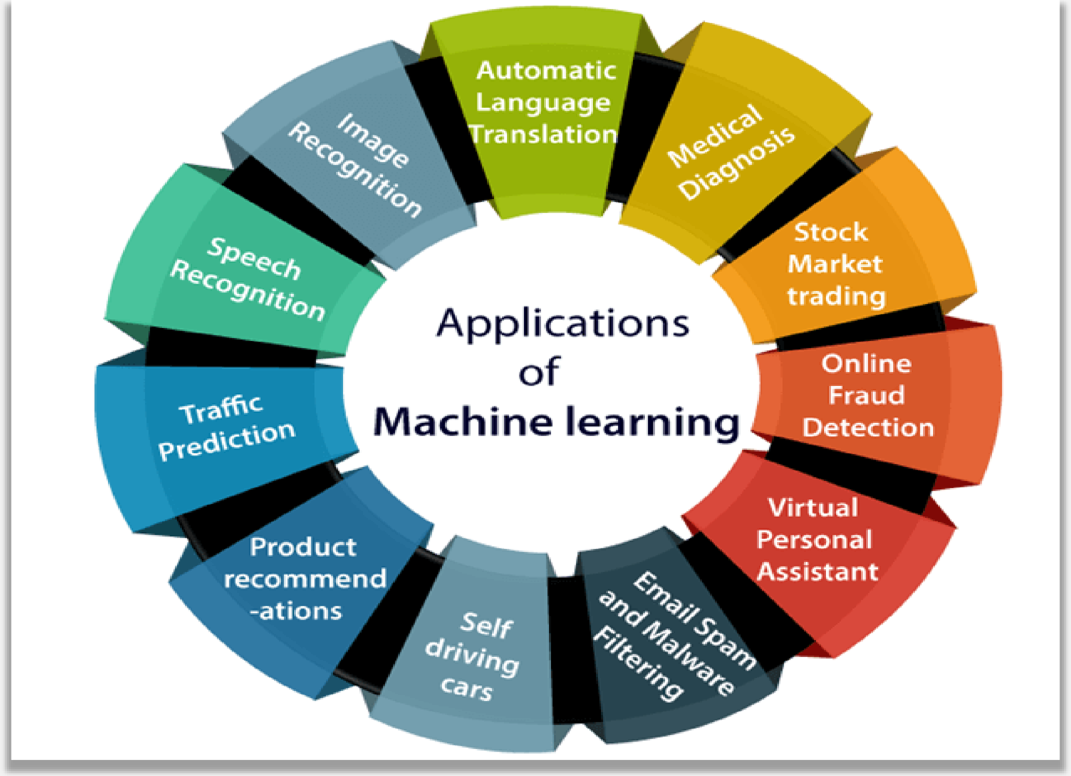
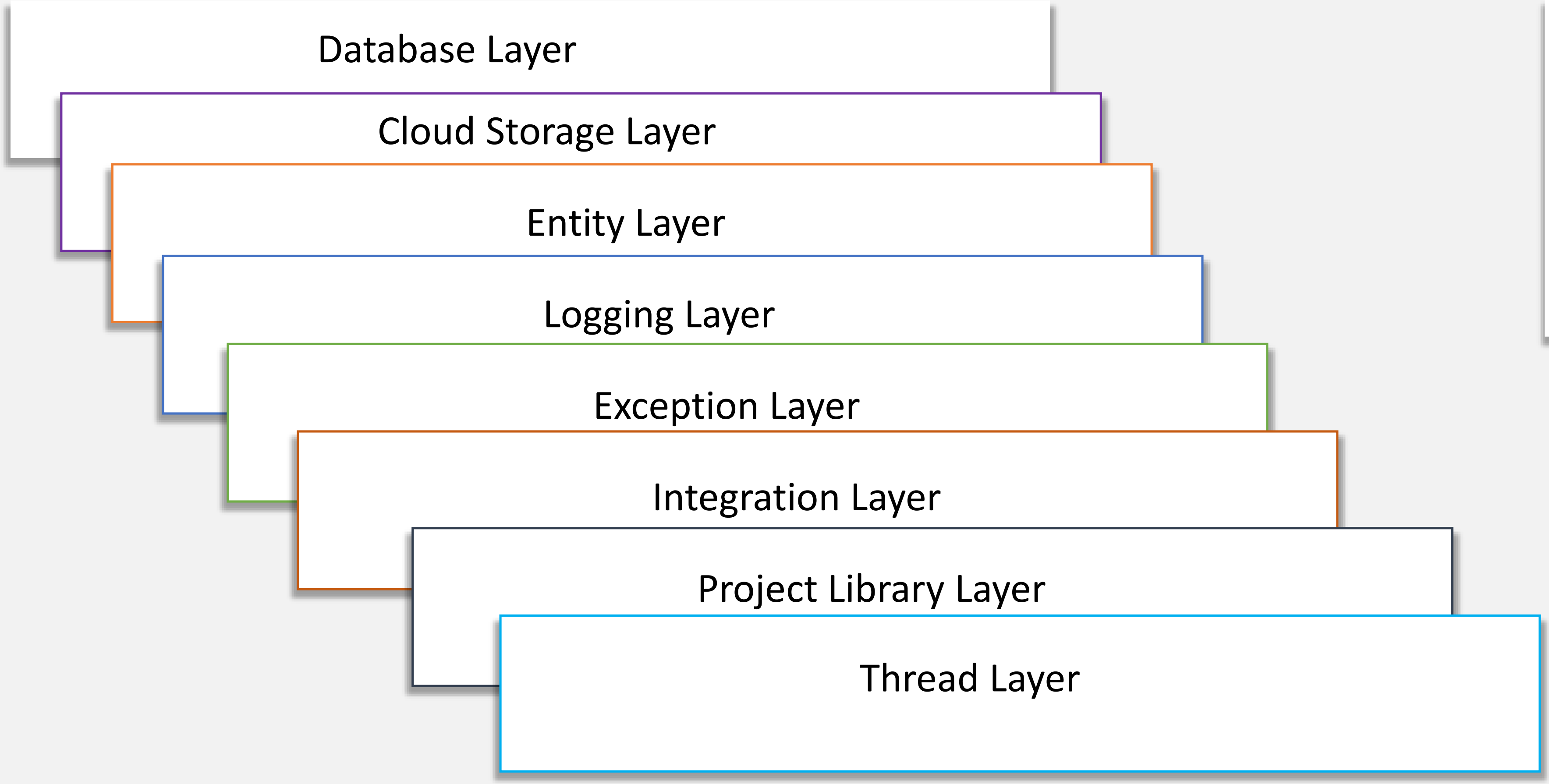
Multiple Machine Learning Project Design As One Project


Applications of Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition

Objective - Development of a python cloud machine learning project for automating training and prediction for multiple projects over the cloud platforms.

LAYERED ARCHITECTURE APPROACH



Database Layer

Cloud Storage Layer

Entity Layer

Logging Layer

Exception Layer

Integration Layer

Project Library Layer

Thread Layer

Other Essential Component

1. Controller
2. Templates
3. Static Component

main.py

DATABASE LAYER

Database layer has to provide high level library to perform all database related operation. Application perform crud operation on database.

1. Creating record.
2. Updating record.
3. Removing record.
4. Selecting record.

mongoDB

SQL

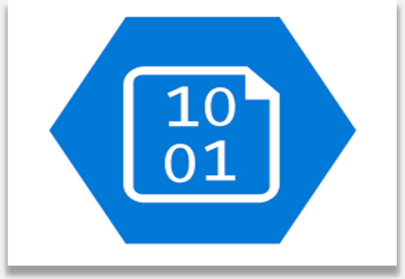
cassandra

MySQL

Sr. No.	Relation Database	NoSQL Database
1	Table	Collection
2	Record	Document
3	Structured	Semi Structured

CLOUD STORAGE LAYER

Microsoft Azure



Azure blob Storage

Amazon Web Services

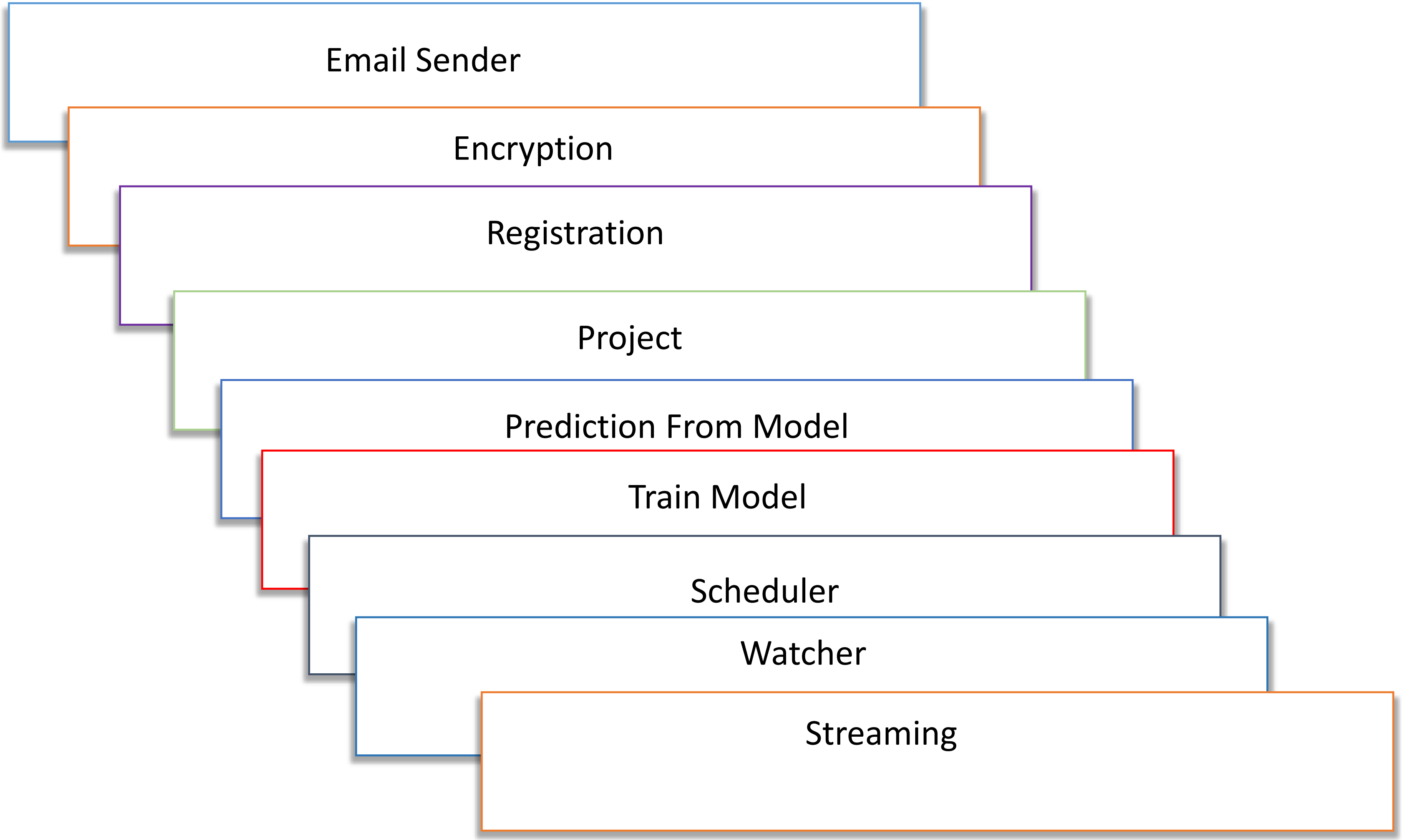
AWS s3 bucket

Google Cloud Platform

Google cloud Storage

Note: You need to build high level library to perform all file related activities on cloud in cloud storage layer. Build same interface in for all cloud.

Entity Layer



Email Sender

Encryption

Registration

Project

Prediction From Model

Train Model

Scheduler

Watcher

Streaming

Logging Layer

Logging is typically a key aspect of any production application; this is because it is important to provide appropriate information to allow future investigation following some event or issue in such applications.

These investigations include:

- **Diagnosing failures;** that is why did an application fail/crash.

- **Identifying unusual or unexpected behaviour**; which might not cause the application to fail but which may leave it in an unexpected state or where data may be corrupted etc.

- **Identifying performance or capacity issues;** in such situations the application is performing as expected by it is not meeting some non-functional requirements associated with the speed at which it is operating or its ability to scale as the amount of data or the number of users grows.

- Dealing with attempted malicious behaviour in which some outside agent is attempting to affect the behaviour of the system or to acquire information which they should not have access to etc. This could happen for example, if you are creating a Python web application and a user tries to hack into your web server.

Exception Layer

What Is an Exception?

The term exception is shorthand for the phrase "exceptional event.“

When an error occurs within a method, the method creates an object and hands it off to the runtime system. The object, called an exception object, contains information about the error, including its type and the state of the program when the error occurred. Creating an exception object and handing it to the runtime system is called throwing an exception.

After a method throws an exception, the runtime system attempts to find something to handle it. The set of possible "somethings" to handle the exception is the ordered list of methods that had been called to get to the method where the error occurred. The list of methods is known as the call stack

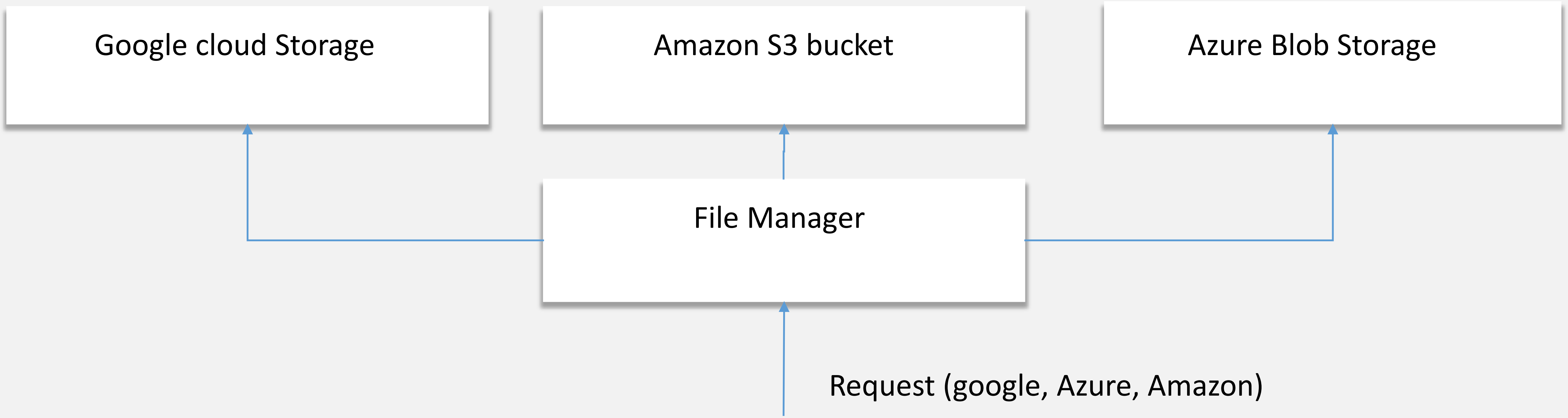
We need to build our customize exception which can help us to give us proper error detail so that we can analyze it better and try to resolve it permanently if possible.

Integration Layer

One of the most important layer among all.

We can implement Integration library which allows application to use cross cloud platform as storage and even multiple database with help of same code.

It will act as routing and propagate the incoming request to appropriate cloud storage or database,



Google cloud Storage

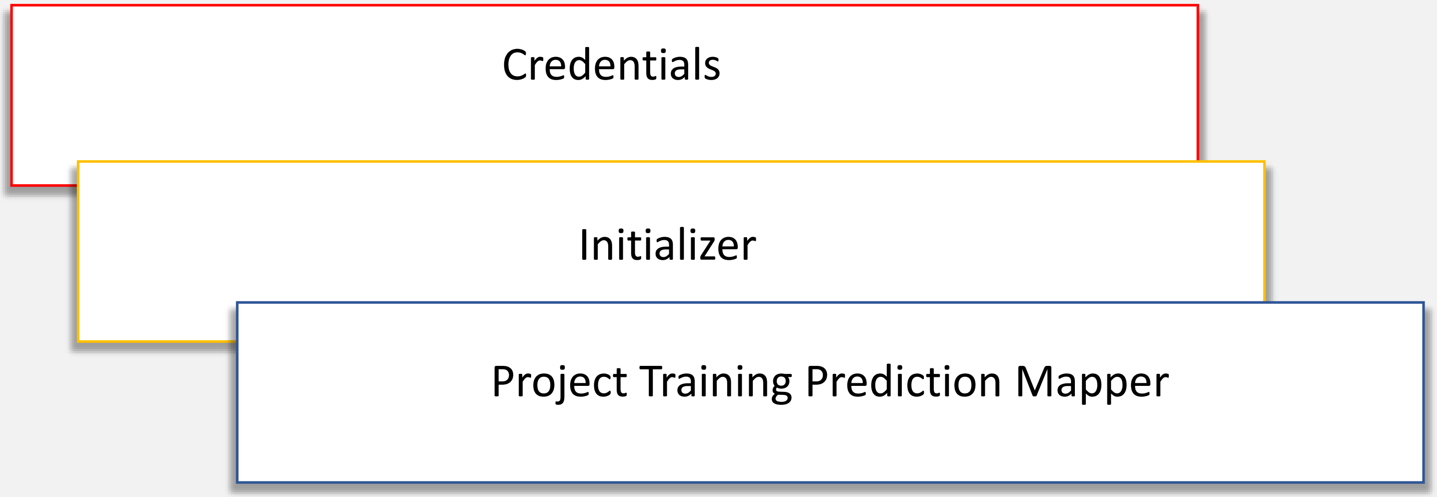
Amazon S3 bucket

Azure Blob Storage

File Manager

Request (google, Azure, Amazon)

Project Library Layer



Credentials

Initializer

Project Training Prediction Mapper

Credentials: Project credentials has to be manage properly in credentials

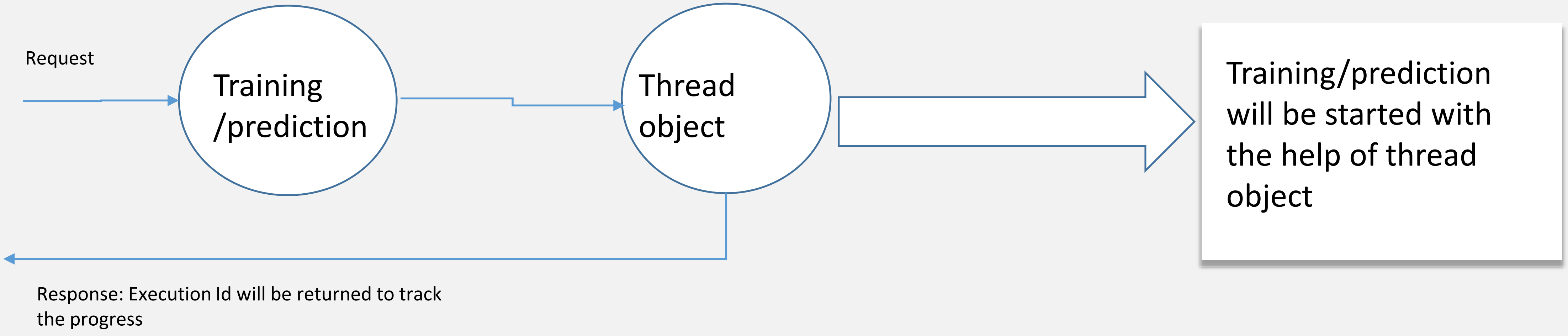
Initializer: It supports machine learning project to begin execution.

Project Training Prediction Mapper: It maps the appropriate training and prediction class to perform training and prediction

Thread Layer

Threading allow us to run training and prediction separately from other execution.

Thread layer: We have build high level library to obtain a thread object for training or prediction operation whenever requested.


Request
Training /prediction
Thread object
Training/prediction will be started with the help of thread object
Response: Execution Id will be returned to track the progress

Components

Authentication Controller

File Operation Controller

Home Controller

Machine Learning Controller

Project Controller

Scheduler Controller

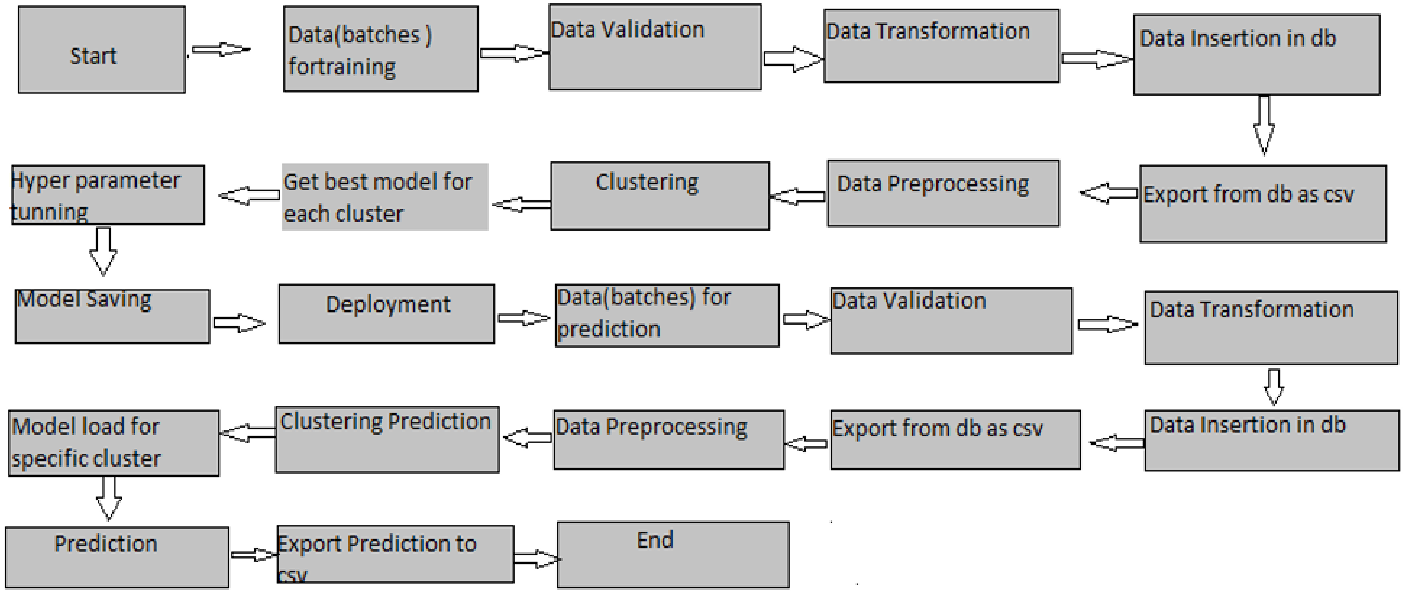
Visualization Controller

Watcher Controller

Other Essential Component

1. Controller
2. Templates
3. Static Component

Architecture for an added project



Data Sharing Agreement :

For each project we add, we will have a data sharing agreement which will contain the following information:-

Sample file name (ex fraudDetection_20062021_101010)
Length of date stamp(8 digits)
Length of time stamp(6 digits)
Number of Columns
Column names
Column data type

Data Validation and Data Transformation :

-->Name Validation - Validation of files name as per the DSA. We have created a regex pattern for validation.
After it checks for date format and time format if these requirements are satisfied, we move such files to "Good_Data_Folder" else "Bad_Data_Folder.“
-->Number of Columns – Validation of number of columns present in the files, and if it doesn't match then the file is moved to "Bad_Data_Folder.“
-->Name of Columns - The name of the columns is validated and should be the same as given in the schema file.
If not, then the file is moved to "Bad_Data_Folder".
-->Data type of columns - The data type of columns is given in the schema file. It is validated when we insert the files into Database.
If the datatype is wrong, then the file is moved to "Bad_Data_Folder".
-->Null values in columns - If any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to "Bad_Data_Folder".

Data Insertion in Database:

Table creation :- Table name “t_motorpv_fraud" is created in the database for inserting the files.
If the table is already present then new files are inserted in the same table.
Insertion of files in the table - All the files in the "Good_Data_Folder" are inserted in the
above-created table. If any file has invalid data type in any of the columns,
the file is not loaded in the table

Model Training

1)Data Export from Db : ->The accumulated data from db is exported in csv format for model training
2)Data Preprocessing
->Performing EDA to get insight of data like identifying distribution , outliers ,trend among data etc.
->Check for null values in the columns. If present impute the null values.
->Encode the categorical values with numeric values.
->Perform Standard Scalar to scale down the values.
3)Clustering –
->KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using KneeLocator function. The idea behind clustering is to implement different algorithms on the structured data
->The Kmeans model is trained over preprocessed data, and the model is saved for further use in prediction.
4)Model Selection –
->After the clusters are created, we find the best model for each cluster. By using 2 algorithms “SVM” and "XGBoost". For each cluster both the hyper tunned algorithms are used. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

Prediction

->The testing files are shared in the batches and we perform the same
Validation operations ,data transformation and data insertion on them.
->The accumulated data from db is exported in csv format for prediction
->We perform data pre-processing techniques on it.
->KMeans model created during training is loaded and clusters
for the preprocessed data is predicted
->Based on the cluster number respective model is loaded and is used
 to predict the data for that cluster.
->Once the prediction is done for all the clusters. The predictions are
saved in csv format and shared.

Q&A

Q1) What’s the source of data?
The data for training is provided by the client in multiple batches
and each batch contain multiple files
Q 2) What was the type of data?
The data was the combination of numerical and Categorical values.
Q 3) What’s the complete flow you followed in this Project?
Project architecture and flow has been shown in above slides.
Q 4) After the File validation what you do with incompatible file
or files which didn’t pass the validation?
Files like these are moved to the Achieve Folder and a list of these files has been
shared with the client and we removed the bad data folder.
Q 5) How logs are managed?
We are using different logs as per the steps that we follow in validation and
modeling like File validation log , Data Insertion ,Model Training log , prediction log
etc.
Q 6) What techniques were you using for data pre-processing?
Removing unwanted attributes
Visualizing relation of independent variables with each other and output variables
Checking and changing Distribution of continuous values
Removing outliers
Cleaning data and imputing if null values are present.
Converting categorical data into numeric values.
Scaling the data

Q 7) How training was done or what models were used?
Before diving the data in training and validation set we performed
clustering over fit to divide the data into clusters.
As per cluster the training and validation data were divided.
The scaling was performed over training and validation data
Algorithms like random forest , XGBoost were used based on the final scores.
model was used for each cluster and we saved that model .
Q 8) How Prediction was done?
The testing files are shared by the client .We Perform the same life cycle
till the data is clustered .Then on the basis of cluster number model is loaded
and perform prediction. In the end we get the accumulated data of predictions.
Q9) What about CI/CD and deployment?
The CI/CD pipeline has been set up using circleci and deployment has been done on heroku.

THANK YOU