1. `data\_preprocessing.py`

Задача : Загрузка и предобработка данных.

Что делает :

- Загрузка данных : Считывает данные из CSV-файла с помощью `pandas`.

- Предобработка данных :

- Преобразует столбец `date` в формат даты.

- Устанавливает столбец `date` в качестве индекса.

- Разделяет данные на признаки (`X`) и целевую переменную (`y`). Признаки включают столбцы `price` и `advertising\_cost`, а целевая переменная — столбец `sales`.

- Разделение данных : Разделяет данные на обучающую и тестовую выборки с помощью `train\_test\_split`.

- Масштабирование данных : Нормализует данные с помощью `StandardScaler`, чтобы улучшить производительность модели.

2. `model.py`

Задача : Создание и обучение модели.

Что делает :

- Создание модели : Создает полносвязную нейронную сеть с несколькими слоями.

- Первый слой принимает входные данные и имеет 64 нейрона с функцией активации ReLU.

- Добавляется слой Dropout для регуляризации, чтобы предотвратить переобучение.

- Добавляются еще два слоя с 32 и 16 нейронами соответственно, также с функцией активации ReLU.

- Выходной слой имеет один нейрон с линейной функцией активации для регрессии.

- Компиляция модели : Компилирует модель с оптимизатором Adam и функцией потерь MSE (среднеквадратичная ошибка).

- Обучение модели : Обучает модель на обучающих данных в течение 50 эпох с размером пакета 32 и выполняет валидацию на 20% обучающих данных.

3. `evaluation.py`

Задача : Оценка модели и визуализация результатов.

Что делает :

- Прогнозирование : Использует обученную модель для прогнозирования на тестовой выборке.

- Визуализация : Строит график, сравнивая фактические продажи с прогнозируемыми продажами.

- Оценка модели : Вычисляет метрики MSE (среднеквадратичная ошибка) и MAE (средняя абсолютная ошибка) для оценки производительности модели.

4. `main.py`

Задача : Основной файл, который запускает весь процесс.

Что делает :

- Загрузка и предобработка данных : Вызывает функцию `load\_and\_preprocess\_data` из `data\_preprocessing.py`, чтобы загрузить и обработать данные.

- Создание модели : Вызывает функцию `create\_model` из `model.py`, чтобы создать модель нейронной сети.

- Обучение модели : Вызывает функцию `train\_model` из `model.py`, чтобы обучить модель на предобработанных данных.

- Оценка модели : Вызывает функцию `evaluate\_model` из `evaluation.py`, чтобы оценить модель и визуализировать результаты.

- Сохранение модели : Сохраняет обученную модель в файл `sales\_demand\_model.h5` для дальнейшего использования.

Заключение:

Ваш проект состоит из нескольких файлов, каждый из которых отвечает за определенный этап процесса:

- `data\_preprocessing.py` : Загружает и предобрабатывает данные.

- `model.py` : Создает и обучает модель нейронной сети.

- `evaluation.py` : Оценивает модель и визуализирует результаты.

- `main.py` : Основной файл, который запускает весь процесс, вызывая функции из других файлов в нужном порядке.

Такая структура проекта упрощает его поддержку и расширение, а также делает код более читаемым и понятным.