## **DETAIL PROJECT REPORT (DPR)**

## **Cement strength Prediction**

Problem Statement:

The aim of the project is to build regression model to predict the Compressive strength of cement block (measured in Mega-Pascal, MPa) based on the quantities of different materials as features from the training data.

**Value of the Project:**

Compressive strength is an important properties of concrete and mortar to determine the strength of the binder (cement) meaning pressure handled by 1 m3 concrete block. It has a significant effect on the performance characteristics of the mixture and ensures the overall quality of the finished product.

Suppose a civil engineer has a job to build a structure, therefore he has to decide on the load of anticipation what should be the mixture of the materials to use to stick a balance.

* Durability – In order to handle a lot of pressure of the structure designed, the measure of the various materials needs to be adjusted to determine the compressive strength.
* Cost-efficient – The concrete materials should not be maximum strength because it might exceed the cost budget of the company.

Therefore the company has asked a software company to build a model to predict the strength for different materials used.

**A.I. STACK:**

**Technology:-**

|  |  |
| --- | --- |
| Interpreter | Anaconda Python3.6 |
| IDE: | Pycharm |
| Framework | Flask |
| Module | Numpy, pandas, sklearn |
| Database | SQLlite |
| Algorithms | KMeans, Random Forest, Linear Regression, Grid Search CV |
| Deployment | AWS, GCP, Azure |

**Environment:-**

A virtual conda environment is initiated with Python version 3.6for installing the required packages. It makes easy to ideally separate different applications and avoid problems with different dependencies. Using virtual environment we can switch between both applications easily and get them running.

The packages imported needs to be listed in file - requirements.txt.

pip freeze > requirements.txt

**Code Implementation:-**

All the codes are written in Python language using Pycharm Interpreter. This project is designed in modular customized approach where all the module files and packages having its own class object and methods. In building the Client app the packages are imported which is located inside the project directory.

**Data generation:-**

The data generated or acquired for development can be accessed by any of the following methods:

* Clients’ server.
* 3rd party API.
* Web Scrapping
* Cloud storage.
* Mobile Apps.
* IOT devices.

**Database:-**

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

* The User provides required data files information.
* The system stores each and every data given by the user or received on request to the SQLlite database. In order to solve any database used as the dataset are in csv file format.

**Logging:-**

* Issue or glitch in run time, logs are maintained in every activity by the system.
* The System is capable of identifying at what step logging required.
* The System should be able to log each and every system flow.
* Developers can choose logging methods and also choose database logging/ File logging as well.
* System should not be hung even after using so many loggings. The process is followed we can easily debug issues so logging is mandatory to do.

**Error Handling:-**

In Python, exceptions can be handled using a try statement. The critical operation which can raise an exception is placed inside the try clause. The code that handles the exceptions is written in the except clause.

Every customized methods and functions are written inside a try-except block. This is going to display the explanation if anything goes wrong. An error can be defined as anything that falls outside the normal and intended usage.

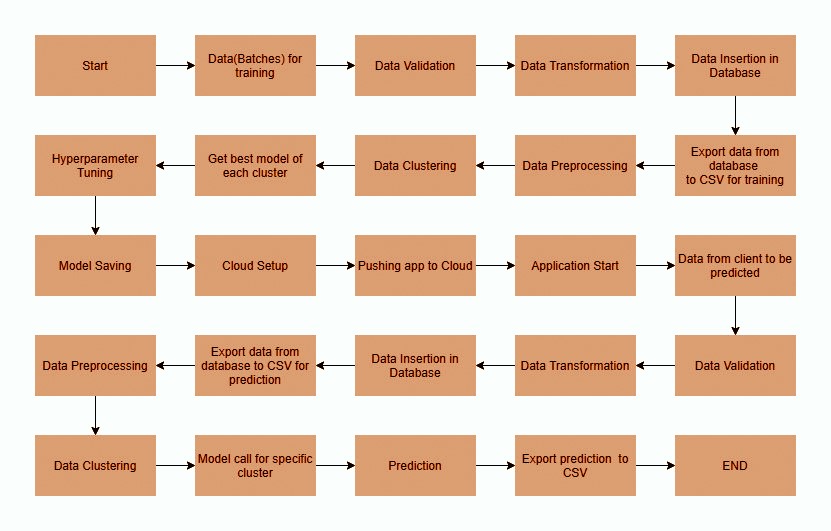
**Deployment:-**

A machine learning model can only begin to add value to an organization when that model’s insights routinely become available to the users for which it was built. The process of taking a trained ML model and making its predictions available to users or other systems is known as **deployment**. After successful running of the application we will deploy it in the following platforms:

* GCP
* AWS
* Azure

**System Architecture:**

**Workflow design:-**



**Data Sharing Agreement (DSA):-**

The training batch files, we also acquire as a "schema" file from the client, which will contains all the relevant information such as:

* Sample file name (ex cement\_strength\_08012020\_120021)
* Length of date stamp(8 digits)
* Length of time stamp(6 digits)
* Number of Columns
* Column names
* Column data type

**Data Info:-**

The concrete compressive strength is the regression problem. The listing represents the following information of the data:

* Name
* Data Type
* Measurement
* Description (Variable)

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Data Type | Measurement | Description (Variable) |
| Cement | quantitative | kg in a m3 mixture | ***Input*** |
| Blast Furnace Slag | quantitative | kg in a m3 mixture | ***Input*** -- Blast furnace slag is a nonmetallic coproduct produced in the process. It consists primarily of silicates, aluminosilicates, and calcium-alumina-silicates |
| Fly Ash | quantitative | kg in a m3 mixture | ***Input*** – it is a [coal combustion product](https://en.wikipedia.org/wiki/Coal_combustion_product) that is composed of the [particulates](https://en.wikipedia.org/wiki/Particulates) (fine particles of burned fuel) that are driven out of coal-fired [boilers](https://en.wikipedia.org/wiki/Boiler) together with the [flue gases](https://en.wikipedia.org/wiki/Flue_gas). |
| Water | quantitative | kg in a m3 mixture | ***Input*** |
| Super-plasticizer | quantitative | kg in a m3 mixture | ***Input***—Super-plasticizers (SP's), also known as high range water reducers, are additives used in making high strength concrete. Their addition to [concrete](https://en.wikipedia.org/wiki/Concrete) or [mortar](https://en.wikipedia.org/wiki/Mortar_(masonry)) allows the reduction of the water to cement ratio without negatively affecting the [workability](https://en.wikipedia.org/wiki/Workability) of the mixture, and enables the production of [self-consolidating concrete](https://en.wikipedia.org/wiki/Self-consolidating_concrete) and high performance concrete |
| Coarse Aggregate | quantitative | kg in a m3 mixture | ***Input*** -- construction aggregate, or simply "[aggregate](https://en.wikipedia.org/wiki/Aggregate_(composite))", is a broad category of coarse to medium grained particulate material used in [construction](https://en.wikipedia.org/wiki/Construction), including [sand](https://en.wikipedia.org/wiki/Sand), [gravel](https://en.wikipedia.org/wiki/Gravel), [crush stone](https://en.wikipedia.org/wiki/Crushed_stone), [slag](https://en.wikipedia.org/wiki/Slag), recycled concrete and geosynthetic aggregates |
| Fine Aggregate | quantitative | kg in a m3 mixture | ***Input***—Similar to coarse aggregate, the constitution is much finer. |
| Age | quantitative | Day (1~365) | ***Input*** |
| Concrete compressive strength | quantitative | MPa | ***Output*** |

***Data Training***

**Data Validation:-**

As per the DSA agreement, we will be maintain the nomenclature of the points and we are going to perform sets of validation on the given training batch files:

* ***Name*** - Based on the given name in the schema file we will validate the name of the batch files. Creating a regex pattern as per the name given in the schema file to use for validation. After validating the pattern in the name, we will proceed to check for the length of date in the file name as well as the length of time in the file name. If all the values are as per requirement, we move such files to *"Good\_Data\_Folder"* else we move such files to *"Bad\_Data\_Folder”.*
* ***Number of Columns*** – Validating number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is imported 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. This is validated when we insert the files into Database. If the data-type 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*".

**Database:**

In order to maintain a data redundancy we are going to use SQLlite to keep training data. SQLite has higher performance and store data in structured manner.

**Insert Data:**

* ***Database Creation and connection*** - Create a database with the given name passed. If the database is already created, open the connection to the database.
* ***Table creation in the database*** - Table with name *"Good\_Data",* is created in the database for inserting the batch files in the *"Good\_Data\_Folder"* based on given column names and data-type 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".*

**Export Data:**

The data in a stored database is exported as a CSV file to be used for further data- pre-processing and model training.

**Data Pre-processing:-**

The input variable are having quantitative data-type but to pass on the data for model training we need to perform some transformation:

* Checking for null values in the columns. If present, impute the null values using the KNN imputer.
* Transform the features using log transformation
* Scale the training data using standard scalar.

**Model Training:-**

**Clustering:**

* K-Means algorithm is used to create clusters in the pre-processed 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 K-means model is trained over pre-processed 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 Regressor" and “Linear Regression”.
* For each cluster, both the algorithms are passed with the best parameters using hyper-parameter tuning technique derived from Grid Search CV algorithm.
* We calculate the R-squared scores as an evaluation matrices 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.

***Test Prediction:***

**Data Validation:-**

Since we validated training data, we are perform different sets of validation sets on the given set of test files:

* ***Name*** - Based on the given name in the schema file we will validate the name of the batch files. Creating a regex pattern as per the name given in the schema file to use for validation. After validating the pattern in the name, we will proceed to check for the length of date in the file name as well as the length of time in the file name. If all the values are as per requirement, we move such files to *"Good\_Data\_Folder"* else we move such files to *"Bad\_Data\_Folder”.*
* ***Number of Columns*** – Validating number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is imported 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. This is validated when we insert the files into Database. If the data-type 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*".

**Test Database:**

**Insert Data:**

* ***Database Creation and connection*** - Create a database with the given name passed. If the database is already created, open the connection to the database.
* ***Table creation in the database*** - Table with name *"Good\_Data",* is created in the database for inserting the batch files in the *"Good\_Data\_Folder"* based on given column names and data-type in the schema file.
* ***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".*

**Export Data:**

The test data in a stored database is exported as a CSV file to be used for further data pre-processing and prediction.

**Data Pre-processing:-**

Test variable are having quantitative data-type but to pass on the data for model training we need to perform some transformation:

* Checking for null values in the columns. If present, impute the null values using the KNN imputer.
* Transform the features using log transformation
* Scale the training data using standard scalar.

**Prediction:-**

* ***Clustering*** ­­­-- K-Means model created during training is loaded, and clusters for the pre-processed prediction data is predicted.
* ***Prediction*** -- Based on the cluster number, the respective model is loaded and the system will predict the data for that cluster.
* ***Saving*** -- Once the prediction is made for all the clusters, the predictions along with the original names before label encoder are saved in a CSV file and shared.

**Deployment:-**

After testing, we will move forward for production. Deployment will be done for the main.py which is the entry point where flask server’s starts and models along with required dependencies into cloud services like AWS, GCP etc.

**Conclusion:**

The design of the project is most important for scalability factor. The main focus was to make an automated system by building customize modular packages. Each package files has its own class object and methods and methods written for better functionality. To get the best predicted results we can use deep-learning models and other clustering models to train the training data for better performance

**Cement strength Prediction Project Q/A:**

*Q1.* Tell me about your current project.

*Answer:* The aim of the project is to build regression model to predict the Compressive strength of cement block (measured in Mega-Pascal, MPa) based on the quantities of different materials. Compressive strength is an important properties of concrete and mortar to determine the strength of the binder (cement) meaning pressure handled by 1 m3 concrete block.

The training files are received as batch files for model training. We are following a customize architecture pipeline where the data follows the pipeline to get trained from data validation to model training. Database is maintained to preserve the records received from the client which ca be used in future training. As this is regression based project we are using Randon Forrest Regressor, Linear Regression models and K-Means model for clustering.

In case of prediction the test batch files are imported from stored database. After validation and pre-processing the test files are pass through the save models to give prediction. For production we have deployed the models in cloud services.

*Q2.* What was the size of the data?

*Answer:* The training are in batch files with total size of 130kb.

*Q3.* What was the data type?

*Answer:* The dataset received was in a csv file format where all the variable were quantitative.

*Q4.* What was the team size and distribution?

*Answer:*

*Q5*. What Hadoop distribution were you using?

*Answer:*

*Q6*.What is the version of distribution?

*Answer:*

Q7.What was the size of the cluster?

*Answer:*

*Q8.* How many nodes were there in all the Dev, UAT, and Prod environments?

*Answer:*

*Q9.* How were you creating and maintaining the logs?

*Answer:* The logs was created along with all the methods and functions written in the code. As the application starts to run, logs are maintained inside the local directory but can be stored in database.

*Q10.* What techniques were you using for data pre-processing for various data science use cases and visualization?

*Answer:* There are multiple steps that we do for data pre-processing, like data cleaning, data integration, data scaling, etc. In case of Machine Learning some of them are listed as follows:

* While preparing data for a model, data should be verified using multiple tables or files to ensure data integrity.
* Identifying and removing unnecessary attributes.
* Identifying, filling or dropping the rows/columns containing missing values based on the requirement.
* Identifying and removing outliers.
* Based on the requirement, form clusters of data to avoid an over-fitted model.
* Scaling the data so that the difference between the magnitudes of the data points in different columns are not very big.
* Converting the categorical data into numerical data.
* Replacing or combining two or more attributes to generate a new attribute which serves the same purpose.
* Trying out dimensionality reduction techniques like PCA(Principal Component Analysis), which tries to represent the same information but in a space with reduced dimensions.

*Q11.* How were you maintaining the failure cases?

*Answer:* When the application is on run and there is an error in the response from local server, it is going to raise an exception and print the failure logs. The particular criteria can be updated the application can run again.

*Q12.* What kind of automation have you done for data processing?

*Answer:* We are maintain an ETL pipeline where the set of processes used to move data from a source or multiple sources into a database. The dataset can be easily extracted from database after the data-pre-processing is done for further model training.

*Q13.* Have you used any scheduler?

*Answer:*

*Q14*. How are you monitoring your job?

*Answer:*

*Q15.* What were your roles and responsibilities in the project?

*Answer:* My responsibilities consisted of gathering the dataset, labelling the text data for the model training, training the model on the prepared dataset, deploying the trained model to the cloud, monitoring the deployed model for any issues.

*Q16.*What was your day to day task?

*Answer:*

*Q17.* In which area you have contributed the most?

*Answer:* I contributed the most to data labelling and model training areas. Also, we did a lot of brainstorming for finding and selecting the best algorithms for our use cases. After that, we identified and finalized the best practices for implementation.

*Q18.* In which technology you are most comfortable?

*Answer:* I have worked in fields viz. Machine Learning, Deep Learning, and Natural Language Processing, Automation and I have nearly equivalent knowledge in these fields. But if you talk about personal preference, I have loved working in Deep Learning and NLP the most.

*Q19.* How you rate yourself in big data technology?

*Answer:*

Q20. In how many projects you have already worked?

*Answer:* I have worked in various small and large scale projects, e.g. Machine learning regression, and classification problems. NLP projects, chatbot building.

*Q21.* How were you doing deployment?

*Answer:* The mechanism of deployment depends on the client's requirement. For example, some clients want their models to be deployed in the cloud, and the real-time calls they take place from one cloud application to another. On the other hand, some clients want an on-premise deployment, and then they do API calls to the model. Generally, a model file is prepared first and then try to expose it through an API for predictions/classifications. The mechanism in which he API gets called depends on the client requirement.

*Q22*. What kind of challenges have you faced during the project?

*Answer:* The biggest challenge that we face is in terms of obtaining a good dataset, cleaning it to be fit for feeding it to a model, and then labelling the prepared datasets. Then comes the task of finding the correct algorithm to be used for that business case. Then that model is optimized. If we are exposing the model as an API, then we need to work on the SLA for the API as well, so that it responds in optimum time.

*Q23.* What will be your expectations?

*Answer:* It’s said that the best learning is what we learn on the job with experience. I expect to work on new projects which require a broad set of skills so that I can hone my existing skills and learn new things simultaneously. Also I’m looking for my professional and personal growth.

*Q24.* What is your future objective?

*Answer:* The field of data science is continuously changing. Almost daily, there is a research paper that changes the way we approach an AI problem. So, it really makes it exciting to work on things that are new to the entire world. My objective is to learn new things as fast as possible and try and implement that knowledge to the work that we do for better code, robust application and in turn, a better user/customer experience.

*Q25.* Why are you leaving your current organization?

*Answer:* I was working on similar kinds of projects for some time now. But the market is rapidly changing, and the skill set required to be relevant in the market is changing as well. The reason for searching a new job is to work on several kinds of projects and improve my skill set. *<Mention about the company profile and if you have the project name that you are being interviewed for as new learning opportunities for you>.*

*Q26.* How did you do Data validation?

*Answer:* Validation of data is done as per the nomenclature of DSA with the client. Performing sets of validation on the given training and test batch files where the data following the points are moving *“Good data folder”.* The data which are not following the traditional aspects are moved to *“Bad data folder”.* The correct aspects based on following validation are:

* Name.
* Number of columns.
* Name of the columns.
* Data-type of the columns.
* Null values in columns.

*Q27.* How did you do Data enrichment?

Answer: Data enrichment is the process of acquiring second- or third-party data, and using it to augment or cleanse your first-party data. In in project we have collected from 3rd party API and requested the client to present other set of batch files.

*Q28.* How would you rate yourself in machine learning?

*Answer:* Well, honestly, my 10 and your 10 will be a lot different as we have different kinds of experiences. On my scale of 1 to 10, I’ll rate myself as an 8.

*Q29.* How would you rate yourself in distributed computation?

*Answer:* I’d rate myself a 7 out of 10.

*Q30.* What are the areas of machine learning algorithms that you already have explored?

*Answer:* I have explored various machine learning algorithms like Linear Regression, Logistic Regression, L1 and L2 Regression, Polynomial Regression, Multi Linear Regression,Decision Trees, Random Forests, Extra Trees Classifier, PCA, XG Boost , ADA Boost, Gradient Boosting, Light Boost, K-Means, K-Means ++, KNN, SVM, SVR, Naïve Bayes, Agglomerative clustering, DBScan, Hierarchical clustering, TFIDF, Word to Vec, Bag of words, Doc to Vec, Kernel Density Estimation are some of them.

*Q31.* In which part of machine learning have you already worked on?

*Answer:* I have worked on both supervised and unsupervised machine learning approaches and building different models using the as per the user requirement.

*Q32.* How did you optimize your solution?

*Answer:* Well, model optimization depends on a lot of factors.

* Train with better data or do data pre-processing steps more efficiently.
* Increase the quantity of data used for training.
* Using different module to get clean data
* Changing models to get better predicted results.

*Q33.* How much time did your model take to get trained?

*Answer:* The training I took was very small so with respect to my system configuration it took less a minute.

*Q34.* At what frequency are you retraining and updating your model?

*Answer:*

*Q35.* In which mode have you deployed your model?

*Answer:* I have deployed the model both in cloud environments as well in the on premise ones based on the client and project requirements.