СТАТИСТИЧЕСКОЕ ИССЛЕДОВАНИЕ ДАННЫХ

РАБОТА АНАЛИТИКА ДАННЫХ

СТАТИСТИЧЕСКИЙ АНАЛИЗ

Статистический анализ возник как область математической статистики.

Машинное обучение возникло как раздел искусственного интеллекта.

Обе области имеют общую направленность по решению задач обучения *"с учителем" и "без учителя".*

Машинное обучение уделяет больше внимания крупномасштабным приложениям и точности прогнозирования.

Статистический анализ направлен больше на интерпретацию моделей и оценку неопределенности.

Границы между этими видами аналитики становятся все более размытыми, что приводит к взаимопересечению и взаимобогащению.

Машинное обучение в настоящее время активно применяется в сфере маркетинга и интернет-аналитики.

ЗАДАЧА ОБУЧЕНИЯ «С УЧИТЕЛЕМ»

supervised learning

Постановка задачи

• Измерение результата ***Y*** (называемого *зависимая* переменная, реакция, цель);

• Вектор ***p*** предикторных измерений ***X*** (называемых **входные** данные, регрессоры, признаки, независимые переменные);

• В задаче *регрессии* значение ***Y*** является количественным (цена, давление крови, выбор кандидата и т.п.);

• В задаче *классификации* ***Y*** принимает значения в ограниченном неупорядоченном наборе (выжил/умер, цифры 0-9, состояние и т.п.);

• Есть обучающие данные *(x1; y1); . . . ; (xN; yN)* - это наблюдения (примеры, экземпляры) этих измерений;

• Есть тестовые данные *(x01; y01); . . . ; (xK; yK)* - это наблюдения (примеры) для проверки полученной модели;

ЗАДАЧА ОБУЧЕНИЯ «С УЧИТЕЛЕМ»

На основе этого обучения выполняется:

- Точное прогнозирование неизвестных случаев;
- Оценка, какие исходные данные влияют на результат и каким образом;
- Проверка качества прогнозов и выводов.

ЗАДАЧА ОБУЧЕНИЯ «С УЧИТЕЛЕМ»

Важно понимать идеи, стоящие за различными методами, чтобы знать, как и когда их использовать.

Сначала нужно пытаться применить более простые методы, прежде чем применять более сложные.

Важно точно оценить эффективность работы метода, чтобы узнать, насколько хорошо / или плохо он работает.

[простые методы часто работают так же хорошо, как и более сложные]

Это область исследований, имеющая важные применение в науке, социологии, промышленности, финансах.

Изучение статистики является фундаментальным компонентом процесса подготовки аналитика данных.

ЗАДАЧА ОБУЧЕНИЯ «БЕЗ УЧИТЕЛЯ»

Постановка задачи

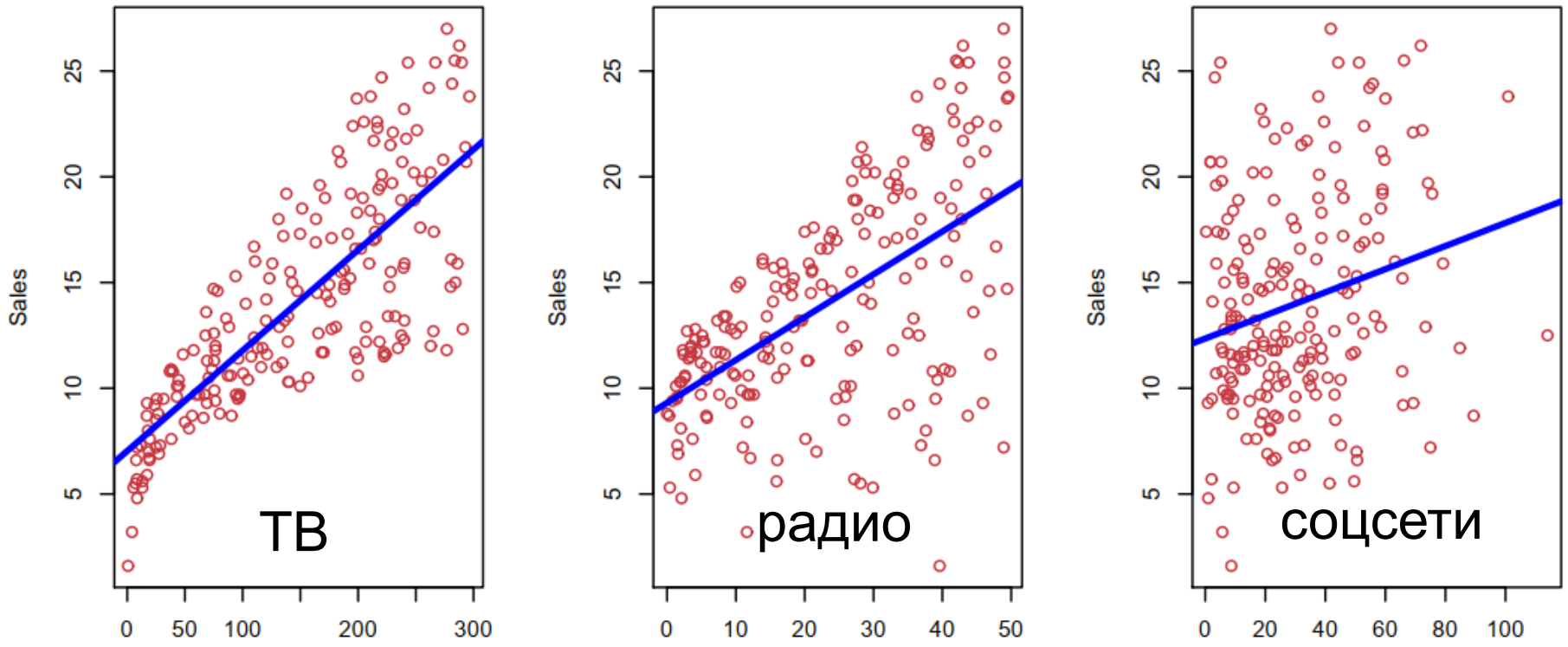
Нет результирующей переменной, только набор предикторов (признаков), измеряемых на наборе образцов (выборок) данных.

Цель более *размытая* - найти *группы выборок* данных, которые ведут себя аналогично; найти *функции*, которые ведут себя аналогично; найти линейные *комбинации* объектов с наименьшей вариабельностью.

- Но сложно понять, насколько хорошо получается найти правильную модель данных.

- Отличается от *обучения с учителем*, может быть полезен в качестве этапа предварительной обработки для последующего анализа.

ЗАДАЧА ОБУЧЕНИЯ «БЕЗ УЧИТЕЛЯ»



Показаны зависимость продаж от рекламы на телевидении, радио и в соцсети. Линии линейной регрессии подходят отдельно по каждому сектору.

Можем ли мы спрогнозировать тренд продаж, используя эти 3 фактора?

Можно ли добиться большей точности прогноза, используя модель *S(3x)?*

ЗАДАЧА ОБУЧЕНИЯ «БЕЗ УЧИТЕЛЯ»

В этой задаче продажи - это цель, которую хотим предсказать.

В общем случае обозначим ответ как ***Y***.

«ТВ» - это функция, входные данные, предиктор; назовем ее X1, аналогично, назовем «радио» как X2 и далее X3.

Назовем входной вектор в совокупности следующим образом:

$$X = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix}$$

Теперь запишем модель в виде $Y = f(X) + \epsilon$

где ε означает ошибки измерений и прочие несоответствия

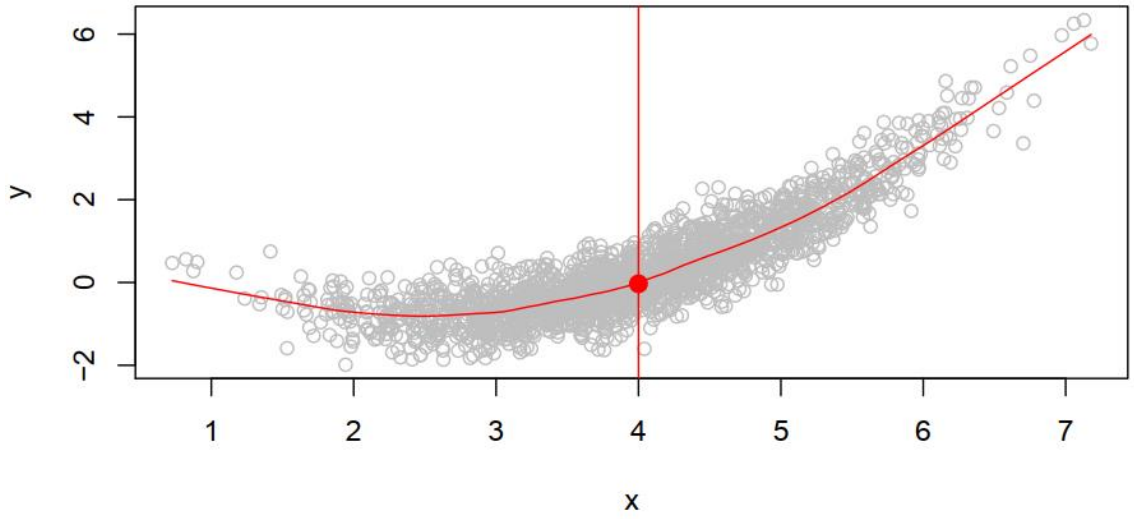
ЗАДАЧА ОБУЧЕНИЯ «БЕЗ УЧИТЕЛЯ»

При хорошем значении *f* можем предсказывать значение Y в новых точках $X = x_i$

Можно понять, какие компоненты X = (X1; X2;… Xn) важны для объяснения *Y*, и какие не имеют отношения к делу, например, трудовой стаж и образование оказывает большее влияние на доход, но семейное положение, как правило, не влияет.

В зависимости от сложности задачи, можно понять как каждый компонент *Xj* из *X* влияет на *Y*.

ЗАДАЧА ОБУЧЕНИЯ «БЕЗ УЧИТЕЛЯ»



Существует ли

идеальная *f(X)*?

Что является хорошим значением для f(X) при любом выбранном значении X, скажем, X = 4?

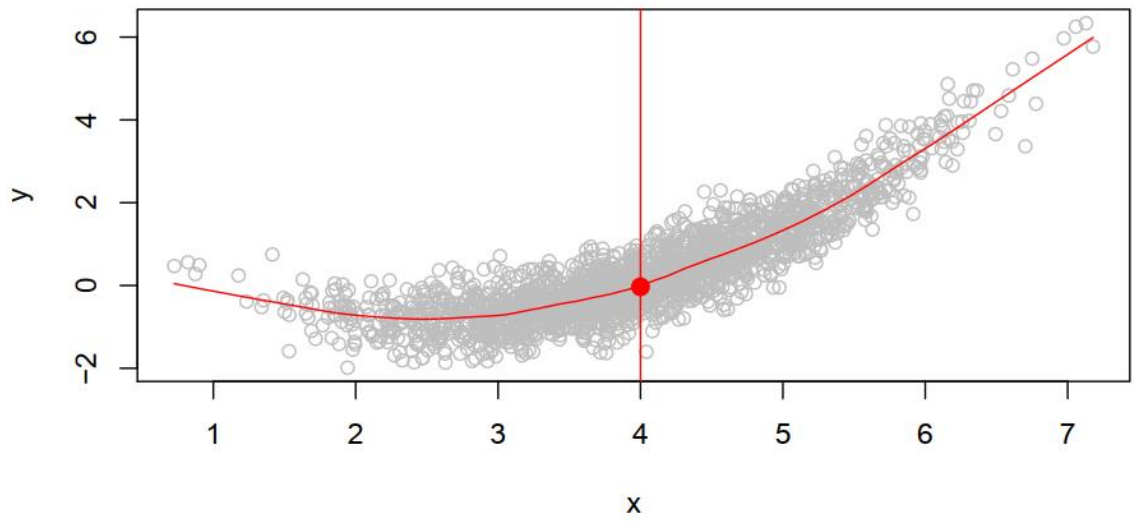
Как видно из графика, есть много значений Y при X = 4.

Хорошим значением является *f(4) = E(Y | X = 4)*

E(Y | X = 4) означает ожидаемое значение (среднее значение) *Y* (*estimate*) при заданном *X = 4*.

Этот идеал *f(x) = E(Y | X = x)* называется функцией **регрессии** (*regression function*).

ЗАДАЧА ОБУЧЕНИЯ “БЕЗ УЧИТЕЛЯ”



Для векторной функции выглядит так

$$f(x) = f(x_1, x_2, x_3) = E(Y|X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

Является ли оптимальным предиктором Y в отношении среднеквадратичной ошибки прогнозирования $f(x) = E(Y \mid X = x)$ - функция, которая минимизирует $E[(Y - g(X))^2 \mid X = x]$ по всей функции g во всех точках $(X = x)$

ЗАДАЧА ОБУЧЕНИЯ “БЕЗ УЧИТЕЛЯ”

Является ли ***Y*** оптимальным предиктором в отношении среднеквадратичной ошибки прогнозирования $f(x) = E(Y \mid X = x)$ - это функция, которая минимизирует $E[(Y - g(X))^2 \mid X = x]$ по всей функции g во всех точках (X = x)

$\varepsilon = Y - f(x)$ неустранимая ошибка, т.е. даже если бы мы знали $f(x)$, мы все равно допустили бы ошибки в прогнозировании, поскольку при каждом X = x обычно существует распределение возможных значений ***Y***.

Для любой оценки $\hat{f}(x)$ из $f(x)$ мы имеем

$$E[(Y - \hat{f}(X))^2 | X = x] = [f(x) - \hat{f}(x)]^2 + \quad \mathrm{Var}(\epsilon)$$

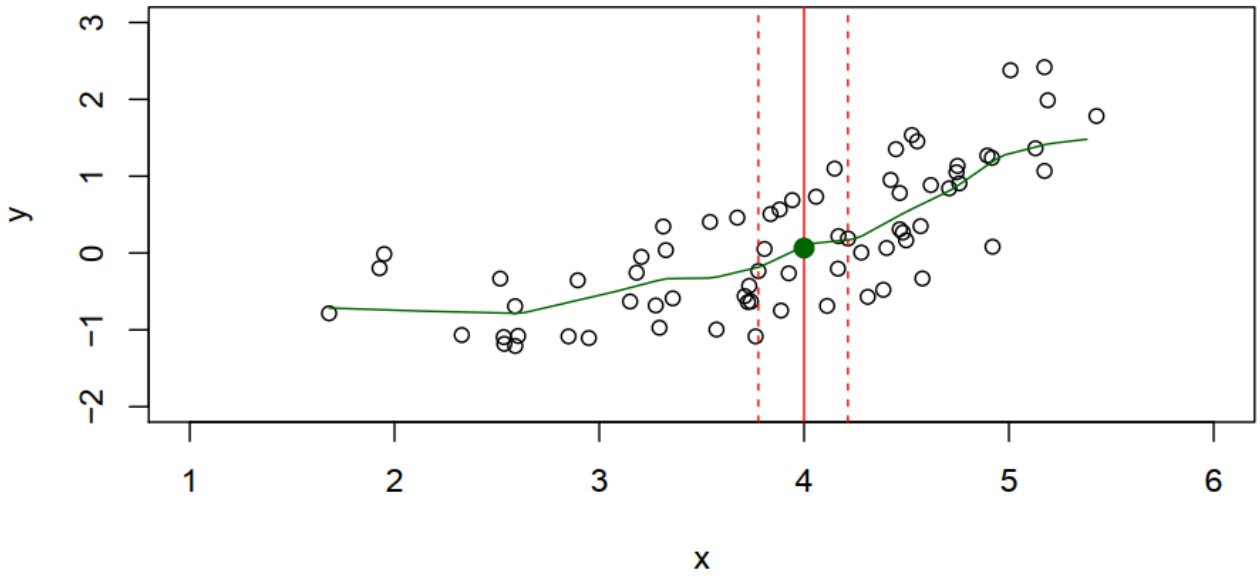
ЗАДАЧА ОБУЧЕНИЯ “БЕЗ УЧИТЕЛЯ”

Как правило, в наличии мало точек данных с $X = 4$,
если точно такие вообще имеются,
мы не всегда сможем вычислить $E(Y \mid X = x)$

Смягчим определение и позволим

$$\hat{f}(x) = \mathrm{Ave}(Y|X \in \mathcal{N}(x))$$

где $N(x)$ - некоторая окрестность (*neighborhood*) x



ВОЗМОЖЕН ЛИ ПРОГНОЗ ?

Перед тем как приступить к прогнозированию, было бы хорошо оценить принципиальную возможность построения качественной модели.
При прогнозировании пытаются обобщить прошлые события и, используя эти знания, предсказать, что будет завтра.
Но как быть при недостатке информации, например, если определенный товар продавался 1 раз в полгода, то сколько его будет продано завтра?
Фактически прогнозирование возможно только для товаров, которые продаются более или менее стабильно и невозможно для товаров, спрос на которые хаотичный.

НУЖЕН ЛИ ПРОГНОЗ ?

Известно, что в большинстве случаев работает правило Парето, согласно которому 80% дохода приносит 20% товаров.
Если товары продаются нерегулярно, возможно их вообще *не стоит* прогнозировать?
Товары отличаются по стабильности продаж и доходности. Прилагать усилия, чтобы получить качественный прогноз по всем товарным позициям, *нерационально*, а в некоторых случаях это сделать *невозможно*.
Усилия нужно дифференцировать: одни позиции прогнозировать, другие - планировать, чему-то уделять максимум внимания, а что-то рассчитывать по жестким правилам.

ВЫБОР ОБЪЕКТОВ ПРОГНОЗИРОВАНИЯ

Для того чтобы определиться с тем, на что направлять усилия при прогнозировании, нужно определиться с критериями.

Одним из простых и удобных способов является комбинирование ABC и XYZ-анализа:

ABC – разбиение на группы по доходности
XYZ – разбиение на группы по стабильности поведения

ABC И XYZ – КАК РАСПРЕДЕЛЯТЬ УСИЛИЯ

ОПТИМАЛЬНОЕ ПЛАНИРОВАНИЕ

Для оптимального планирования необходимо учитывать не точность предсказания, т.к. прогнозирования в данном случае нет вообще, но другие показатели: минимизация площадей, финансов, загрузки транспорта и т.п.

Например, для доходных, но нестабильно продаваемых товаров (AZ) иметь страховой запас, равный среднемесячным продажам, а для низкодоходных и нестабильно продаваемых (CZ) - минимально возможный страховой запас, например, 1 единицу.

ПРОГНОЗИРОВАНИЕ ПРОДАЖ

Для процессов, которые можно и нужно прогнозировать, рекомендуется построение не одной, а нескольких моделей прогнозирования.

Далее из этих моделей необходимо **выбирать лучшую** на основе некоторого критерия, например, та модель, что лучше всего объяснила продажи за последний месяц, считается лучшей и используется для прогнозирования.

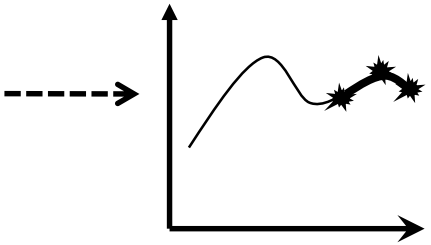
Сильной стороной подобного подхода является возможность автоматизировать прогноз методом машинного обучения и выбрать лучшую прогностическую модель.

Подобный подход позволяет автоматически обрабатывать огромные массивы данных.

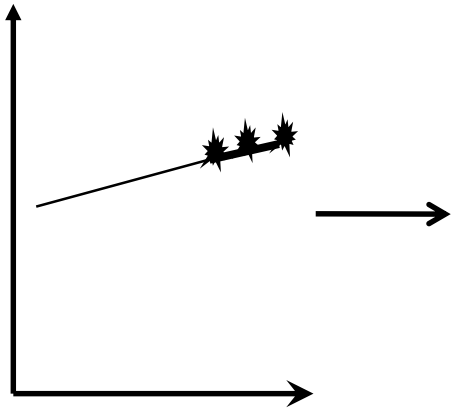
РАСЧЕТ ПОТРЕБНОСТИ

В целом должна выполняться цепочка Прогнозирование – Планирование – Расчет потребности.
Для прогнозируемых позиций тоже рекомендуется выполнять этап планирования.

Прогнозирование



Планирование



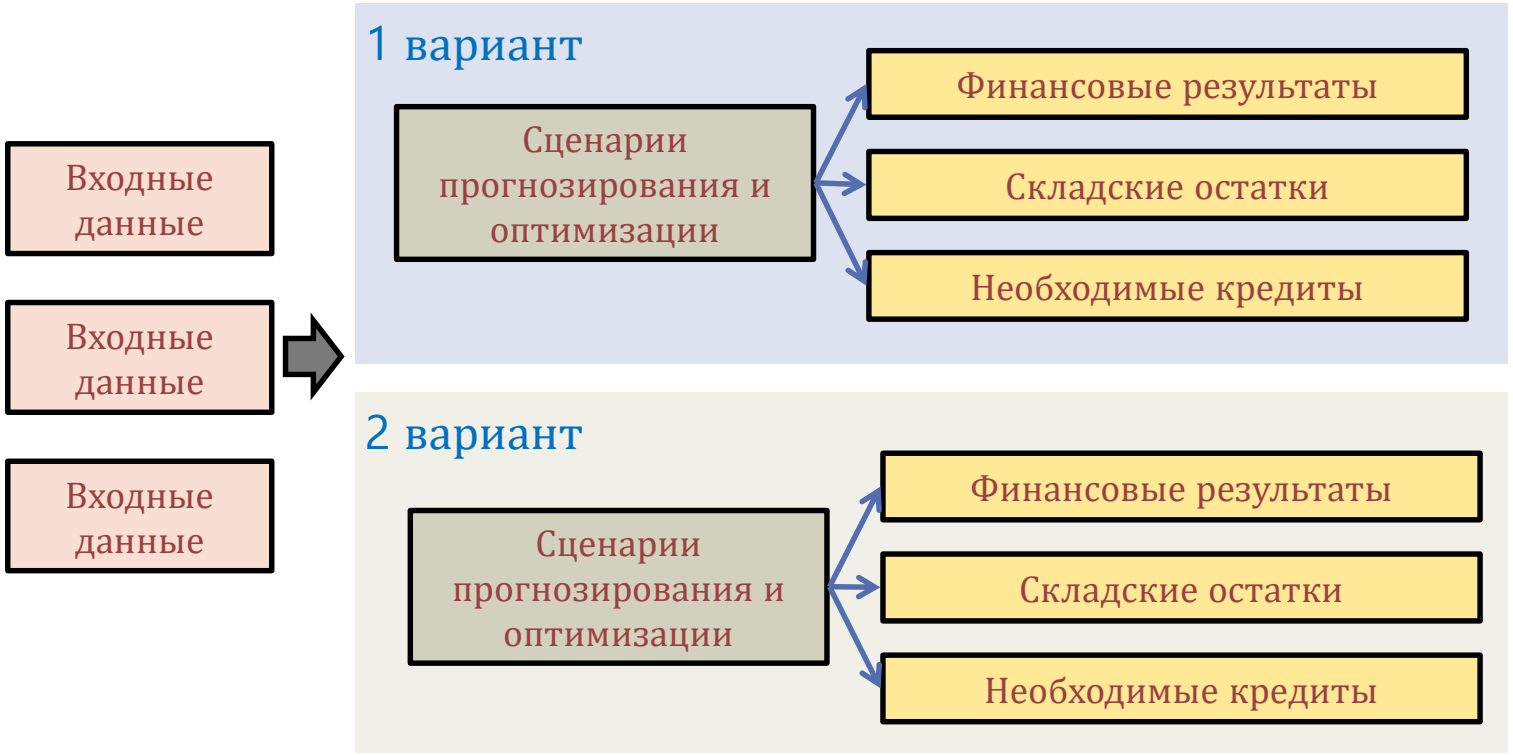
Потребность

Товар	Остаток	Потребность
		10
		47
		175

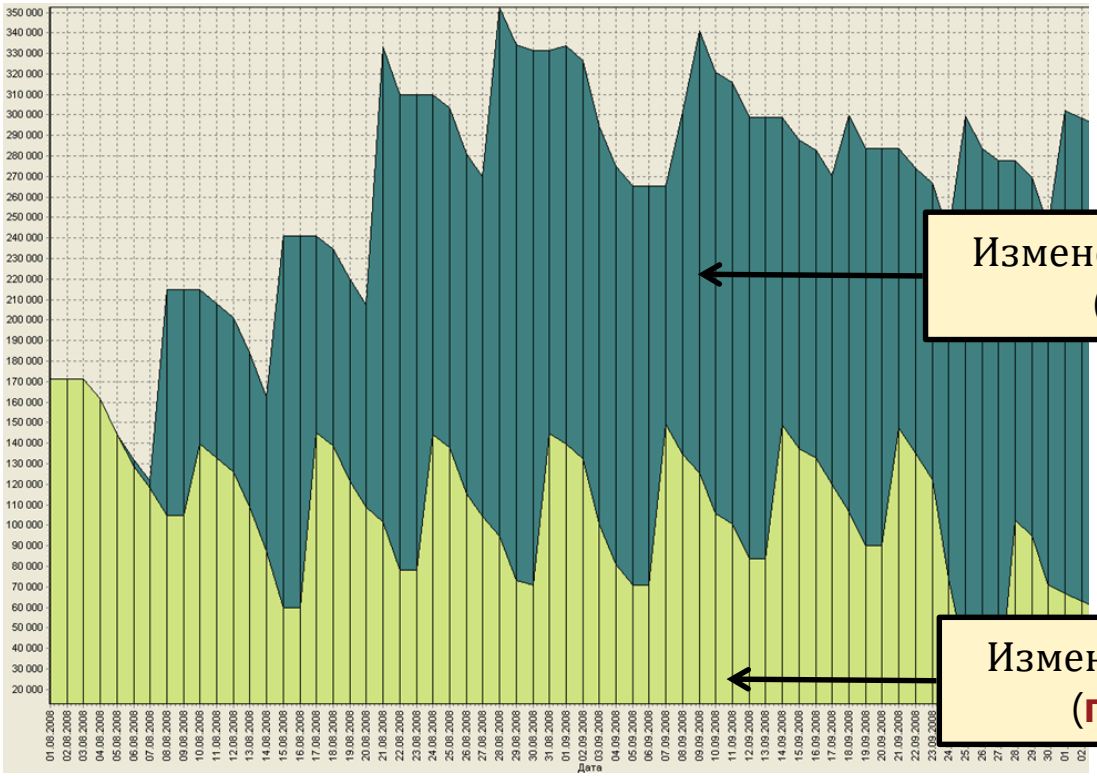
Получение прогнозируемых величин не является конечным этапом в работе определения потребности и величины разовой поставки. Далее следует трансформация расчетных величин в план закупок, учитывающий ограничения и дополнительную информацию.

АНАЛИЗ «ЧТО-ЕСЛИ»

После того как построены сценарии планирования, прогнозирования и оптимизации, появляется возможность анализировать по принципу «**что-если**», т.е. «обыгрывать» различные варианты будущего.



ОЦЕНКА ЭФФЕКТИВНОСТИ ОПТИМИЗАЦИИ



Изменение складских остатков (**до оптимизации**)

Изменение складских остатков (**после оптимизации**)

Разница в складских остатках при использовании оптимизационных методов и без них – это и есть неэффективно используемые средства.

Оценить можно, «прогоняя» модели на исторических данных.

Краткое введение в pandas

Базовые структуры данных в pandas

Два новых типа классов для обработки данных

Series

одномерный маркированный массив, содержащий данные любого типа такие как целые числа, строки, объекты Python и т.д.

DataFrame

двумерная структура данных, которая хранит данные в виде двумерного массива или таблицы со строками и столбцами

https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html

краткое введение в pandas

Создание Series из списка значений, позволяющее создать значение по умолчанию RangeIndex

```
s = pd.Series([1, 3, 5, np.nan, 6, 8])
```

```
0  1.0
1  3.0
2  5.0
3  NaN
4  6.0
5  8.0
dtype: float64
```

краткое введение в pandas

Создание DataFrame путём передачи массива NumPy с индексом datetime с использованием date_range() и помеченных столбцов

```
dates = pd.date_range("20250101", periods=6)
```

```
DatetimeIndex(['2025-01-01', '2025-01-02',
'2025-01-03', '2025-01-04', '2025-01-05',
'2025-01-06'], dtype='datetime64[ns]',freq='D')
```

```
df = pd.DataFrame(np.random.randn(6, 4),
index=dates, columns=list("ABCD"))
```

краткое введение в pandas

```
df
```

	A	B	C	D
2025-01-01	0.469112	-0.282863	-1.509059	-1.135632
2025-01-02	1.212112	-0.173215	0.119209	-1.044236
2025-01-03	-0.861849	-2.104569	-0.494929	1.071804
2025-01-04	0.721555	-0.706771	-1.039575	0.271860
2025-01-05	-0.424972	0.567020	0.276232	-1.087401
2025-01-06	-0.673690	0.113648	-1.478427	0.524988

краткое введение в pandas

Создание DataFrame путем передачи словаря объектов, где ключами являются заголовки столбцов, а значениями — значения столбцов.

```
df2 = pd.DataFrame(
  {
   "A": 1.0,
   "B": pd.Timestamp("20250102"),
   "C": pd.Series(1,index=list(range(4)),dtype="float32"),
   "D": np.array([3] * 4, dtype="int32"),
   "E": pd.Categorical(["test","train","test","train"]),
   "F": "foo",
  }
)
```

краткое введение в pandas

```
df2
```

```
     A          B         C   D      E    F
0  1.0 2025-01-02  1.0  3   test  foo
1  1.0 2025-01-02  1.0  3  train  foo
2  1.0 2025-01-02  1.0  3   test  foo
3   1.0 2025-01-02  1.0  3  train  foo
```

```
df2.dtypes
```

```
A           float64
B    datetime64[s]
C           float32
D             int32
E          category
F            object
dtype: object
```

краткое введение в pandas

Для просмотра верхних и нижних строк фрейма используйте DataFrame.head() и DataFrame.tail()

Для отображения DataFrame.index *или* DataFrame.columns

```
df.index
DatetimeIndex(['2025-01-01', '2025-01-02', '2025-01-03',
'2025-01-04', '2025-01-05', '2025-01-06'],
dtype='datetime64[ns]', freq='D')
```

```
df.columns
Index(['A', 'B', 'C', 'D'], dtype='object')
```

краткое введение в pandas

Транспонирование данных

```
df.T
```

```
   2025-01-01 2025-01-02 2025-01-03 2025-01-04 2025-01-05 2025-01-06
A   0.469112   1.212112  -0.861849   0.721555  -0.424972  -0.673690
B  -0.282863  -0.173215  -2.104569  -0.706771   0.567020   0.113648
C  -1.509059   0.119209  -0.494929  -1.039575   0.276232  -1.478427
D  -1.135632  -1.044236   1.071804   0.271860  -1.087401   0.524988
```

краткое введение в pandas

DataFrame.sort_index() сортирует данные по оси

```
df.sort_index(axis=1, ascending=False)
```

```
                   D         C         B         A
2025-01-01 -1.135632 -1.509059 -0.282863  0.469112
2025-01-02 -1.044236  0.119209 -0.173215  1.212112
2025-01-03  1.071804 -0.494929 -2.104569 -0.861849
2025-01-04  0.271860 -1.039575 -0.706771  0.721555
2025-01-05 -1.087401  0.276232  0.567020 -0.424972
2025-01-06  0.524988 -1.478427  0.113648 -0.673690
```

краткое введение в pandas

DataFrame.sort_values() сортирует по значению

```
df.sort_values(by="B")
```

```
                   A         B         C         D
2025-01-03 -0.861849 -2.104569 -0.494929  1.071804
2025-01-04  0.721555 -0.706771 -1.039575  0.271860
2025-01-01  0.469112 -0.282863 -1.509059 -1.135632
2025-01-02  1.212112 -0.173215  0.119209 -1.044236
2025-01-06 -0.673690  0.113648 -1.478427  0.524988
2025-01-05 -0.424972  0.567020  0.276232 -1.087401
```

краткое введение в pandas

Булева индексация

Выбрать строки, где df.A больше *0*

```
df[df["A"] > 0]
```

```
                   A         B         C         D
2025-01-01  0.469112 -0.282863 -1.509059 -1.135632
2025-01-02  1.212112 -0.173215  0.119209 -1.044236
2025-01-04  0.721555 -0.706771 -1.039575  0.271860
```

краткое введение в pandas

Булева индексация

Выбор значений из DataFrame, где выполняется логическое условие

```
df[df > 0]
```

	A	B	C	D
2025-01-01	0.469112	NaN	NaN	NaN
2025-01-02	1.212112	NaN	0.119209	NaN
2025-01-03	NaN	NaN	NaN	1.071804
2025-01-04	0.721555	NaN	NaN	0.271860
2025-01-05	NaN	0.567020	0.276232	NaN
2025-01-06	NaN	0.113648	NaN	0.524988

краткое введение в pandas

Булева индексация

Использование метода фильтрации isin()

```
df2 = df.copy()

df2["E"] = ["one", "one", "two", "three", "four", "three"]


df2[df2["E"].isin(["two", "four"])]
```

```
                   A         B         C         D      E
2025-01-03 -0.861849 -2.104569 -0.494929  1.071804    two
2025-01-05 -0.424972  0.567020  0.276232 -1.087401   four
```

краткое введение в pandas

DataFrame.dropna() удаляет все строки, в которых отсутствуют данные

```
df1.dropna(how="any")
```

```
                   A         B         C    D    F    E
2025-01-02  1.212112 -0.173215  0.119209  5.0  1.0  1.0
```

DataFrame.fillna()заполняет недостающие данные

```
df1.fillna(value=5)
```

```
                   A         B         C    D    F    E
2025-01-01  0.000000  0.000000 -1.509059  5.0  5.0  1.0
2025-01-02  1.212112 -0.173215  0.119209  5.0  1.0  1.0
2025-01-03 -0.861849 -2.104569 -0.494929  5.0  2.0  5.0
2025-01-04  0.721555 -0.706771 -1.039575  5.0  3.0  5.0
```

краткое введение в pandas

Рассчитать среднее значение для каждого столбца

```
df.mean()
A   -0.004474
B   -0.383981
C   -0.687758
D    5.000000
F    3.000000
dtype: float64
```

Рассчитать среднее значение для каждой строки

```
df.mean(axis=1)
2025-01-01    0.872735
2025-01-02    1.431621
2025-01-03    0.707731
2025-01-04    1.395042
2025-01-05    1.883656
2025-01-06    1.592306
Freq: D, dtype: float64
```

краткое введение в pandas

DataFrame.agg() и DataFrame.transform() использует определяемую пользователем функцию, применяемую к ячейкам для агрегации или преобразования.

```
df.agg(lambda x: np.mean(x) * 5.6)
```

```
A    -0.025054
B    -2.150294
C    -3.851445
D    28.000000
F    16.800000
dtype: float64
```

```
df.transform(lambda x: x * 101.2)
```

	A	B	C	D	F
2025-01-01	0.000000	0.000000	-152.716721	506.0	NaN
2025-01-02	122.665737	-17.529322	12.063922	506.0	101.2
2025-01-03	-87.219115	-212.982405	-50.086843	506.0	202.4
2025-01-04	73.021382	-71.525239	-105.204988	506.0	303.6
2025-01-05	-43.007200	57.382459	27.954680	506.0	404.8
2025-01-06	-68.177398	11.501219	-149.616767	506.0	506.0