**Project:1 - Project management Software and predict the Budget**

**ML Type : Regression**

**Problem statement:**

A project manager’s role demands creating a smoother process, deriving reliable outcomes, and coaching individuals and teams to perform more effectively. Given the nature of the job, project managers cannot afford to lose time doing repetitive tasks. However, it turns out that project managers spend a large portion of their time doing administrative tasks such as planning, tracking, analysis of projects, allocate resources like equipment rooms, staff, & other resources.

The good news is that artificial intelligence is stepping in this field and transforming the way projects are managed across the world. AI tools can reduce the workload of project managers by helping them in project scheduling, resource allocation, forecasting based on data.

**Solution:**

Create the AI based all in one platform.

**Benefits:**

* AI based all-in-one project planning software that provides solutions for project management, collaboration, resource management, and task management and also automation capabilities allow users to manage projects, portfolios, and work systems in one place.
* The AI-based software manages and monitors strategies through the execution and generates reports from various internal data and also works amazingly to visualize.
* AI-based will help to predict the budget for various portfolios.

**Data source:**

* Portfolio registration page
* Product registration page
* Project registration page
* Quality /Risk metrics.
* Resource data
* Planning data
* Other software API data.

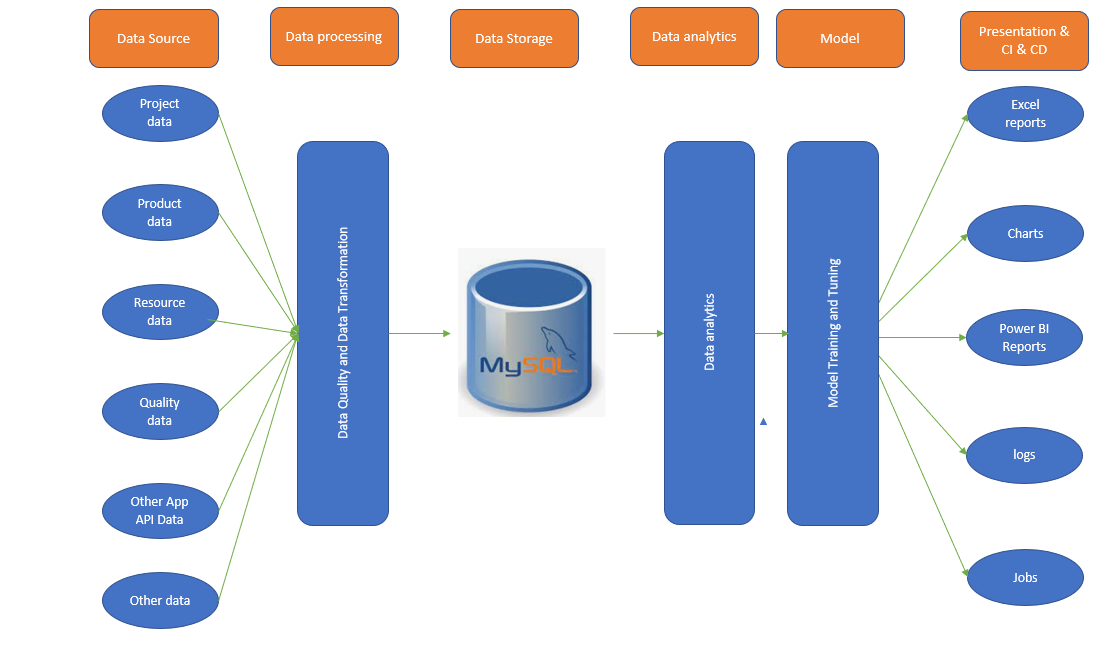
**Data Size:**

* Approximately 1000 records per week

**Data Format:**

* .XLSX
* .CSV
* .XLSB

**System Architecture:**



**Data sharing agreement: ( Master data management )**

* File name should be defined nomenclature
* File creation date
* File processing date
* Column Name
* Number of column
* Column data Type.
* Year

**Data Preprocessing:**

* Template validation
* Row wise null value check
* Data Transformation

**Dashboards:**

* Portfolio budget consumption
* Product budget consumption
* Project budget consumption
* Resource /Material/non resource utilization

**AI Stack:**

* 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.
* To train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.
* Model Selection - After clusters are created, we find the best model for each cluster. We are using two algorithms, "Random Forest" and "XGBoost". For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the accuracy/RMSE 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

**Project:2 Classify the diesel Pump good or Bad**

**ML Type: – Classification**

**Problem statement:**

Diesel Pumps that come for repair are assembled and manually tested. A complaint was received from an external customer such pumps are provides the reduced performance. Upon investigation by technical team, it was found that the issue was due to a missing mechanical part in the pump. Currently, the manufacturing unit has no mechanism to detect such problems during repair.

**Solution:**

Mount the industrial sensor on the pump during simulation Test and observe pattern the classify the good or bad pumps.

**Benefits:**

* Customer royalty
* Reduced rework
* Live dashboard
* Tracking

**Data source:**

* Industrial sensor

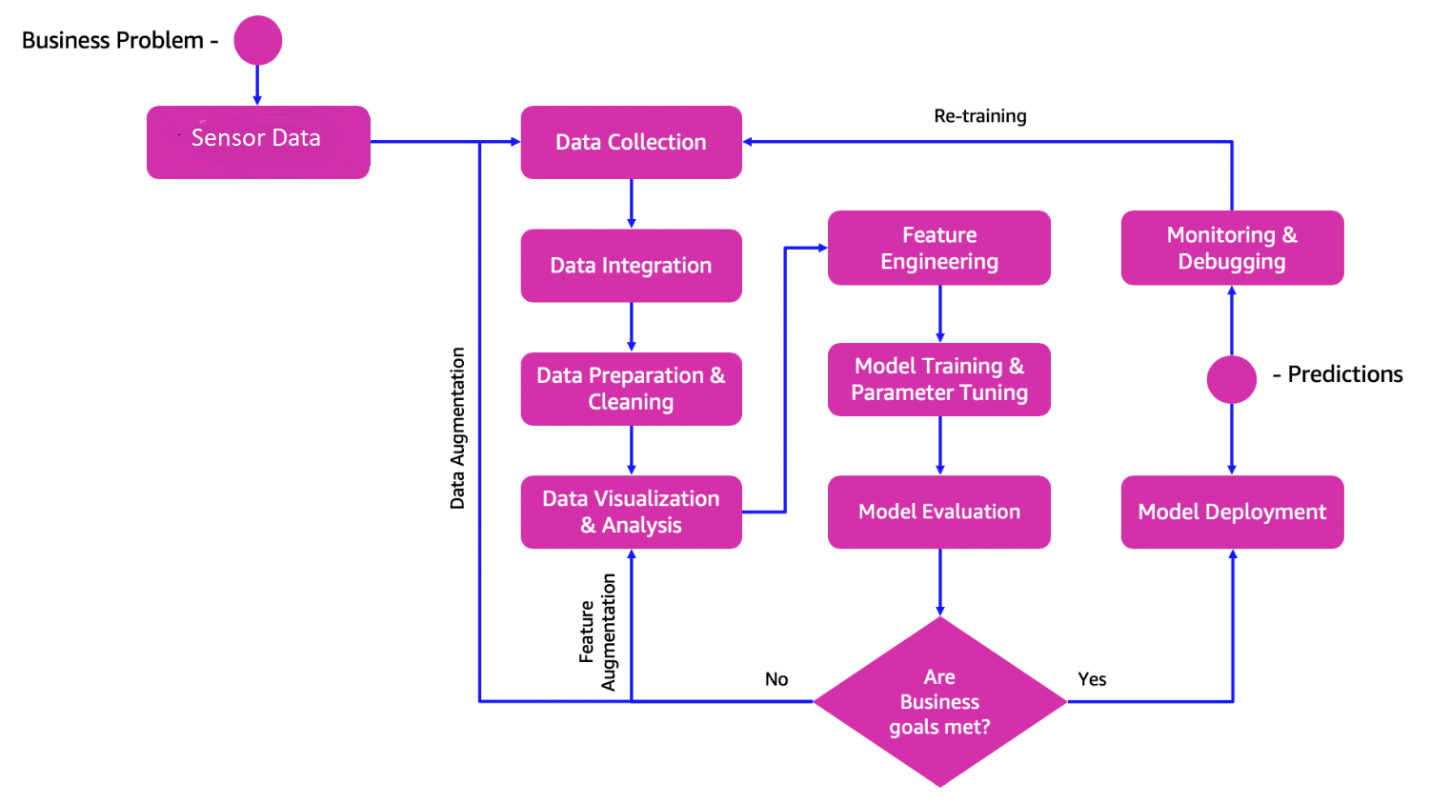
**Data format:**

* .bin file from sensor.
* Customized adapter/convertor script will covert bin file to .CSV format.

**Data Frequency:**

* Depends on baud rate selection. Approximately 10 sensor samples per minutes captured into bin file.

**System Architecture:**



**Data sharing agreement: ( Master data management )**

* File name should be defined nomenclature
* File creation date
* Sensor Time stamp( Always current Unix Time stamp)
* Column Name
* Number of column
* Column data Type.

.**Data Preprocessing:**

* Null value check
* Timestamp delta validation

**Data Insertion in Database:**

* Table creation in the database - Table with name - "Good\_Data", is created in the database for inserting the files in the "Good\_Data\_Folder" based on given column names and datatype in the schema file. If the table is already present, then the new table is not created and new files are inserted in the already present table as we want training to be done on new as well as old training files.
* 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 and is moved to "Bad\_Data\_Folder".

**Model Training:**

* Data Export from Db - The data in a stored database is exported as a CSV file to be used for model training.
* Data Preprocessing
* Perform the EDA like relationship between variables, distribution, outlier, visualize the Trend
* Check for null values in the columns. If present, impute the null values using the KNN imputer.
* Check if any column has zero standard deviation, remove such columns as they don't give any information during model training.
* Encoding and standard scaler

**AI Stack:**

* 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.
* To train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.
* Model Selection - After clusters are created, we find the best model for each cluster. We are using four algorithms, "Random Forest" ,”SVM”, "XGBoost" and Multilayer Neural network. For each cluster, four algorithms are passed with the best parameters derived from GridSearch. 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

**Project:3 – Human Activity Recognition**

**ML &DL Type: – Classification**

**Problem statement:**

Recognizing human activities and monitoring population behavior are fundamental needs of our society. Population security, crowd surveillance, healthcare support and living assistance, and lifestyle and behavior tracking are some of the main applications that require the recognition of human activities. HAR very popular due to two factors: the increasingly low cost of hardware and the wide spread of mobile devices equipped with inertial sensors. The use of smartphones to both acquire and process signals opens opportunities in a variety of application contexts such as surveillance, healthcare, and delivering.

**Solution:**

Human activity recognition is the problem of classifying sequences of accelerometer data recorded by smart phones into known well-defined movements. Movements are often normal indoor activities such as standing, sitting, jumping, and going up stairs. Sensors are often located on the subject such as a smartphone or vest and often record accelerometer data in three dimensions (x, y, z).

**Benefits:**

* Monitor health
* Discover activity pattern
* Detect the activity
* Improve wellbeing

**Data Source:**

* MEMS(Micro Electro-Mechanical System) accelerometer data

**Data format:**

* .bin file from sensor.
* Customized adapter/convertor script will covert bin file to .CSV format.

**Data Frequency:**

* Depends on baud rate selection. Approximately 20 observations per second (52Hz)samples per minutes captured into bin file.

**Activities:**

The specific activities performed were:

1: Walking

2: Sitting

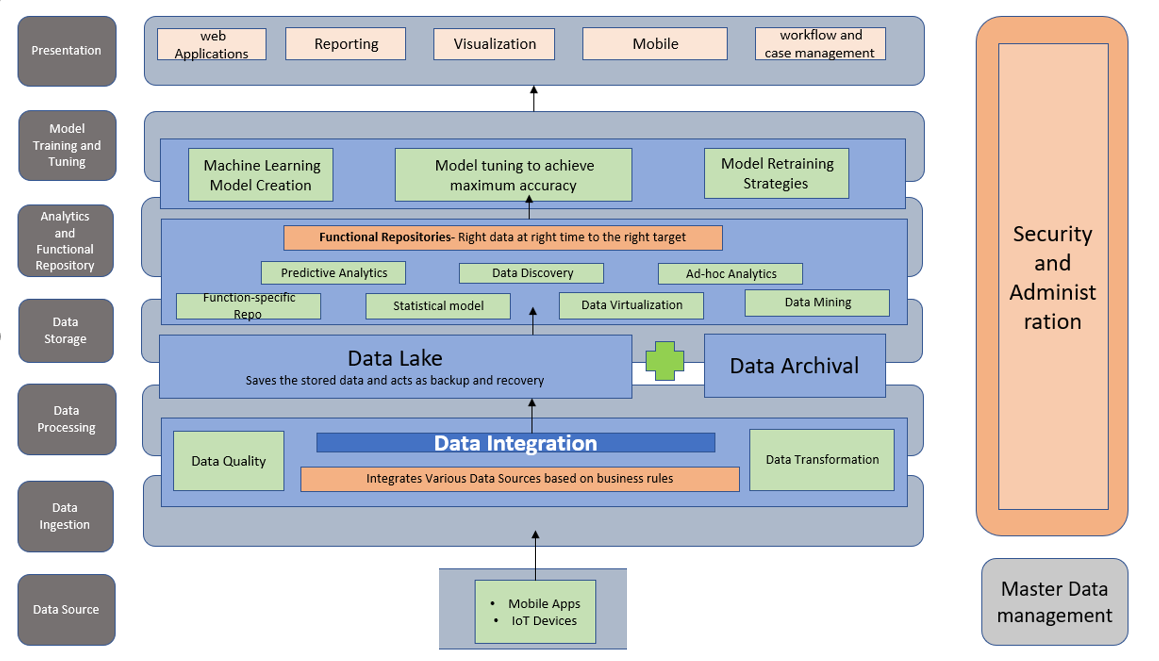
3: Standing

4: Jogging

5: Upstairs

6: Downstairs.

**System Architecture:**

****

**Data sharing agreement: ( Master data management )**

* File name should be defined nomenclature
* File creation date
* Sensor Time stamp( Always current Unix Time stamp)
* Column Name
* Number of column
* Column data Type.

**Data Preprocessing & Transformation:**

* Drop null values
* Drop the rows where the timestamp = 0
* Down-sampling the accelerometer observations to fractions of a second might be helpful, e.g. 1/4, 1/2, 1, 2 seconds.
* Truncating the first 60 seconds of the raw data might be prudent as it appears to be related to the start-up of the experiment and all subjects are performing
* Using simple outlier detection and removal methods such as values that are 3-to-4 times the standard deviation from the mean for each axis may be useful.
* Perhaps removing activities with relatively fewer observations.
* Perhaps rebalancing the activities by oversampling under-represented activities or under-sampling over-represented activities in a training dataset may help with modeling.
* Experimenting with different window sizes would be interesting (e.g. 1, 5, 10, 30 seconds), especially in corroboration with down-sampling of the observations.
* Standardizing the data per subject will almost certainly be required for any cross-subject model. Normalizing data across subjects after per-subject standardization may also be useful.

**Data Insertion in Database:**

* Table creation in the database - Table with name - "Good\_Data", is created in the database for inserting the files in the "Good\_Data\_Folder" based on given column names and datatype in the schema file. If the table is already present, then the new table is not created and new files are inserted in the already present table as we want training to be done on new as well as old training files.
* 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 and is moved to "Bad\_Data\_Folder".

**EDA:**

* Perform the EDA like relationship between variables, distribution, outlier, visualize the Trend

**Model Training:**

* Data Export from Db - The data in a stored database is exported as a CSV file to be used for model training.
* Data Preprocessing
* Check for null values in the columns. If present, impute the null values using the KNN imputer.
* Check if any column has zero standard deviation, remove such columns as they don't give any information during model training.
* Encoding and standard scaler

**AI Stack:**

* 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.
* To train data in different clusters. The Kmeans model is trained over preprocessed data and the model is saved for further use in prediction.
* Model Selection - After clusters are created, we find the best model for each cluster. We are using two ML algorithms and one DL Algorithm "Random Forest" ,"XGBoost" and LSTM RNN. For each cluster, Three algorithms are passed with the best parameters derived from GridSearch. 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