Exploratory Data Analytics

What is EDA?

EDA means Exploratory Data Analysis. The purpose of data analysis is to explore. Exploration means try to understand what kind of data I have in my hand. Using EDA we try to get the answer to the following questions.

- What kind of data is this? (file format, volume of data, number of columns, metadata data of image/video/audio or some feedback in English or other languages, or tabular data, etc)
- How complex is this data? (How many files are there? primary key? how these files are connected to each other? is nested data in some files? is some field having nested data, etc.)
- Is this data sufficient for meeting our ultimate goal, i.e. Model building?
- Is there any missing data? Data needed but not given by the business or not available at all or costly to get that data etc.
- Are there any missing values? In the given dataset do we have complete information or some values are missing for some records or some columns?
- What are different independent and dependent fields?
- Is there any relationship between different independent variables of the dataset? If yes then how strong is that relationship?
- Are observations independent or tightly coupled like we see in time-series data?

In the data scient project lifecycle, EDA is not a sequential, one-time, isolated process. Till the time data is not ready for modeling we keep doing EDA and cleaning the data. So, EDA is followed by a list of decisions taken to clean the dataset, and finally, data cleaning steps are implemented. If the dataset is not in the good shape after the first iteration of EDA we continue EDA in the next cycle. In this article, I am not referring to

EDA as just visualizing and understanding the dataset but all the steps required till the dataset is not ready for modeling.

Importance of EDA

Any data scientist can tell you the importance of a clean dataset. In the absence of a clean and balanced dataset expecting a good AI model is like expecting water in the desert. It looks it is there but it is never there. Just having lots of data is not exciting enough for creating a model. We, data scientists, are in love with data, playing with different kinds of data, extracting data from different sources, combining data so that it starts making sense, cleaning the data so that visualizing patterns is easy, creating features so that a robust model can be created is a thrill for us. As much complex, a dataset is that much exciting is the journey of creating a clean dataset.

When people enter into the data science field they get some training on the tool and about the capabilities of those tools. But in the absence of an overall approach and process, they develop a bad model without realizing their EDA and Feature Engineering work is not solid enough. We all know an age-old saying that a fool with a tool is still a fool. What is the use of learning all the tools if we cannot use that tool properly? The processes are as important as the tool is. From my experience, I am describing below a high-level process that will help you doing good EDA and creating a clean and balance dataset along with feature engineering.

I am not diving into the steps required before the dataset creation. There is different complexity there and that is more focused on the data engineering side. Therefore in this article keeping that outside. Example steps given below are just sample and do not think it is a final or exhaustive list. Depending upon your dataset, project objective, tools, and hardware resources available you need to add or remove some of the steps I am describing below.

Note: Application programmer, DBA, Database developer calls “fields” and but in data science, the same thing is referred to as a column, feature. I may use both of these terms, therefore do not assume these are different things. Similarly “record” in data science/ statistics is referred to as observation, record, sample. I may use all these terms therefore do not get confused around these terms.

EDA Stage 1: Basic Exploration

Understanding Dataset and Datatypes: EDA means Exploratory Data Analysis. The purpose of this step is to understand the dataset at a high level. We need to understand the file format of data files, volume, character set, etc. I look at basic exploration as following 3 steps.

EDA Stage 1.1: Understanding the dataset as a file.

- How big is a file?
- How many files are there?
- Is there any data dictionary available?
- What kind of file format (CSV, Excel, JSON, text, etc)?
- Is it tabular data or image data with pixel values?

- Is it a text file containing customer feedback, describing products, or short stories?
- How to load in the memory? Sometimes files are in GB or TB. Other times files are in zip format, etc.
- What Charset is used (sometimes data may be in different languages like Hindi, Tamil, French, Chinese, Japanese, etc)?
- How many records, how many columns are there in the dataset?
- Is a data dictionary needed? If needed and not provided can we create a data dictionary or do we need to request from the data provider?

EDA Stage 1.2: Understanding fields/columns

- Are column names given making sense? Sometimes they are just named as 1,2,3 etc.?
- Are column names proper? Sometimes all upper letters, all lower letters, unnecessary space, special character, etc.
- What are column technical data type numeric/bool/text/float/object/date/datetime etc?
- What are column business data types nominal/ordinal/continuous?
- What is the content of columns? Date format? Is number with currency code? Decimal is coma or dot etc?
- Apply business common sense and look for more interesting columns. Are they present in the dataset? If you expect something which is not there then make a note of that.
- Missing value in any column? Which column? What are you expecting there? How much % are missing values?
- Sometimes data given to you is coming from a data entry form that has default values like select, none, all, etc. While filling the form if a user does not select anything then these values go into the database directly. In fact, they are null values. We need to treat them properly.

EDA Stage 1.3: Look for the target field.

- What kind of problem. Classification, Clustering, Regression? Supervised/ unsupervised?
- If supervised classification problem then the dataset is balanced or not? If the dataset is not balanced then we may need to balance it by using some sampling techniques.
- If supervised regression problem then the distribution of target column is normal or heavily tailed? If the distribution is not normal then we may need to make it normal by using some transformation.
- Any missing data in the target column. If the target value is missing then we need to handle it properly.
- Target column is a single value or a vector?

EDA Stage 2: Univariate, Bivariate & Multivariate Analysis

EDA Stage 2.1: Univariate: Check the distribution of each field.

- In the case of a continuous column, a column has normal distribution or not? For example, for the sales column, you can plot a histogram and check whether the average sale is a median sale or not? Let us assume a dataset has 10K sales transaction records and the average sale is $500 then it does not mean 5000 transactions are less

than $500 (average sale) and the remaining 5000 transactions are more than $500. If a column is not normally distributed then we need to do some transformation.
- In the case of ordinal values, a column has uniform distribution or not? For example, if a column is "education" then plot a bar chart and see whether all education groups are equally represented or most of the people in the dataset are illiterate people or something else.
- If data is heavily tailed? If yes then which side? For example, if your column has the GDP of nations then this distribution will be heavily right-tailed. I guess more than 80% of countries will have one high peak and the rest will tail long on the right side.
- If data skewed? which side, left or right? What is the business meaning of this skewness? For example, if the distribution of the age column is right-skewed it means the dataset has more representation from young people. Keep in mind that skewed data may not be tailed.
- How many outlier values? Can you put them in a percentage like .001%?

EDA Stage 2.2: Bivariate Analysis: Relationship between each column and target field.

- Check how each field is connected to the target field. Random (no relation), positive (increase with the increase of column value), negative (decreases with the increase of column value)
- Check how strong is relationship. This will give you an idea, which variable contributes more to the prediction of the target variable.

EDA Stage 2.3: Bivariate Analysis: Check co-linearity of all fields.

- We know income has a relationship with expense, wealth has a relationship with comfort, blood pressure has a relationship with anxiety, etc. If you check these independent variables they move heavily with each other. This is called co-linearity. Co-linear variables are not good for the model. Identify which pair of variables have high co-linearity. Note: In a pair of high co-linear features you need to drop one.
- Make a list of which variables can be dropped due to high co-linearity

EDA Stage 2.3: Multivariate Analysis:

- Let us say one independent variable is education and it has 10 categories. Generally, we know education and income have a relationship. But this relationship is not equally strong for all education levels. For example, if you check the distribution of income columns for uneducated people you will find a different kind of relationship (outliers will be there, maybe some uneducated criminal or uneducated politician has huge wealth or some uneducated lucky person got some lottery and started his own business). So this distribution has different average, median, and SD. But, if you check the income distribution of Ph.D. holders that distribution will have different average, median, and SD (You will observe outliers here as well, for example, some Ph.D. holders without a job because of many reasons).
- Analyze how these different categories of an independent variable will impact the prediction ability of your model.

The decision for Data Cleaning

By this time you have not done any data cleaning. But you have understood what kind of cleaning is needed, what needs to be cleaned etc. In this stage, you take a decision on what needs to be cleaned, and how that cleaning can be performed.

1. **Decision 1.1: Take the Decision which field is important and which is not.** For example, some columns may have 80% null value columns some have 10% null values. Take a decision, which columns to be removed completely. Sometimes 99% values in a column are the same. So mark those for removal.
2. **Decision 1.2: How to treat null values**: Null value doesn't mean only these are null values in the column. Sometimes default values of a form are pushed into a dataset can be the null value. For example 'select', ' ', 'pickup' etc.
3. **Decision 1.3: Null value Records:** For example, some records may have 80% null value columns some have 10% null values. Take a decision which records should be removed completely.
4. **Decision 1.4: Null value of Continuous Column:** If some column has continuous and some values are missing then how will you fill the null values. In this case, mostly you will choose the mean of that column. But sometimes you can take a different strategy to fill this numeric null value field.
5. **Decision 1.5: Null value of Categorical Column:** If some column has categorical values, say quality grade or education, etc then how will you fill the null value for a record? In most cases, you can use the median value of the column. But you can use a different strategy.
6. **Decision 1.7:** Sometimes null values can be filled with mode values. Is there any column that needs this treatment?
7. **Decision 1.8: Usage of Tool:** You want to do the null value imputations manually or you want to use tools like sklearn.simpleimputer, sklearn.iterativeimputer, sklearn.KNNimputer etc.
8. **Decision 1.9: Usage of Model:** Sometimes you create a basic linear regression model using non-null value records and columns. The target column for this model training can be a field that you want to impute and you can use this for null value imputation for a particular column. This way you need to create as many models as many null value columns are there in the dataset. This approach works if any record has a maximum of one null value column. For the ordinal value column, you can use algorithms such as logistic regression, KNN, etc.

The decision to Address Data Imbalance Issue

1. **Decision 2.1:** Oversampling: Generate records for a class that is under-sampled. This makes sense when an under-sampled class is very precious and too little data is available for this class. For example, a bank transaction dataset, which has fraud and non-fraud transactions. Fraud transactions are extremely less and very precious data.
2. **Decision 2.2:** Undersampling: Remove records of a class that is over-sampled. This makes sense when you have lots of data and you can afford to through away some without any risk.
3. **Decision 2.3:** If over sample then which technique ROS, SMOTE, etc.

Data Cleaning

By the time you reach here, you have some action plan to clean the dataset. Now, perform the following actions as per the decision taken in previous steps. Make a note, by the time you finish this step data still, may not be normally distributed, and in future steps, we may need to do feature engineering.

- Null value handling.
- Outlier Handling.
- Target Imbalance Handling.
- Independent Variable Imbalance Handling.

EDA Stage 3

After cleaning the dataset if you perform the following steps you will find the dataset is balanced and clean. But if you still observe some imbalance or distribution issue then you can take a call and perform previous steps once again on this clean dataset. Perform these steps to know whether you should move ahead or once again iterate the data cleaning cycle.

- Data distribution of each field. – Univariate
- Distribution of each continuous data field for target field (continuous) – Univariate
- Distribution of each continuous data field for each value of the target field (categorical). – Multivariate
- Distribution of each categorical data field for target field (continuous). – Multivariate
- Distribution of each Categorical data field for each value of target field (categorical). – Multivariate
- Check co-linearity of all fields. – Bivariate
- Oversample – smote, adasyn, random oversampler

Feature Engineering

By the time you reach here, you will notice some of the continuous fields still do not have a normal distribution. So now you need to visualize what kind of transformation will turn that field into a normal distribution. There are chances you need some other features which are not there in the dataset. For example, you have longitude and latitude information but you want to calculate the distance between taxi pick up and drop points. You can create a distance feature. In this process, you may need to drop some existing fields. You need to perform the following steps.

1. Based on domain expertise you can create some useful features. For example create distance from longitude, latitude. Create speed from distance and time. Create throughput from requests and time etc.
2. If a certain column is not normally distributed then based on data science experience you can create some useful features. Use log, exp, power, inverse math functions with various base numbers. You can remove the original variable. A final variable should be normally distributed. You can also use boxcox.

EDA Stage 4

By the time you reach this stage, your dataset must be completely ready for modeling. But how will you know that? Perform one more round of EDA. Follow the steps mentioned earlier. If you still find some glitch then perform cleaning and feature engineering steps as mentioned earlier.

EDA Complete

At this stage EDA is complete and we can use our dataset for model training. Future steps are not related to EDA but in order to visualize the complete lifecycle, I am explaining them in short so that aspiring data scientists can visualize the complete project life cycle.

Creating Dataset for Training

When a dataset is ready we can use it for training. But if we use a complete dataset for the training then it is impossible for us to know how the model performs on the unseen data. After training, if you evaluate the model performance using a training dataset then it is like checking the memorization ability of a child. There are some children who can memorize everything but that is not called learning. The real test of learning is when you are able to apply what you have learned to unseen data, situations, etc. To achieve this we need to split our dataset into train and test datasets. Sometimes test dataset is given to us by the customer and it is used for UAT. In that case, our split dataset is called train and validation datasets and not train, test datasets. We need to follow these steps to make the training dataset ready for training purposes.

- Decide how much data is needed for training. Generally, 70% to 80% of data is used for the training, and the remaining data is used for testing.
- Create Train, Test /Validation Dataset.
- Scale all columns of the Train dataset. There are many techniques of scaling but most techniques are standard scaler or min-max scaler.
- Save the scaler and use it to fit the test/validation data just before your want to test the performance of your newly created model.

Modeling

This is the place where we create the main juice. This is the place for which all data preparation was done. For different kinds of problem statements, there are different kinds of algorithms. We have hundreds of algorithms. These algorithms are developed by various researchers and most of them are freely available for our implementation. Linear Regression, Logistic Regression, KMean, KNN, Random Forest, Decision Tree are just a few names. On the other side, we have deep learning-based architecture, and data scientists need to create a neural network to develop different models. During model creation, we need to develop different models with different sets of hyperparameters. For model development, we need the following steps.

- Select an algorithm and create multiple models with that algorithm and a different set of hyperparameters.
- Create multiple models with multiple algorithm and multiple set of hyperparameters.
- Establish metrics for model performance. Model performance metrics can be r2, rmse, accuracy, recall, precision, roc-auc, error rate, etc.

- Check the model performance on the train dataset.

Model Evaluation

In this step, we evaluate various models created against the test or validation dataset. Generally, we follow these steps for model evaluation.

- Transform the data as was done before training. Use the same transformer you created for training data transformation.
- Scale the validation/test dataset from the scale created earlier. (Transformation and scaling on test/validation dataset are done just before the testing the model.)
- Predict the results using the model and test/validation data.
- Evaluate the performance of the model using the metrics selected.

Model Selection

In this step, we select a model for final deployment. Metrics, which we get in the previous step are definitely the main influencer in model selection but we also need to keep in mind the size of the model, hardware available, performance of the model in terms of time, other processing required for prediction, for example, transformation, scaling, etc. To select a final model for deployment we need to follow these steps.

- Prepare an excel sheet. Write the name of each model along with hyperparameters and all the parameters which we will use for selecting model selection.
- Complete this excel sheet using the performance data of each model.
- Assign a weight to different parameters and get a final score of each model. For example, the time taken to predict may have higher weightage compare to the weightage of the accuracy of the model.
- Schedule a meeting with the sponsor and present him/her performance of different models, and time-based performance. Also discuss with them resources required to run and monitor the model.
- If explainablity is another criterion then convince yourself how you will explain the prediction of the model to the customer.
- Convince management and yourself and select the best model for deployment.

Model Deployment

For model development, we used various libraries, programming languages, and a particular operating system. In the production environment before we put our model, we need to make sure that any update to these libraries (because of other models on the production environment) or update in OS patches should not lead to model failure. To avoid this we need to dockerize our model with all the packages and then put this docker on the public or private app server. Based on the business and security requirement you enable certain ports and IP addresses via which our model can be consumed.

Exposing Model via User Interface

To consume the above model user must enter some data which can be input for the model. For the data input and model output display, we need to create a user interface. This user interface will call the above model using the service name. The service is available on a certain IP address and port number. Service will return results and our UI should show the result to the customer. This work is done by the web development team. They develop a web app and deploy that on a webserver.

Model Monitoring

When a model is serving users all the data input from the customer is captured in the system. The data recorded and used for model monitoring are following but not limited to this.

- Input parameters (given by the user)
- Model prediction (given by service)
- Time-taken to predict (recorded by the service)
- The time period of the day when requests come
- The IP address from where the request coming
- The customer is happy about the prediction (feedback captured on the UI screen)
- etc.

All the above data is used for performance evaluation and model retraining or retiring a model.

Summary

In this article, we learned.

- What is EDA?
- The iterative nature of EDA and Data cleaning
- Importance of feature scaling
- Importance of feature engineering
- Different steps are required for data cleaning and imbalance handling.
- What are different stages of the data science project life cycle