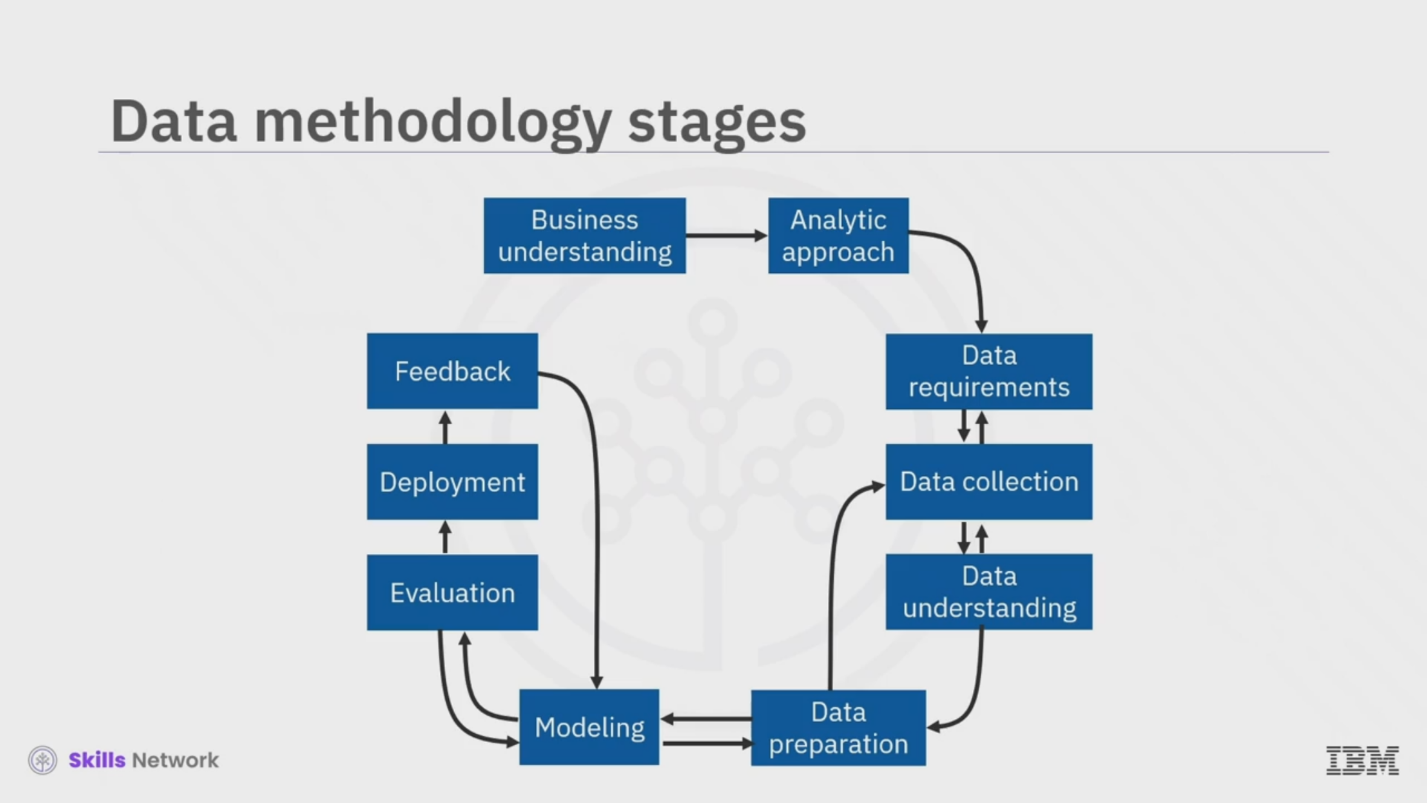
**1. Coursera.**

**Итеративная методология работы с данными**, охватывающая полный цикл от понимания бизнес-проблемы до получения обратной связи



**1. Business understanding**

**Понимание бизнеса**  
На этом этапе определяют **цель проекта** с точки зрения бизнеса. Какие задачи нужно решить? Какие метрики важны? Кто конечный пользователь решения?

💬 Пример: Нужно спрогнозировать отток клиентов, чтобы вовремя отправлять им удерживающие предложения.

**2. Analytic approach**

**Аналитический подход**  
Выбирается подход к решению задачи: классификация, регрессия, кластеризация и т.д. Определяется тип модели, нужные алгоритмы и способы анализа.

💬 Пример: Используем классификацию, чтобы определить, уйдёт клиент или нет.

**3. Data requirements**

**Требования к данным**  
На основе аналитического подхода определяется, какие данные нужны, в каком объёме, из каких источников, и какие признаки необходимо иметь.

💬 Пример: Нужны данные по покупкам, взаимодействию с рассылками, обращению в поддержку.

**4. Data collection**

**Сбор данных**  
Процесс извлечения данных из различных источников: базы данных, API, CSV-файлы и т.п.

💬 Пример: Загружаем историю покупок клиентов за последний год из CRM.

**5. Data understanding**

**Понимание данных**  
Знакомство с данными: анализ форматов, пропусков, выбросов, распределений, зависимостей. На этом этапе часто строят визуализации.

💬 Пример: Обнаружили, что часть клиентов имеет пропущенные значения по возрасту и доходу.

**6. Data preparation**

**Подготовка данных**  
Очистка, обработка, создание новых признаков, масштабирование, кодирование категориальных признаков и т.п.

💬 Пример: Заполнили пропуски медианой, категориальные признаки закодировали one-hot encoding.

**7. Modeling**

**Построение модели**  
Обучение моделей машинного обучения, подбор гиперпараметров, кросс-валидация. Сравнение различных моделей.

💬 Пример: Обучили логистическую регрессию и случайный лес, логрег даёт лучший AUC.

**8. Evaluation**

**Оценка**  
Оценка качества модели по метрикам, таким как ROC AUC, accuracy, precision, recall. Проверка, достигаются ли бизнес-цели.

💬 Пример: Модель предсказывает отток с точностью 85%, но recall только 60% — можно улучшить.

**9. Deployment**

**Развёртывание**  
Модель внедряется в продуктив: через API, веб-приложение, дашборд или batch-обработку. Обеспечивается стабильная работа.

💬 Пример: Сделали FastAPI сервис, который каждую ночь обрабатывает новые данные.

**10. Feedback**

**Обратная связь**  
Получение фидбэка от пользователей и мониторинг работы модели: насколько она помогает, не деградирует ли со временем.

💬 Пример: Через 2 месяца выяснилось, что модель хуже работает для новых пользователей — возвращаемся к подготовке данных или дообучению.

Этот процесс **итеративный**: фидбэк может вернуть нас на этап подготовки данных, переоценки бизнес-задачи или доработки модели.

На основе твоего проекта по предсказанию вероятности покупки пользователем в течение 90 дней я посмотрим, как реализуются **этапы из методологии**. Разобьём по шагам с короткими примерами кода.

**1. Business understanding**

**Цель:** определить, купит ли пользователь товар в течение 90 дней.

python

КопироватьРедактировать

# Бизнес-цель: увеличить продажи за счёт персонализированных рекомендаций и вовремя таргетировать тех, кто с высокой вероятностью купит.

**2. Analytic approach**

**Задача:** бинарная классификация (purchase\_within\_90\_days: 0 или 1).

python

КопироватьРедактировать

# Используем LogisticRegression, RandomForest или XGBoost

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

**3. Data requirements**

**Признаки, которые могут повлиять на покупку:**

* кол-во сообщений
* взаимодействие с ними
* история покупок
* данные о пользователе (возраст, сегмент и т.д.)

python

КопироватьРедактировать

features = [

'message\_count', 'send', 'open', 'click',

'purchase', 'unsubscribe', 'complain',

'days\_last\_purchase', 'quantity\_sum', 'price\_mean'

]

**4. Data collection**

**Сбор данных** — например, из CSV или БД.

python

КопироватьРедактировать

import pandas as pd

df = pd.read\_csv('user\_events.csv')

**5. Data understanding**

python

КопироватьРедактировать

print(df.info())

print(df.describe())

print(df['purchase'].value\_counts())

Анализ пропусков:

python

КопироватьРедактировать

df.isna().sum()

**6. Data preparation**

python

КопироватьРедактировать

df[features] = df[features].fillna(0) # Заполняем пропуски

# Масштабирование

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_scaled = scaler.fit\_transform(df[features])

**7. Modeling**

python

КопироватьРедактировать

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, df['target'], test\_size=0.2, random\_state=42)

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

**8. Evaluation**

python

КопироватьРедактировать

from sklearn.metrics import roc\_auc\_score, classification\_report

y\_pred = model.predict(X\_test)

y\_proba = model.predict\_proba(X\_test)[:,1]

print("ROC AUC:", roc\_auc\_score(y\_test, y\_proba))

print(classification\_report(y\_test, y\_pred))

**9. Deployment**

Оборачиваем в FastAPI:

python

КопироватьРедактировать

from fastapi import FastAPI

import joblib

app = FastAPI()

model = joblib.load("model.pkl")

@app.post("/predict")

def predict(features: dict):

X = pd.DataFrame([features])

X\_scaled = scaler.transform(X)

prediction = model.predict\_proba(X\_scaled)[0][1]

return {"probability\_of\_purchase": prediction}

**10. Feedback**

Собираем метрики и логируем данные:

python

КопироватьРедактировать

# Предикт лог

import datetime

with open("predictions\_log.csv", "a") as f:

f.write(f"{datetime.datetime.now()},{prediction}\n")

Плюс добавление retraining pipeline на новых данных — можно автоматизировать.

**2. Яндекс Практикум**

Классический процесс анализа данных. Вот его этапы:

**Постановка задачи**

Уточнение задачи и формирование образа конечного результата.

**Получение данных**

Данные всегда где-нибудь хранятся: в базе, файле, на сайте, в онлайн-таблице. Если вы знаете, где взять данные для проекта, на этом этапе важно получить их в удобном для анализа формате. Если же вам предстоит работа с новыми источниками данных, обратитесь к разработчикам или менеджерам — они подскажут, где лежат данные и как их получить.

**Предобработка данных**

На практике данные часто содержат пропуски, дубликаты, ошибки. Например, при заполнении формы регистрации на сайте кто-нибудь обязательно забудет написать фамилию, а цена товара вместо ожидаемых рублей может быть выражена копейками. Ваша задача во время предобработки — найти и устранить такие проблемы. Обнаружить абсолютно все оплошности вам вряд ли удастся, но результатом этого этапа будут готовые к дальнейшему анализу данные.

**Исследовательский анализ данных**

Исследовательский анализ данных, или EDA, предполагает построение графиков по каждому столбцу в наборе данных. Такая визуализация позволяет увидеть некоторые особенности, закономерности и аномалии. Например, график может показать, что определённый курьер в компании доставляет заказы неправдоподобно быстро. Сделанные наблюдения помогут вам разобраться в результатах дальнейшего анализа.

**Подготовка прототипа решения**

На этом этапе вы готовите первый рабочий вариант решения, или прототип. Работа над аналитической задачей — цикличный процесс. Иногда прототип может стать её финальным решением, но обычно хороший результат требует неоднократных доработок.

**Проверка результатов**

Решение подготовлено — пора изучить результаты. Это очень ответственный момент: вы определяете, решена ли задача, или нужно ещё что-нибудь доделать. Для этого соотнесите результаты с поставленной задачей, образом конечного результата и оцените правдоподобность самих результатов. Оценивая результат, руководствуйтесь здравым смыслом. Могут ли быть такие значения в этом контексте? Почему значения именно такие? Итог этого этапа — завершение проекта или выявленные недоработки.

**Доработка решения**

Когда во время проверки результатов в прототипе обнаруживаются несовершенства, нужно перейти к доработке. Этот этап замыкает цикл из трёх основных шагов решения аналитической задачи: подготовка прототипа решения → проверка результатов → внесение изменений. Цикл повторяется до тех пор, пока результат вас не удовлетворит.

**Оформление результатов**

Задача решена, но заказчик об этом пока не знает. Вы можете сделать выгрузку данных или подготовить отчёт с визуализацией результатов, выводами и рекомендациями. Лучше всего спросить у заказчика, в каком формате предоставить конечный результат. Дополнительно уточните, как заказчик планирует применить полученное решение: возможно, вы предложите более подходящий вариант представления результата.

**Проверка реальностью**

Наиболее интересный, но и самый сложный этап. Результаты работы специалиста по анализу данных обычно трудно сразу же оценить. Например, вы изучили информацию о пользователях приложения и представили свои рекомендации по повышению продаж. После этого заказчик исследования, обычно менеджер из команды развития бизнеса, решает — принять ваши предложения или нет. Если ответ — да, через некоторое время можно будет оценить их эффективность. Так вы узнаете, действительно ли ваше решение было корректным.

Каждый этап хорошо декомпозированной задачи имеет какой-нибудь весомый результат: подготовленные к анализу данные, график, программа. Выполнение всех шагов приводит к заранее определённому конечному результату.

**Уточнение задачи (у заказчика, у бизнеса):**

1. Уточнение первоначального запроса

2. Формулировка промежуточного результата

3. Уточнение требований к результату

Примеры:

Нам нужна выгрузка по всем платящим пользователям за вчерашний день! Очень желательно в ближайшее время.

1. Зачем?

Хотим посмотреть, какие подписки они оформили.

Надо проанализировать, как пользователи работают с нашим приложением на Android.

1. Какой элемент в работе с приложением вы хотите проанализировать?

Как пользователи начинают играть: из своего профиля или через главное меню.