MACHINE LEARNING PROJECT

Low Level Design Report

On

BACKORDER PREDICTION

Made by: Gaurav Singh

Domain: E-commerce

TABLE OF CONTENTS

1. INTRODUCTION

1.1 What is Low-Level design document?

Low-Level Design (LLD) is a component-level design process that follows a step-by-step refinement process. It provides the details and definitions for the actual logic for every system component. It is based on HLD but digs deeper, going into the separate modules and features for every program in order to document their specifications.

1.2 Scope

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code. Low-level design is created based on the high-level design. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

A good low-level design document makes the program easy to develop when proper analysis is utilized to create a low-level design document. The code can then be developed directly from the low-level design document with minimal debugging and testing. Other advantages include lower cost and easier maintenance.

1.3 Constraints

We will predict backorder or not on the the given data columns which the model has used for training.

2. Technical Specifications

2.1 Dataset

- **sku** : unique id for a product
- **national_inv**: present national level of inventory of the product
- **lead_time** : the amount of time between when a purchase order is placed to replenish products and when the order is received in the warehouse.
- **in_transit_qty** : qty of goods in transit
- **forecast_3_month** : Forecasted sales of the product for the next 3 months.
- **forecast_6_month** : Forecasted sales of the product for the next 6 months.
- **forecast_9_month** : Forecasted sales of the product for the next 9 months.
- **sales_1_month** : Actual Sales of the product in the last 1 month.
- **sales_3_month** : Actual Sales of the product in the last 3 months.
- **sales_6_month** : Actual Sales of the product in the last 6 months.
- **sales_9_month** : Actual Sales of the product in the last 9 months.
- **min_bank**: Minimum amount of stock recommended to have.
- **potential_issue**: Any problem identified with the product or part.
- **pieces_past_due** : product kept for long time, past their expiry date.
- **perf_6_month_avg** : Average performance of product over last 6 months.
- **perf_12_month_avg**: Average performance of product over last 12 months.
- **local_bo_qty** : (undeliverable orders / total number of orders)*100.
- **deck_risk** : risk associated with keeping the items in stock

- **ppap_risk** : used to determine whether a production will produce parts with consistency and repeatability
- **stop_auto_buy** : Has the auto buy for the product, which was back ordered, cancelled.

TARGET FEATURE : went_on_backorder - Whether an items was backordered or not

2.2 Overview

sku	national_i	lead_time	in_transit	forecast_3	forecast_6	forecast_9	sales_1_m	sales_3_m	sales_6_m	sales_9_m	min_bank	potential	pieces_pa	perf_6_m	perf_12_n	local_bo_	deck_risk	oe_constr	ppap_risk	stop_auto	rev_stop	went_on_backorder
1883577	4	8	0	0	0	0	0	0	0	0	0	No	0	0.6	0.47	0	No	No	Yes	Yes	No	No
1883578	5	8	0	0	0	2	0	1	3	10	2	No	0	0.91	0.89	0	No	No	No	Yes	No	No
1883579	1417	8	92	614	1634	2181	185	744	1639	2410	204	No	0	0.84	0.84	0	No	No	No	Yes	No	No
1883580	1022	8	12	0	0	0	21	95	201	268	47	No	0	0.93	0.9	0	No	No	No	Yes	No	No
1883581	0	8	0	0	0	0	0	0	0	0	0	No	0	0.97	0.98	0	No	No	No	Yes	No	No
1883582	325	8	51	75	100	225	42	135	312	456	31	No	0	0.96	0.96	0	No	No	No	Yes	No	No
1883583	37	8	0	43	78	121	12	75	127	182	20	No	0	0.83	0.83	0	No	No	No	Yes	No	No
1883584	12	8	0	0	0	0	0	0	1	5	2	No	0	0.76	0.77	0	No	No	No	Yes	No	No
1883585	42	8	10	88	159	232	25	82	150	235	24	No	0	0.69	0.69	0	No	No	No	Yes	No	No
1883586	966	8	215	709	1658	2699	216	791	1663	2691	402	No	0	0.99	0.91	0	No	No	No	No	No	No
1883587	0	8	0	3	3	3	0	4	4	4	0	No	0	0.17	0.33	0	No	No	No	Yes	No	No

2.3 Input Schema


```
{
    "SampleFileName": "BackOrder_08012020_120000.csv",
        "LengthOfDateStampInFile": 8,
        "LengthOfTimeStampInFile": 6,
        "NumberofColumns" : 23 ,
        "ColName": {
            "sku" : "Integer" ,
            "national_inv" : "float" ,
            "lead_time" : "float" ,
            "in_transit_qty" : "float" ,
            "forecast_3_month" : "float",
            "forecast_6_month" : "float",
            "forecast_9_month" : "float",
            "sales_1_month" : "float",
            "sales_3_month" : "float",
            "sales_6_month"  : "float",
            "sales_9_month": "float",
            "min_bank" : "float" ,
            "potential_issue" : "object",
            "pieces_past_due" :   "float" ,
            "perf_6_month_avg"  : "float" ,
            "perf_12_month_avg" : "float" ,
            "local_bo_qty": "float" ,
            "deck_risk": "object" ,
            "oe_constraint": "object",
            "ppap_risk": "object",
            "stop_auto_buy": "object",
            "rev_stop": "object",
            "went_on_backorder": "object"
        }
    }
```

3.Prediction

The user gives required information mentioned in the schema. The system should be able to predict whether it’s backorder or not based on the user information.

4.Logging

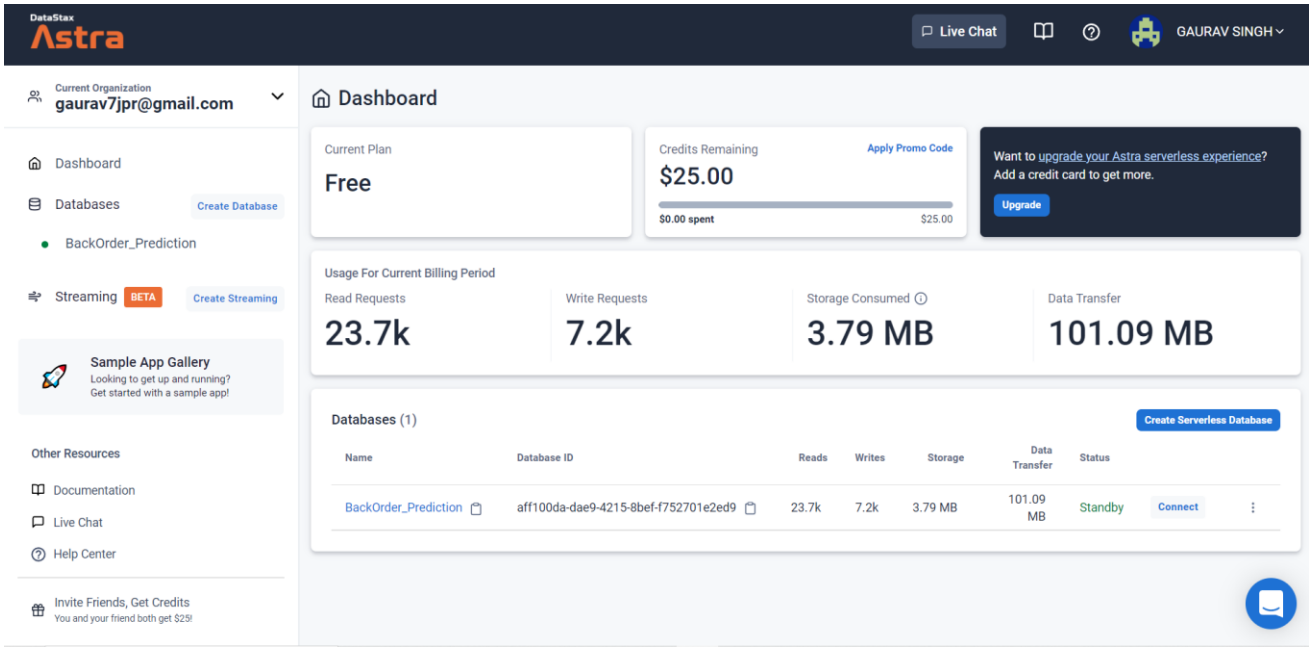
Logs are created from the start , so that we can debug easily if any issue arises.

The system should log every event so that the user will know what process is running internally. Logging is implemented using python's standard logging library.
Initial step-by-step description:-

- The system should be able to log each and every system flow.
- System must be able to handle logging at greater scale because it helps debugging the issue and hence it is mandatory to do.

5.Database

The input files from client are added to the database after data file validation and performing all the data preprocessing steps. The final data is added to the database in a good data table.The data can be downloaded into csv for further use in Model Development.



6.Deployment

The app is deployment in Heroku cloud Platform.

7.Technology Stack

Front End	Html, Css, Js,Bootstrap
Back End	Flask, Pandas, NumPy, scikit-learn etc
Database	Cassandra
Deployment	Heroku