API 参考

发行版本 *v1.0*

lihetong

2025 年 12 月 29 日

内容目录:

欢迎来到 EcommerceSalesAnalysis 项目文档!

这是一个电商平台销售数据分析项目，包含数据加载、清洗、探索、可视化、特征工程、模型训练和评估等模块。

CHAPTER 1

介绍

这是一个电商平台销售数据分析系统，主要功能包括：

- 数据加载与预处理
- 数据探索与分析
- 数据可视化
- 特征工程
- 模型训练与评估

CHAPTER 2

模块参考

2.1 data_loader

Functions

data_loader()	从数据库中获取数据或者是从 csv 中读取数据获取数据基本结构:return: 返回原始数据
get_basic_info()	获取数据基本信息
get_mysql_engine()	获取数据库连接:return:
load_from_mysql(engine, table_name)	从 MySQL 读取整张表数据:param engine: 数据库:param table_name: 表名:return: DataFrame 原始数据集
load_raw_data()	加载原始用户数据, 并规范列名 - 读取 CSV / 数据库 - 不做任何清洗与修改 - 保证“原始性”:return: 返回加载好的原始数据
save_to_mysql(df, table_name, engine)	load_raw_data 已经清洗过列名, 去除空格, 可以正常存入先使用 if 下面的代码导入数据, 然后使用 if 判断数据没问题:param df: 原始数据:param table_name: 提前设定好的表名:param engine: 数据库
table_has_data(engine, table_name)	

data_loader.**data_loader**() → DataFrame

从数据库中获取数据或者是从 csv 中读取数据获取数据基本结构:return: 返回原始数据

data_loader.**get_basic_info**() → dict

获取数据基本信息

:param df: 传入原始数据:return: 返回部分数据

data_loader.**get_mysql_engine**()

获取数据库连接:return:

data_loader.**load_from_mysql**(*engine*, *table_name: str*) → DataFrame

从 MySQL 读取整张表数据:param engine: 数据库:param table_name: 表名:return: DataFrame 原始数据集

data_loader.**load_raw_data**() → DataFrame

加载原始用户数据, 并规范列名 - 读取 CSV / 数据库 - 不做任何清洗与修改 - 保证“原始性” :return: 返回加载好的原始数据

data_loader.**save_to_mysql**(*df: DataFrame*, *table_name: str*, *engine*)

load_raw_data 已经清洗过列名, 去除空格, 可以正常存入先使用 if 下面的代码导入数据, 然后使用 if 判断数据没问题:param df: 原始数据:param table_name: 提前设定好的表名:param engine: 数据库

data_loader.**table_has_data**(*engine*, *table_name: str*) → bool

参数

- **engine**
- **table_name**

返回

2.2 data_clean

Functions

data_clean(df, numeric_cols, categorical_cols)	数据清洗主函数:return: 清洗好的新数据集 df_new
handle_categorical_missing(df, categorical_cols)	类别型特征缺失值处理：填充为'Unknown' :param df: 原始数据集:param categorical_cols: 类别型列:param fill_value: 填充值，默认为'Unknown' :return: 填充后的数据
handle_missing_values(df, numeric_cols[, ...])	数值型特征缺失值处理：1.
mark_abnormal_values(df)	异常值标记（不修改原始数值）当前仅对 age 进行简单规则标记 - age <= 0 或 age > 100 视为异常:param df: DataFrame :return: 添加异常标记列后的 DataFrame 数据集
remove_duplicates(df)	重复值处理 - 统计重复行数量 - 删除完全重复的行（保留第一条）:param df: DataFrame :return: 去重后的 DataFrame 数据集
save_clean_data(df, path)	保存清洗后的数据:param df: DataFrame :param path: 保存路径

data_clean.**data_clean**(*df: DataFrame*, *numeric_cols: list*, *categorical_cols: list*) → DataFrame

数据清洗主函数:return: 清洗好的新数据集 df_new

data_clean.**handle_categorical_missing**(*df: DataFrame*, *categorical_cols: list*, *fill_value: str = 'Unknown'*) → DataFrame

类别型特征缺失值处理：填充为'Unknown' :param df: 原始数据集:param categorical_cols: 类别型列:param fill_value: 填充值，默认为'Unknown' :return: 填充后的数据

data_clean.**handle_missing_values**(*df: DataFrame*, *numeric_cols: list*, *fill_value: int = -1*) → DataFrame

数值型特征缺失值处理：1. 使用指定值（默认 -1）填充缺失 2. 为每个数值特征创建缺失指示变量:param df: 原始 DataFrame 数据集:param numeric_cols: 数值列:return: 填充后的数据

data_clean.**mark_abnormal_values**(*df: DataFrame*) → DataFrame

异常值标记（不修改原始数值）当前仅对 age 进行简单规则标记 - age <= 0 或 age > 100 视为异常:param df: DataFrame :return: 添加异常标记列后的 DataFrame 数据集

data_clean.**remove_duplicates**(*df: DataFrame*) → DataFrame

重复值处理 - 统计重复行数量 - 删除完全重复的行（保留第一条）:param df: DataFrame :return: 去重后的 DataFrame 数据集

data_clean.**save_clean_data**(*df: DataFrame*, *path: str*) → None

保存清洗后的数据:param df: DataFrame :param path: 保存路径

2.3 data_explore

Functions

analyze_feature_by_group(df, group_col, ...)	分析某个特征在不同分组中的分布:param df: pandas DataFrame，输入的数据集:param group_col:str，分组列名（如'lifecycle'）:param feature_col:str，要分析的特征列名:param normalize:bool，是否计算比例而不是计数，默认为 True（计算比例）:return:DataFrame，交叉表显示特征在不同分组中的分布
data_explore()	展示所有探索的数据:return:
explore_categorical_features(df[, top_n])	分析类别型特征的分布情况:param df: 输入原始的数据集:param top_n:int，显示每个类别的前 N 个值，默认显示前 10 个:return: 字典，键为类别型列名，值为该列的值分布 Series
explore_correlation(df[, method, threshold])	分析数值特征之间的相关性:param df:DataFrame，原始数据:param method: 相关系数计算方法 ('pearson', 'spearman', 'kendall'), 默认 pearson :param threshold: 相关性阈值，用于筛选强相关特征对:return: corr_matrix：相关性矩阵 strong_corr：强相关特征对（DataFrame）
explore_missing_values(df)	统计各字段缺失率:return:DataFrame，包含每个字段的缺失数量和缺失率，按缺失率降序排列
explore_numeric_features(df)	分析数值型特征的描述性统计:param df: 输入原始的数据集:return:DataFrame，包含各数值型特征的统计量（计数、均值、标准差、最小值、25%、50%、75%、最大值）
split_columns_clean(df)	用于清洗数据划分列:param df: DataFrame 原始数据:return: 数值列和类别列

data_explore.**analyze_feature_by_group**(*df: DataFrame*, *group_col: str*, *feature_col: str*, *normalize: bool = True*) → DataFrame

分析某个特征在不同分组中的分布:param df: pandas DataFrame，输入的数据集:param group_col:str，分组列名（如'lifecycle'）:param feature_col:str，要分析的特征列名:param normalize:bool，是否计算比例而不是计数，默认为 True（计算比例）:return:DataFrame，交叉表显示特征在不同分组中的分布

data_explore.**data_explore**()

展示所有探索的数据:return:

data_explore.**explore_categorical_features**(*df: DataFrame*, *top_n: int = 10*) → dict

分析类别型特征的分布情况:param df: 输入原始的数据集:param top_n:int，显示每个类别的前 N 个值，默认显示前 10 个:return: 字典，键为类别型列名，值为该列的值分布 Series

data_explore.**explore_correlation**(*df: DataFrame*, *method: str = 'pearson'*, *threshold: float = 0.7*) → DataFrame

分析数值特征之间的相关性:param df:DataFrame，原始数据:param method: 相关系数计算方法 ('pearson', 'spearman', 'kendall'), 默认 pearson :param threshold: 相关性阈值，用于筛选强相关特征对:return:

corr_matrix : 相关性矩阵 strong_corr : 强相关特征对（DataFrame）

data_explore.**explore_missing_values**(*df: DataFrame*) → DataFrame

统计各字段缺失率:return:DataFrame，包含每个字段的缺失数量和缺失率，按缺失率降序排列

data_explore.**explore_numeric_features**(*df: DataFrame*) → DataFrame

分析数值型特征的描述性统计:param df: 输入原始的数据集:return:DataFrame，包含各数值型特征的统计量（计数、均值、标准差、最小值、25%、50%、75%、最大值）

data_explore.**split_columns_clean**(*df: DataFrame*)

用于清洗数据划分列:param df: DataFrame 原始数据:return: 数值列和类别列

2.4 data_visualize

Functions

data_visualize()	数据可视化总入口:return:
plot_categorical_distribution(df, ...)	绘制所有类别型特征的频数分布
plot_category_vs_numeric(df, ...)	类别特征 vs 数值特征的箱线图
plot_correlation_heatmap(df)	绘制数值型特征相关性热力图
plot_numeric_distribution(df, numeric_cols)	绘制所有数值型特征的分布图

data_visualize.**data_visualize**() → None

数据可视化总入口:return:

data_visualize.**plot_categorical_distribution**(*df: DataFrame*, *categorical_cols: list*)

绘制所有类别型特征的频数分布

data_visualize.**plot_category_vs_numeric**(*df: DataFrame*, *categorical_cols: list*, *numeric_cols: list*)

类别特征 vs 数值特征的箱线图

data_visualize.**plot_correlation_heatmap**(*df: DataFrame*)

绘制数值型特征相关性热力图

data_visualize.**plot_numeric_distribution**(*df: DataFrame*, *numeric_cols: list*)

绘制所有数值型特征的分布图

2.5 feature_engineer

Functions

add_missing_indicators(df, numeric_cols[, ...])	为数值型特征添加缺失值指示变量:param df:DataFrame 清洗好的数据集:param numeric_cols: 需要处理缺失值的数值型列名列表:param fill_value: 表示缺失值的占位值，默认是 -1 :return: 处理后的
build_features_for_dl(df[, scaler])	深度学习模型特征工程主入口特点：- 数值特征：标准化 - 类别特征：整数编码（Embedding 用）- 不做 One-Hot :param df:DataFrame 清洗好的数据集:return: df_new:DataFrame 编码后的数据 scaler: 数值特征标准化器 encoders: 编码器
build_features_for_ml(df, numeric_cols, ...)	传统机器学习模型（LR / RF）特征工程主入口步骤：1. 数值特征标准化 2. 类别特征 One-Hot 编码:param df: :param numeric_cols: :param categorical_cols: :param scaler: :return: df_new:DataFrame 最终可用于模型训练的数据 scaler: 数值特征标准化器.
encode_categorical_for_dl(df, categorical_cols)	深度学习用的类别特征编码 - 每个类别列编码成整数 - 后续交给 Embedding 层:param df:DataFrame 清洗好的数据集:param categorical_cols: 类别型列名:return: df_new:DataFrame 编码后的数据 encoders: 编码器
one_hot_encode(df, categorical_cols)	:param df:DataFrame 清洗好的数据集:param categorical_cols: 类别型列名:return:One-Hot 编码后的数据
scale_numeric_features(df, numeric_cols[, ...])	数值特征标准化:param df:DataFrame :param numeric_cols: 数值型列名:param scaler: 训练阶段传 None，预测阶段传已有 scaler :return: df_new:DataFrame 标准化后的数据 scaler:
split_columns_by_type(df, target_col)	自动划分需要标准化的列, 数值列和缺失值指示异常列:param df: DataFrame 添加指示变量的数据:return: numeric_cols: 标准化列 categorical_cols: 类别列 indicator_cols: 缺失值指示列和异常列

feature_engineer.**add_missing_indicators**(*df: DataFrame*, *numeric_cols: list*, *missing_value: int | float = -1*) → DataFrame

为数值型特征添加缺失值指示变量:param df:DataFrame 清洗好的数据集:param numeric_cols: 需要处理缺失值的数值型列名列表:param fill_value: 表示缺失值的占位值，默认是 -1 :return: 处理后的

feature_engineer.**build_features_for_dl**(*df: DataFrame*, *scaler: StandardScaler | None = None*)

深度学习模型特征工程主入口特点：- 数值特征：标准化 - 类别特征：整数编码（Embedding 用）- 不做 One-Hot :param df:DataFrame 清洗好的数据集:return:

df_new:DataFrame 编码后的数据 scaler: 数值特征标准化器 encoders: 编码器

feature_engineer.**build_features_for_ml**(*df: DataFrame*, *numeric_cols: list*, *categorical_cols: list*, *scaler: StandardScaler | None = None*)

传统机器学习模型（LR / RF）特征工程主入口步骤：1. 数值特征标准化 2. 类别特征 One-Hot 编码:param df: :param numeric_cols: :param categorical_cols: :param scaler: :return:

df_new:DataFrame 最终可用于模型训练的数据 scaler: 数值特征标准化器

feature_engineer.**encode_categorical_for_dl**(*df: DataFrame*, *categorical_cols: list*)

深度学习用的类别特征编码 - 每个类别列编码成整数 - 后续交给 Embedding 层:param df:DataFrame 清洗

好的数据集:param categorical_cols: 类别型列名:return:

df_new:DataFrame 编码后的数据 encoders: 编码器

feature_engineer.**one_hot_encode**(*df: DataFrame*, *categorical_cols: list*) → DataFrame

:param df:DataFrame 清洗好的数据集:param categorical_cols: 类别型列名:return:One-Hot 编码后的数据

feature_engineer.**scale_numeric_features**(*df: DataFrame*, *numeric_cols: list*, *scaler: StandardScaler | None = None*)

数值特征标准化:param df:DataFrame :param numeric_cols: 数值型列名:param scaler: 训练阶段传 None，预测阶段传已有 scaler :return:

df_new:DataFrame 标准化后的数据 scaler:

feature_engineer.**split_columns_by_type**(*df: DataFrame*, *target_col: str*)

自动划分需要标准化的列, 数值列和缺失值指示异常列:param df: DataFrame 添加指示变量的数据:return:

numeric_cols: 标准化列 categorical_cols: 类别列 indicator_cols: 缺失值指示列和异常列

2.6 config

2.7 main

Functions

main()	程序主函数, 一键启动:return:

main.**main**()

程序主函数, 一键启动:return:

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

CHAPTER 4

项目介绍

EcommerceSalesAnalysis 是一个电商平台销售数据分析项目。该项目旨在通过数据科学的方法对电商平台的销售数据进行深入分析，提供有价值的业务洞察。

4.1 功能特性

- **数据加载**: 支持多种格式的销售数据加载
- **数据清洗**: 自动识别和处理数据中的异常值、缺失值
- **数据探索**: 提供丰富的统计分析功能
- **数据可视化**: 生成直观的图表展示数据特征
- **特征工程**: 提取有用的特征用于模型训练
- **模型训练**: 使用机器学习算法预测销售趋势
- **模型评估**: 评估模型性能并提供可视化结果

4.2 技术栈

- Python 3.x
- Pandas - 数据处理
- NumPy - 数值计算
- Matplotlib/Seaborn - 数据可视化
- Scikit-learn - 机器学习
- Sphinx - 文档生成

Python 模块索引

索引

O

P

R

S

T