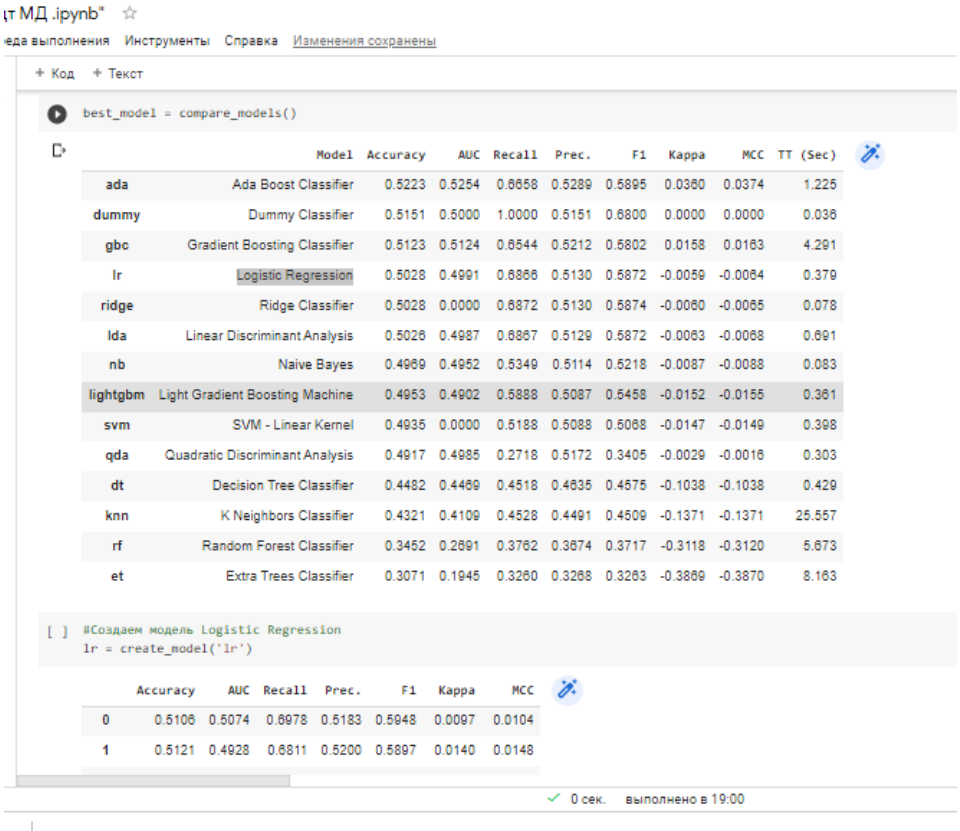


```
+ Код  + Текст

ops['Year']=ops['OrderDate'].dt.year
ops['Month']=ops['OrderDate'].dt.month
ops.info()

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 377741 entries, 0 to 377740
Data columns (total 10 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   OrderDate             377741 non-null  datetime64[ns]
 1   ProductName           377741 non-null  object
 2   ProductCategory       377741 non-null  object
 3   DiscountAmount        377741 non-null  float64
 4   SaleAmount            377741 non-null  float64
 5   RegionName            377741 non-null  object
 6   StateName             377741 non-null  object
 7   OrderProcessingSpeed  377741 non-null  object
 8   Year                  377741 non-null  int64
 9   Month                 377741 non-null  int64
dtypes: datetime64[ns](1), float64(2), int64(2), object(5)
memory usage: 28.8+ MB
```

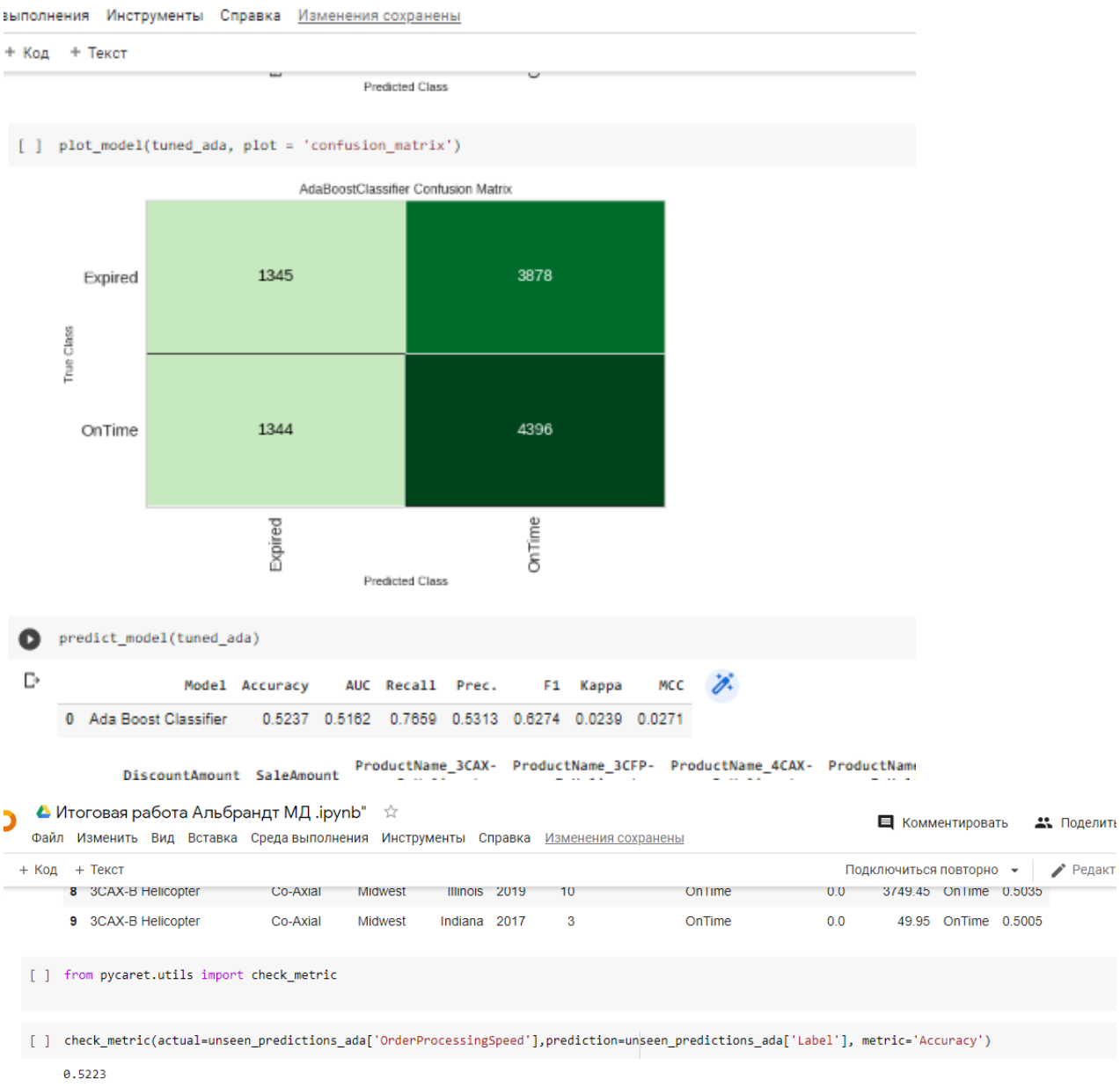
Проведено разделения на обучающее и тестовое множество (Код прописан в блокноте раздел Классификация), проведена оценка подмножества на дисбаланс по результирующему показателю.



В обучении наилучшие метрики были показаны моделью Ada Boost Classifier. Она же была выбрана как конечная модель.



При обучении модель показала метрики хуже, чем модели Логистической регрессии и Случайного леса, которые обучали в Knime.



4. Анализ данных, обучение моделей регрессии и классификации с использованием BI-платформ.

Ссылки:

https://app.powerbi.com/view?r=eyJrIjoiYmM4MjMzNWItY2FmMi00YzYzLTgyMDEtNDQ3MDA5YTU1YWI0IiwidCI6ImM4YzY5YWFlLTMyYmEtNDNkMS05ZjU5LWY5OGM5NWZiMjI3YiIsImMiOjl9&pageName=ReportSectionff30e3949a605a5973d1 – отчет Power BI

https://public.tableau.com/shared/HWWBGY6X3?:display_count=n&:origin=viz_share_link - отчеты в Tableau

https://public.tableau.com/shared/SZWPGPCCM?:display_count=n&:origin=viz_share_link – отчеты в Tableau расширенная версия

В рамках работы был проведен анализ данных с использованием BI-платформ Tableau и Power BI, созданы интерактивные отчеты и внедрены в блокнот Colab.

Так же на платформе Power BI были созданными модели машинного обучения классификации и регрессии.

На платформе Tableau был создан интерактивный отчет показывающий матрицу товаров в разбивке по ABC-XYZ-анализу, с выведенной доп. Аналитикой объема продаж по каждому наименованию.

<ABC_XYZ анализ, с расшифровкой количества заказов по категориям>

При проведении анализа четко выделено распределение товаров по категориям анализа. Если обратить внимание на расшифровку в рамках товара и категорий следует отметить, что есть ряд позиций, по которым явно прослеживается рост продаж, большинство из них и вошло в категория А. Видны так же товары, выбывшие из ассортимента, что проиллюстрировано также графически.

Также можно пр оанализировать.

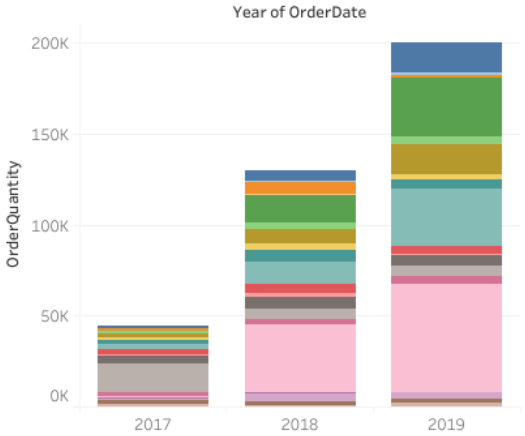
ABC_XYZ

ABC	XYZ		
	X	Y	Z
C	Tailspin Heli - Pro Mk III - 5ch	3CFP-I Helicopter	3CAX-B Helicopter 4CAX-B Helicopter 4CFP-I Helicopter P47 4 Channel Piper Cub 3 Channel Tailspin Aviator Mk2-11 Trainer - Tailspin GL-120 Trainer - Tailspin GL-155
B			P51 SkyTrainer Tailspin Aviator Mk2-12 Tailspin Aviator Mk2-15 Tailspin Heli - Co-Ax Pro Mk I - 4ch Tailspin Warbird BM32
A			6CCP-A Helicopter P47 5 Channel Piper Cub 4 Channel Tailspin Heli - Max Pro Flight - 6ch

OrderQuantity

ProductCat..	ProductName	Year of OrderDate 2017	2018	2019
Co-Axial	3CAX-B Helicopter	1,065	5,491	16,858
	4CAX-B Helicopter	1,789	6,595	1,405
	Tailspin Heli - Co-A..	2,311	3,381	3,824
Collective pitch	6CCP-A Helicopter		15,014	32,395
	Tailspin Heli - Max ..	990	37,228	59,680
Fixed pitch	3CFP-I Helicopter		872	1,191
	4CFP-I Helicopter		519	
	Tailspin Heli - Pro ..	203	45	
Glider	Tailspin Aviator M..	1,276	2,575	698
	Trainer - Tailspin G..	2,275	2,245	2,259
	Trainer - Tailspin G..	1,340	733	2,011
Trainer	Piper Cub 3 Channel	2,138	6,385	5,012
	Piper Cub 4 Channel	2,693	12,494	31,265
	SkyTrainer	2,373	4,874	4,615
	Tailspin Aviator M..	4,330	6,411	5,605

OrderQuantityBar



Данный отчет позволяет отследить динамику продаж по каждой группе, либо вхождение конкретной позиции из списка в категорию согласно анализу.

<ABC_XYZ анализ, с расшифровкой количества заказов по категориям>

При проведении анализа четко выделено распределение товаров по категориям анализа. Если обратить внимание на расшифровку в рамках товара и категорий следует отметить, что есть ряд позиций, по которым явно прослеживается рост продаж, большинство из них и вошло в категория А. Видны так же товары, выбывшие из ассортимента, что проиллюстрировано также графически.

Также можно пр оанализировать. .

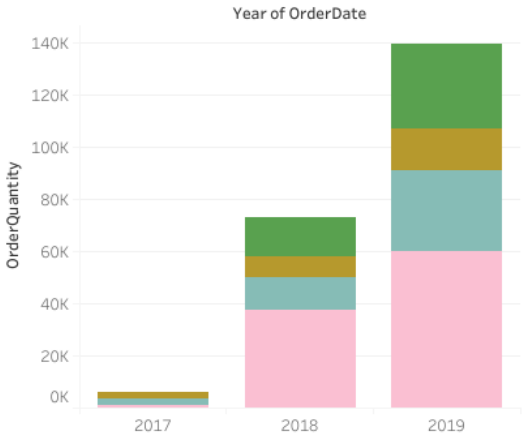
ABC_XYZ

ABC	XYZ		
	X	Y	Z
C	Tailspin Heli - Pro Mk III - 5ch	3CFP-I Helicopter	3CAX-B Helicopter 4CAX-B Helicopter 4CFP-I Helicopter P47 4 Channel Piper Cub 3 Channel Tailspin Aviator Mk2-11 Trainer - Tailspin GL-120 Trainer - Tailspin GL-155
B			P51 SkyTrainer Tailspin Aviator Mk2-12 Tailspin Aviator Mk2-15 Tailspin Heli - Co-Ax Pro Mk I - 4ch Tailspin Warbird BM32
A			6CCP-A Helicopter P47 5 Channel Piper Cub 4 Channel Tailspin Heli - Max Pro Flight - 6ch

OrderQuantity

ProductCat..	ProductName	Year of OrderDate 2017	2018	2019
Collective pitch	6CCP-A Helicopter		15,014	32,395
	Tailspin Heli - Max ..	990	37,228	59,680
Trainer	Piper Cub 4 Channel	2,693	12,494	31,265
Warbird	P47 5 Channel	2,091	8,001	15,995

OrderQuantityBar



Добавлена страница по анализу суммы продаж в разбивке по годам и категориям товаров, по количеству продаж внутри категорий и признаков, с возможностью интерактивного отслеживания отдельных составляющих.

<ABC_XYZ анализ, с расшифровкой количества заказов по категориям>

При проведении анализа четко выделено распределение ..

Также можно проанализировать и соотнести распределение продаж в разрезе категорий, уровню сложности и ABC-XYZ- разбивке. Если обратить внимание на разбивку в рамках уровня сложности и категории, по которым явно прослеживается рост продаж и отнесение к группе A, можно пересмотреть ассортимент продукции, расширив его тем самым по наиболее прибыльным либо стабильным направлениям, что проиллюстрировано также графически.

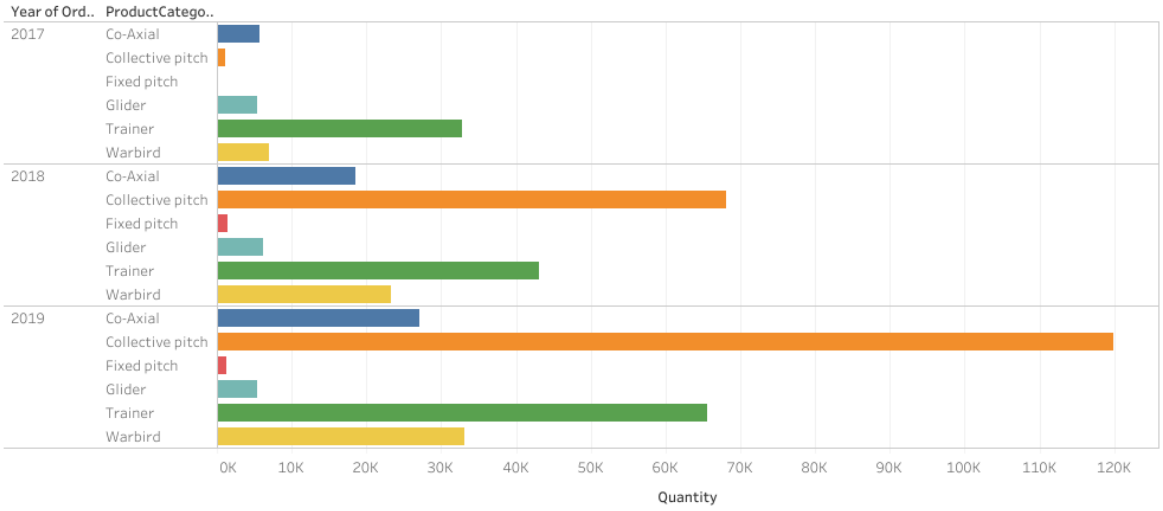
Category_Demographic

	ProductCategory					
Demograph..	Co-Axial	Collective pitch	Fixed pitch	Glider	Trainer	Warbird
Advanced			248		13,826	52,364
Beginner			2,667		15,926	
Intermediate	22,473			9,468	111,779	10,899
Novice	28,842			7,405		
Professional		189,156				

ABC_XYZ_SUMSALE

	OrderDate		
ABC_XYZ	2017	2018	2019
AZ	1,579,220	31,282,029	65,575,606
BZ	6,194,981	7,240,505	7,464,188
CX	64,466	14,206	
CY		104,247	168,976
CZ	1,198,752	3,222,269	3,162,305

ProductCategory_Quantity Bar



<ABC_XYZ анализ, с расшифровкой количества заказов по категориям>

При проведении анализа четко выделено распределение ..

Также можно проанализировать и соотнести распределение продаж в разрезе категорий, уровню сложности и ABC-XYZ- разбивке. Если обратить внимание на разбивку в рамках уровня сложности и категории, по которым явно прослеживается рост продаж и отнесение к группе A, можно пересмотреть ассортимент продукции, расширив его тем самым по наиболее прибыльным либо стабильным направлениям, что проиллюстрировано также графически.

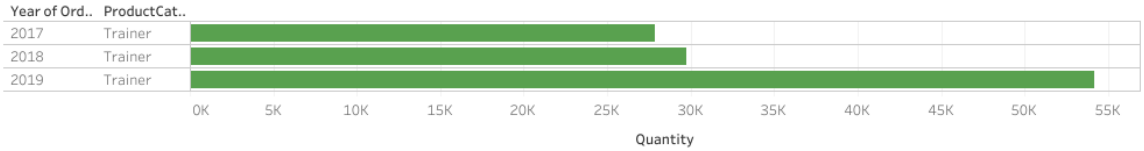
Category_Demographic

	ProductCategory					
Demograph..	Co-Axial	Collective pitch	Fixed pitch	Glider	Trainer	Warbird
Advanced			248		13,826	52,364
Beginner			2,667		15,926	
Intermediate	22,473			9,468	111,779	10,899
Novice	28,842			7,405		
Professional		189,156				

ABC_XYZ_SUMSALE

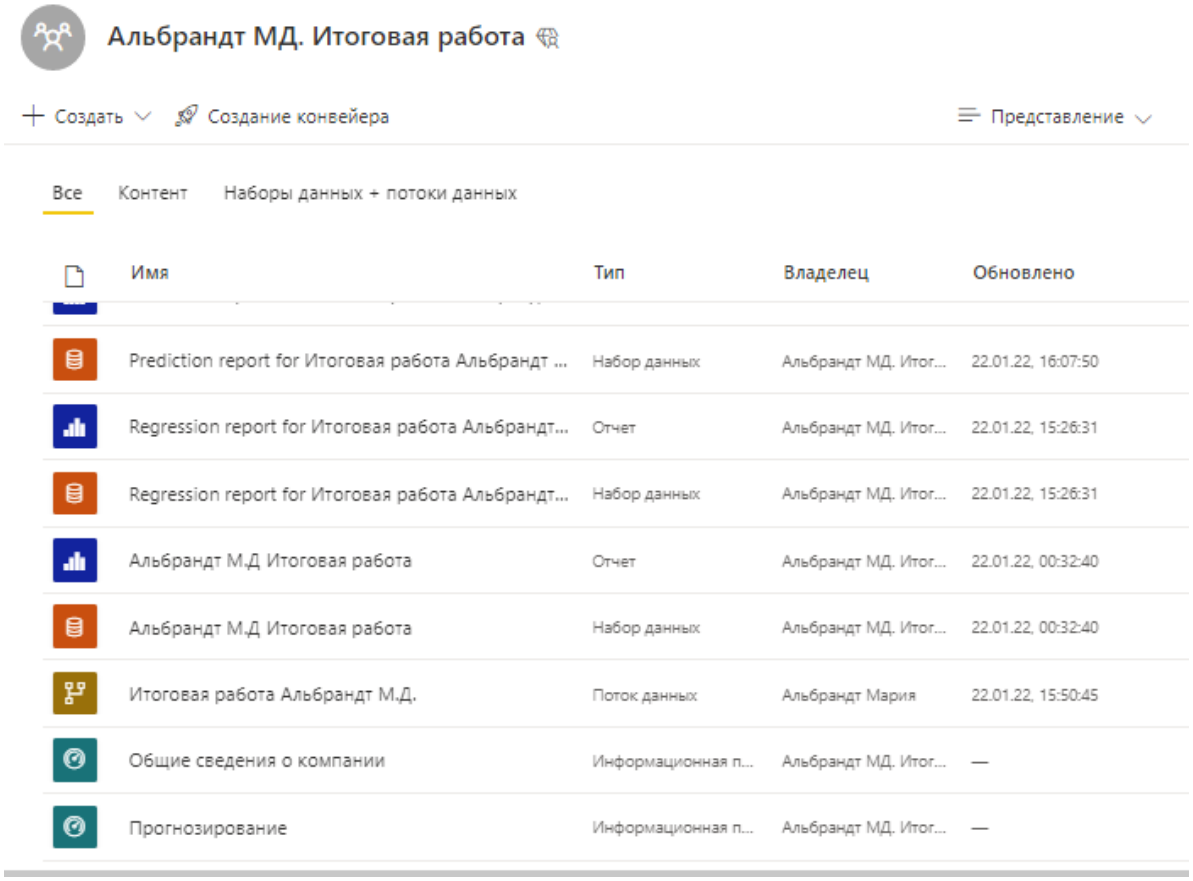
	OrderDate		
ABC_XYZ	2017	2018	2019
AZ	488,828	2,761,022	7,552,364
BZ	4,093,602	2,457,220	2,622,212

ProductCategory_Quantity Bar



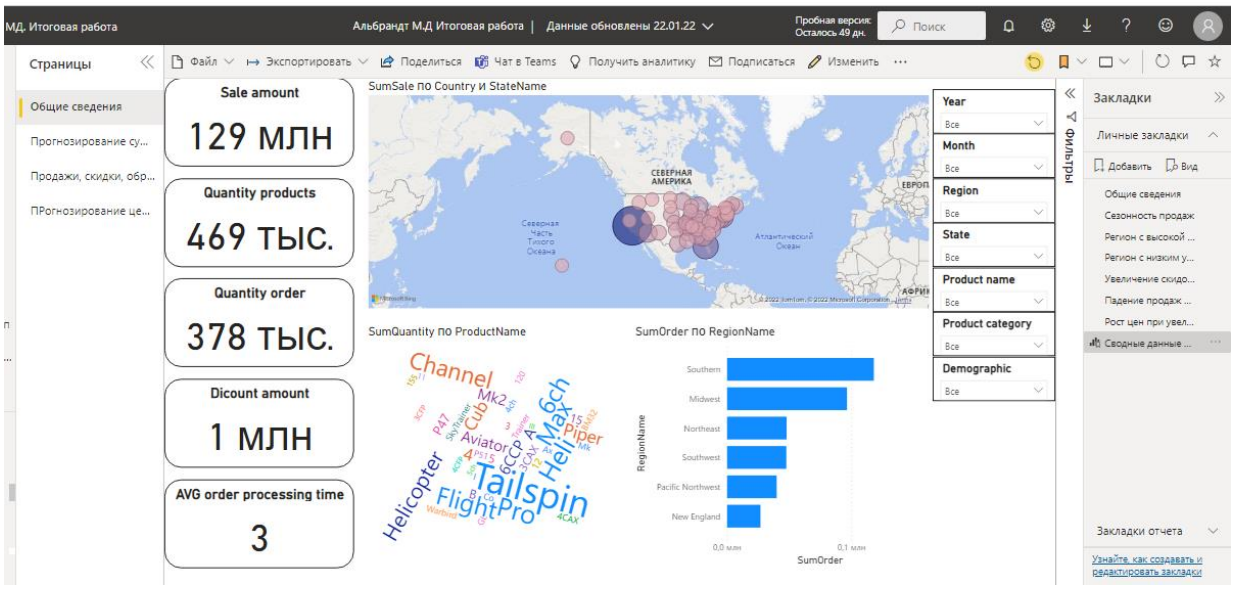
На платформе Power BI был создан отчет, содержащий сводную информацию о компании, данные по прогнозированию и сводную аналитику по продажам и

срокам обработки заказов, две панели мониторинга и две модели обучения, зафиксированы закладки с интересующей информацией.

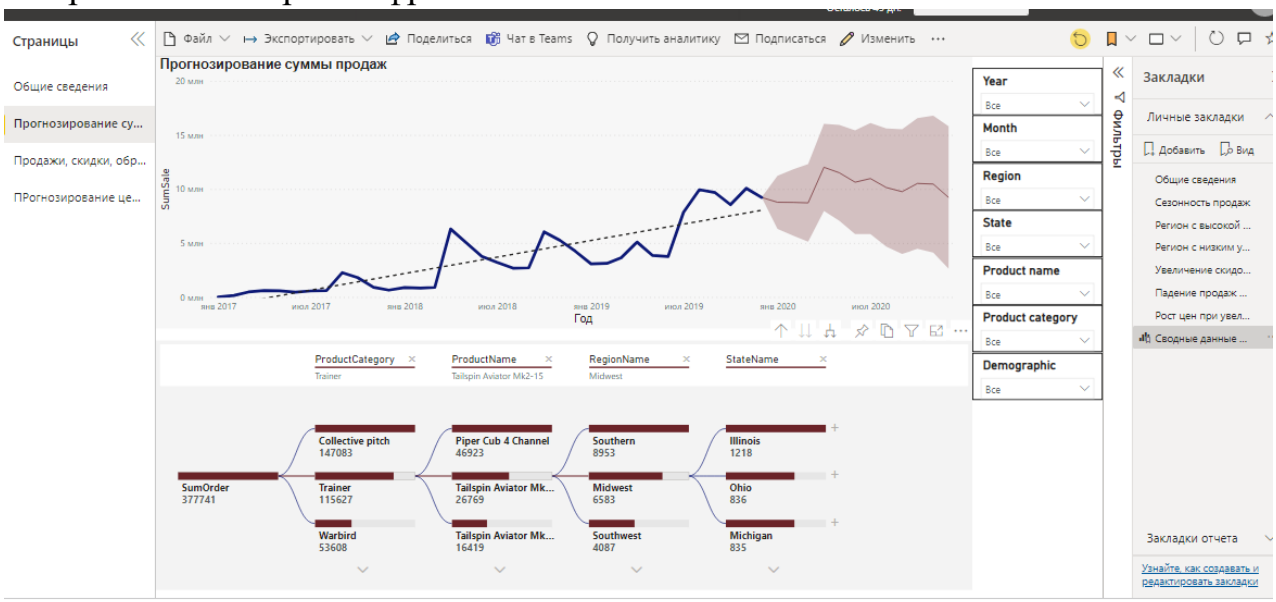


Отчет содержит следующие страницы:

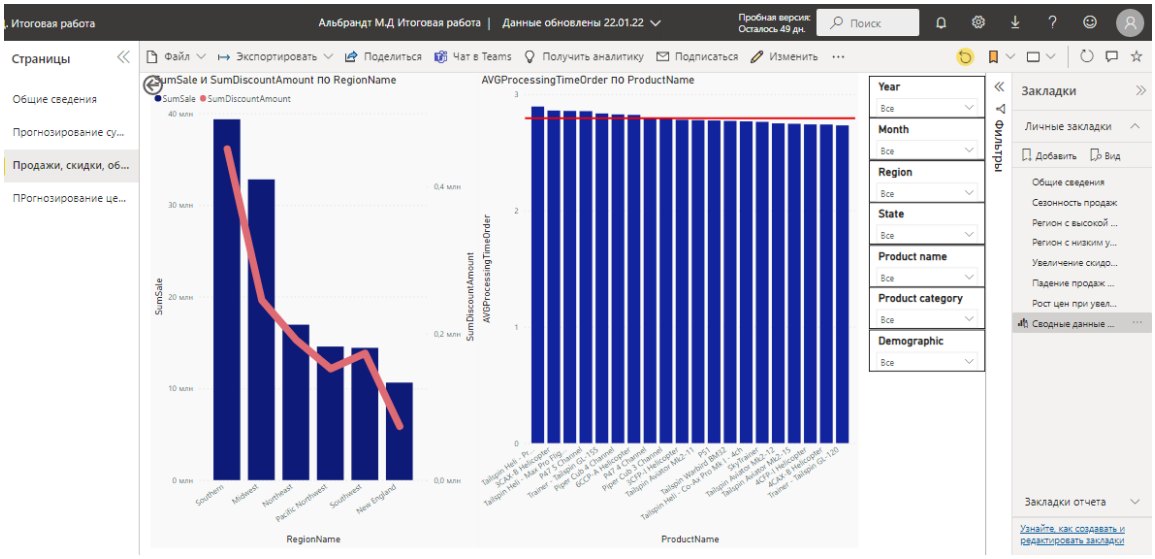
1. Общие сведения. Страница содержит основные показатели KPI: сумма продаж за минусом скидок, количество проданных товаров, количество заказов, общая сумма предоставленных скидок по промокодам, среднее время обработки заказа. Также отражает географию распределение заказов.



2. Прогнозирование суммы продаж. График отражает текущие данные о продажах, также показывает прогноз на ближайший год, Дерево декомпозиции позволяет увидеть расшифровку по категоризации товаров в плоть до наименования и региона продаж, с выстраиванием прогноза по каждой конкретной ветке расшифровки.

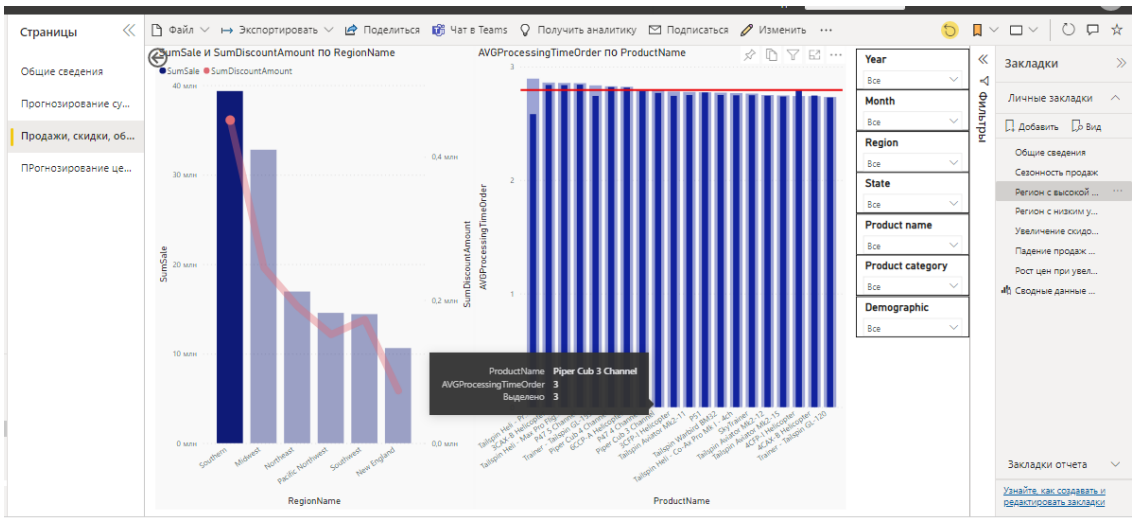


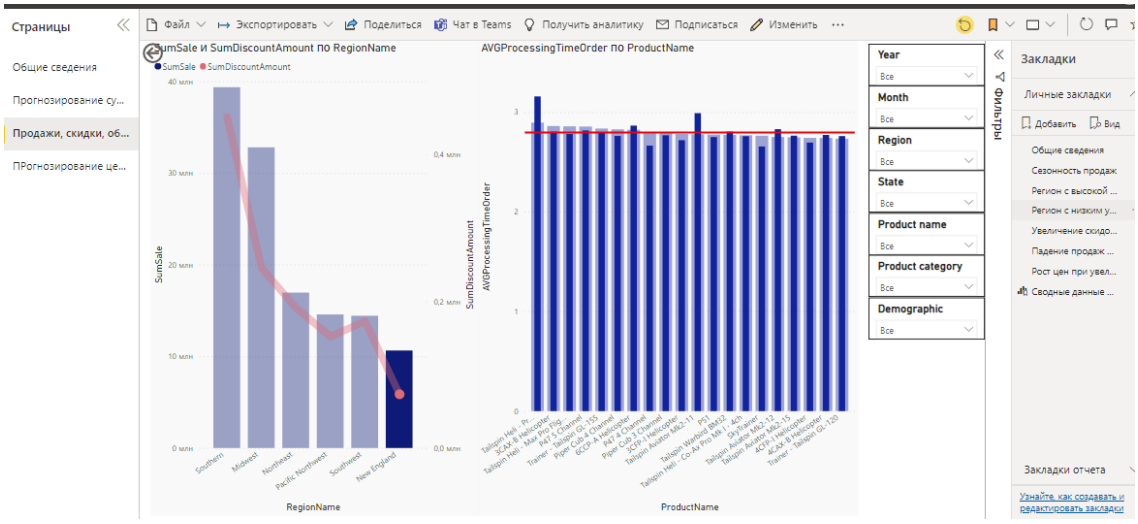
3. Продажи, скидки, обработка заказов. На странице отражены сводные данные о сумме продаж в сопоставлении с предоставленной сумой скидок, и среднем временем обработки заказов по каждой товарной позиции.



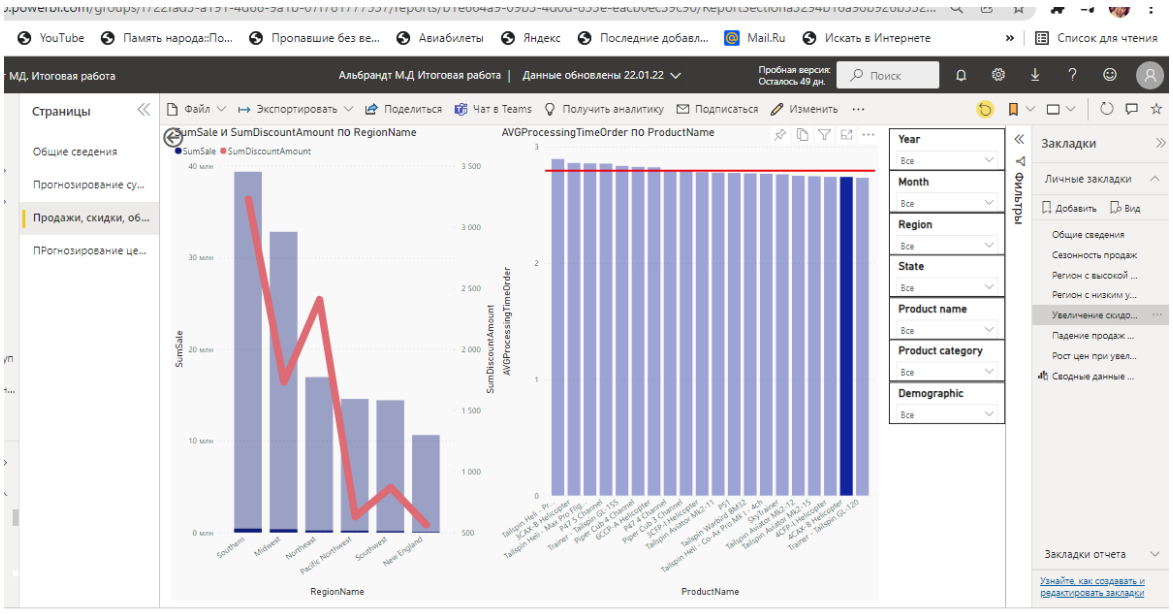
При работе с данной страницей выявлены следующие закономерности:

- В Регионе с высоким уровнем продаж средний срок обработки заказов по всем позициям стабильный, в регионе же с низким уровнем продаж замечены увеличения сроков обработки заказов по ряду позиций. Что позволяет установить связь суммы продаж по региону и вероятности увеличения срока обработки заказа.

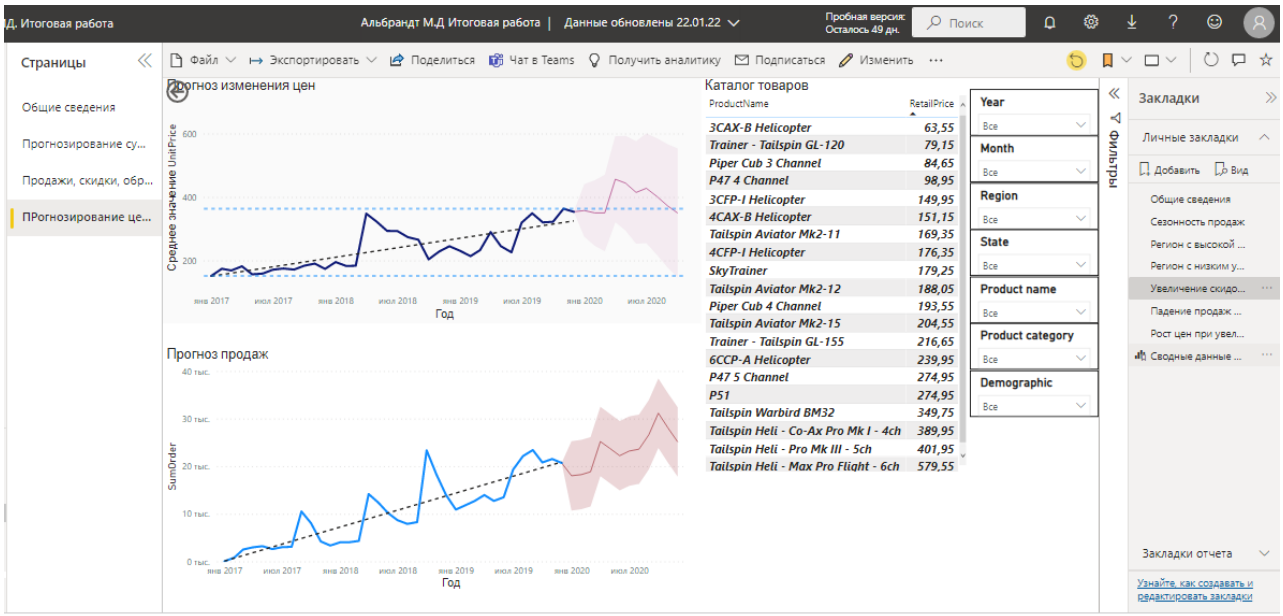




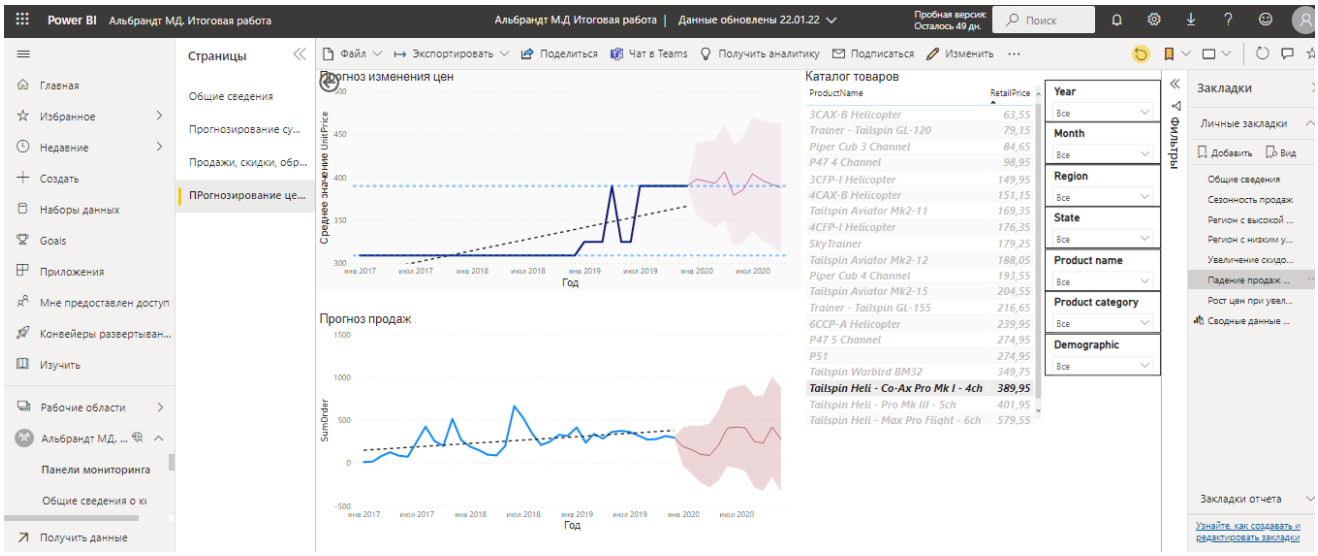
На некоторые позиции замечено увеличения объема скидок обратно пропорциональное объему продаж по ряду регионов, что также позволяет установить связь данных показателей.



4. Прогнозирование цен и объема продаж. Данная страница показывает динамику изменения объема продаж и изменения среднего уровня цен по магазину, также актуальный Прайс-лист и позволяет отследить динамику по каждой позиции прайса.



Можно отследить, для конкретной позиции товаров замечено падение продаж при росте цены.

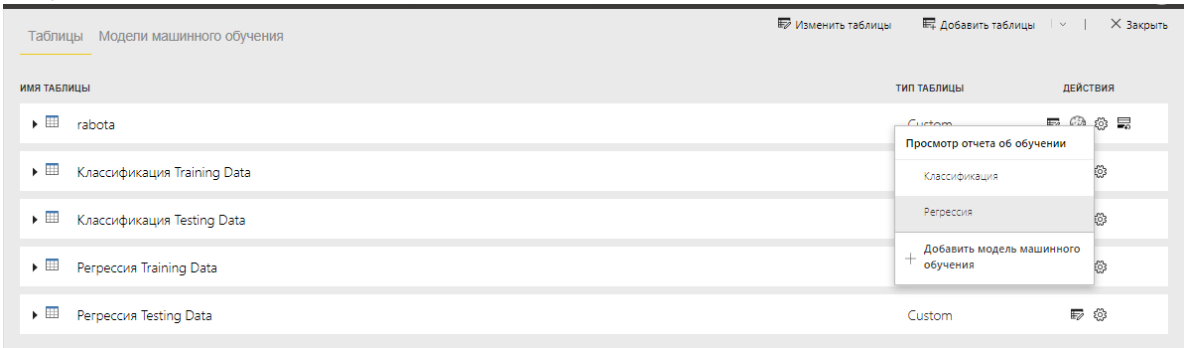


На основании отчета сформированы две панели мониторинга:

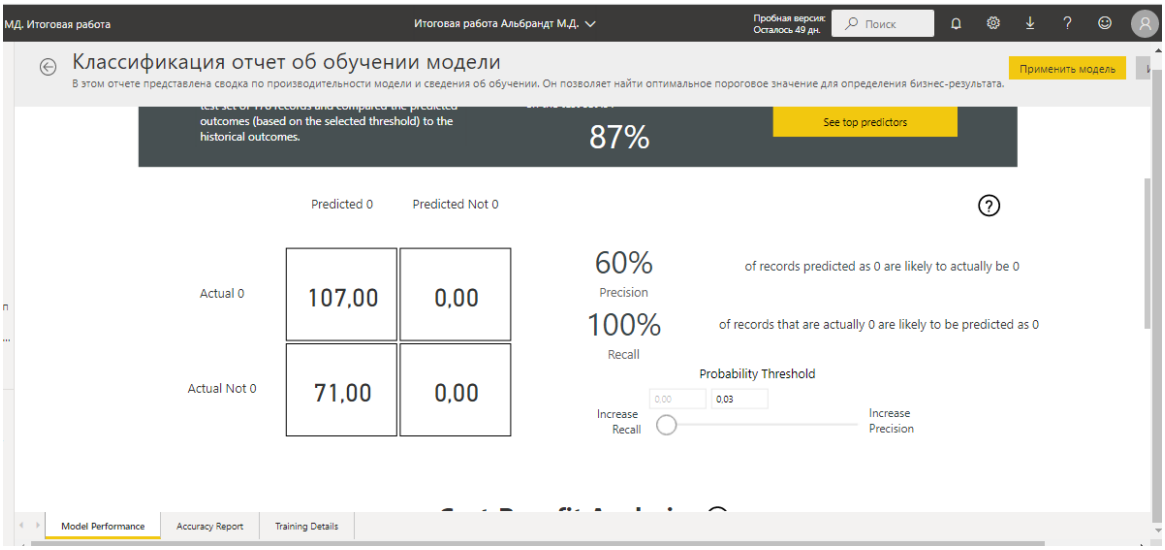
- Общие сведения о компании, содержащая 1 и 3 страницы;

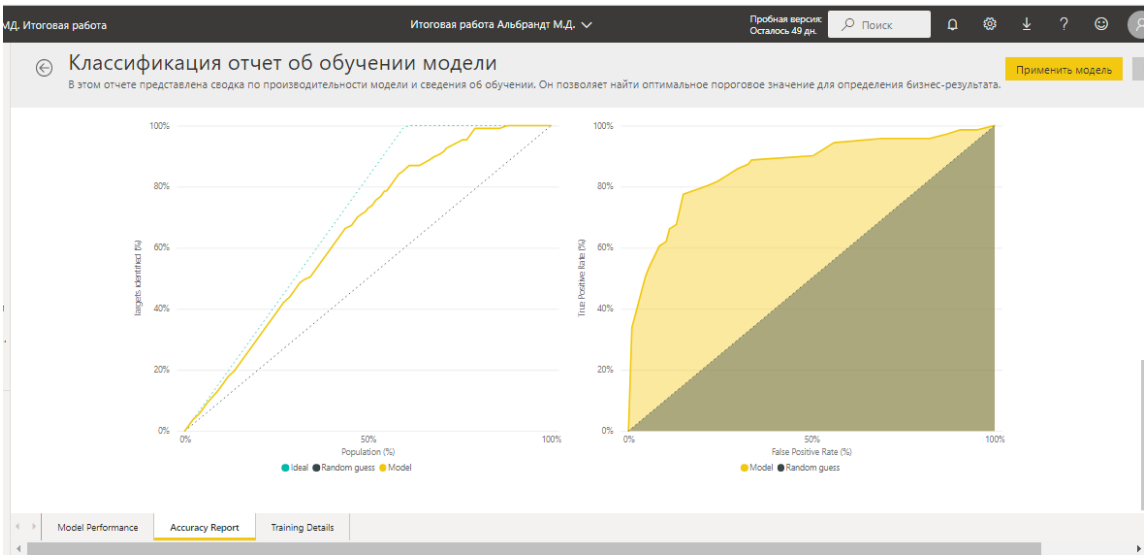
- Прогнозирование, содержащие 2 и 4 страницы.

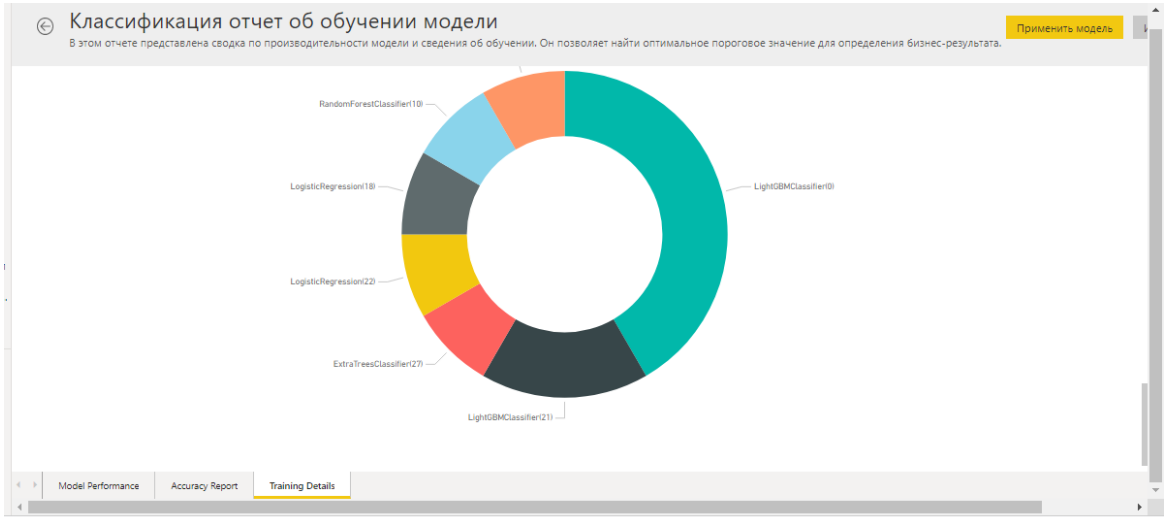
Обучены две модели:

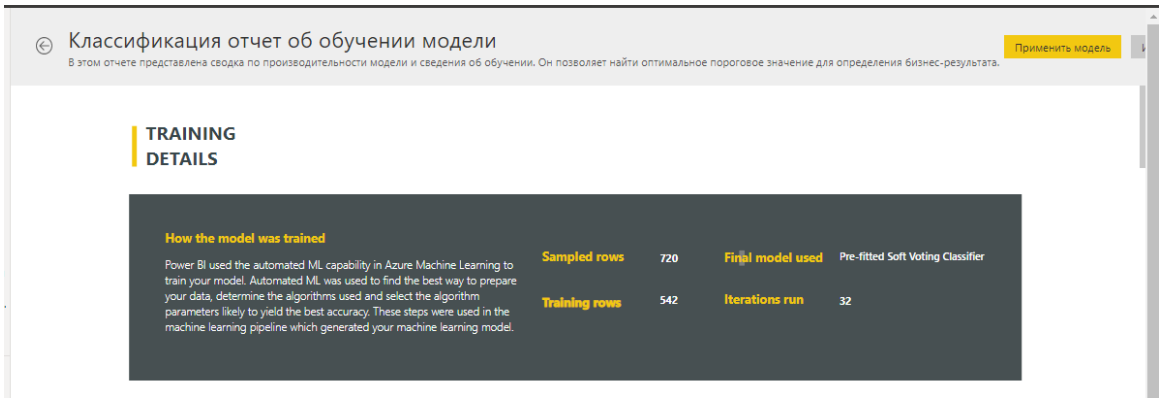


1. Классификация. Для обучения модели классификации был выбран классификационный показатель срока обработки заказа (просрочен заказ или нет).



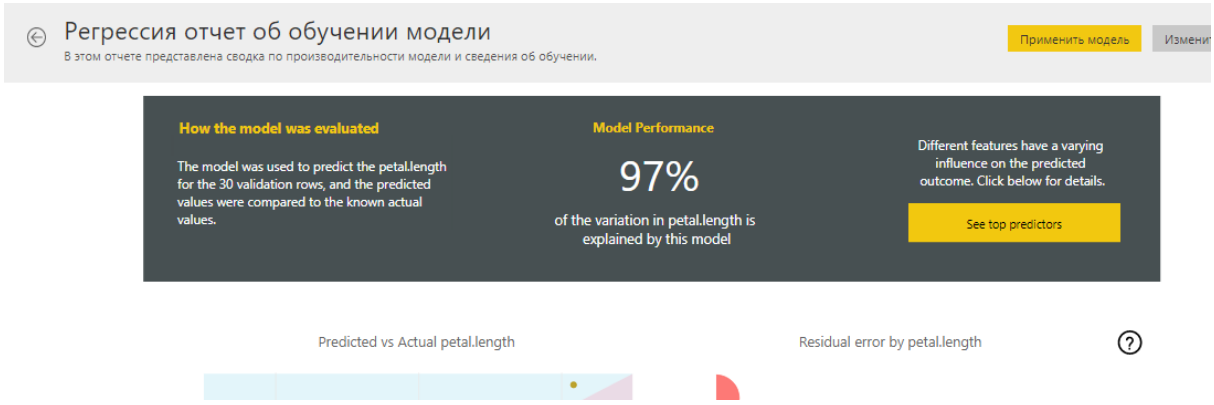


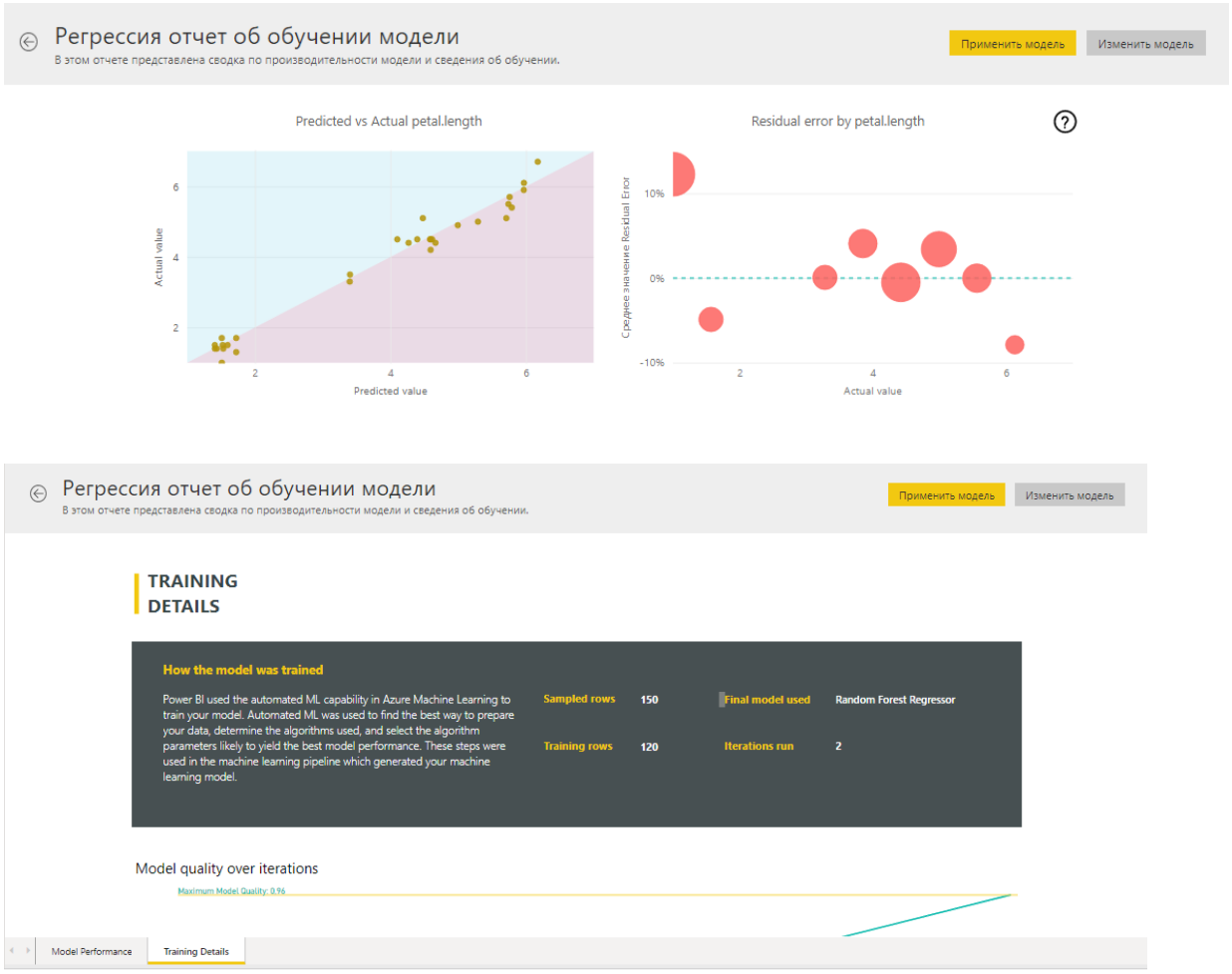




На текущий момент, модель, обученная в Power BI, имеет лучшие показатели из всех моделей, рассмотренных выше.

2. Регрессия. Для обучения модели регрессии была выбран показатель Сумма продаж.





По результатам метрик модель, обученная в Power BI, имеет лучшие показатели из всех моделей, рассмотренных выше.