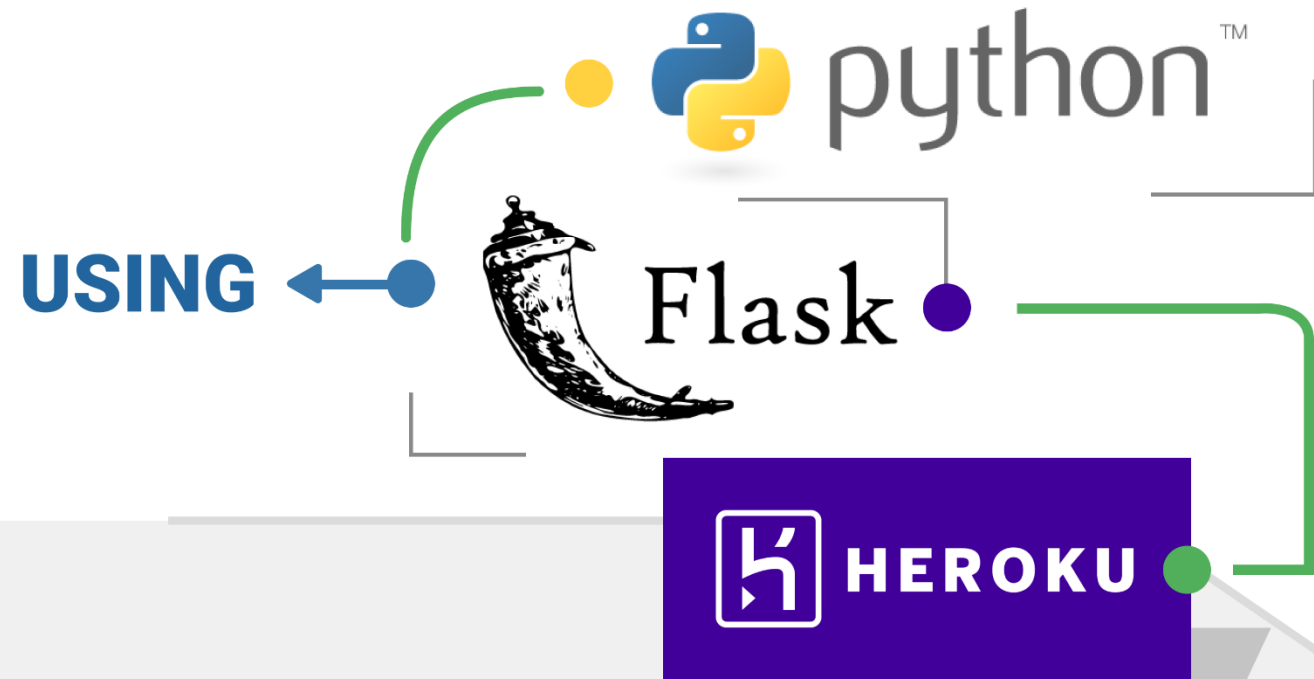
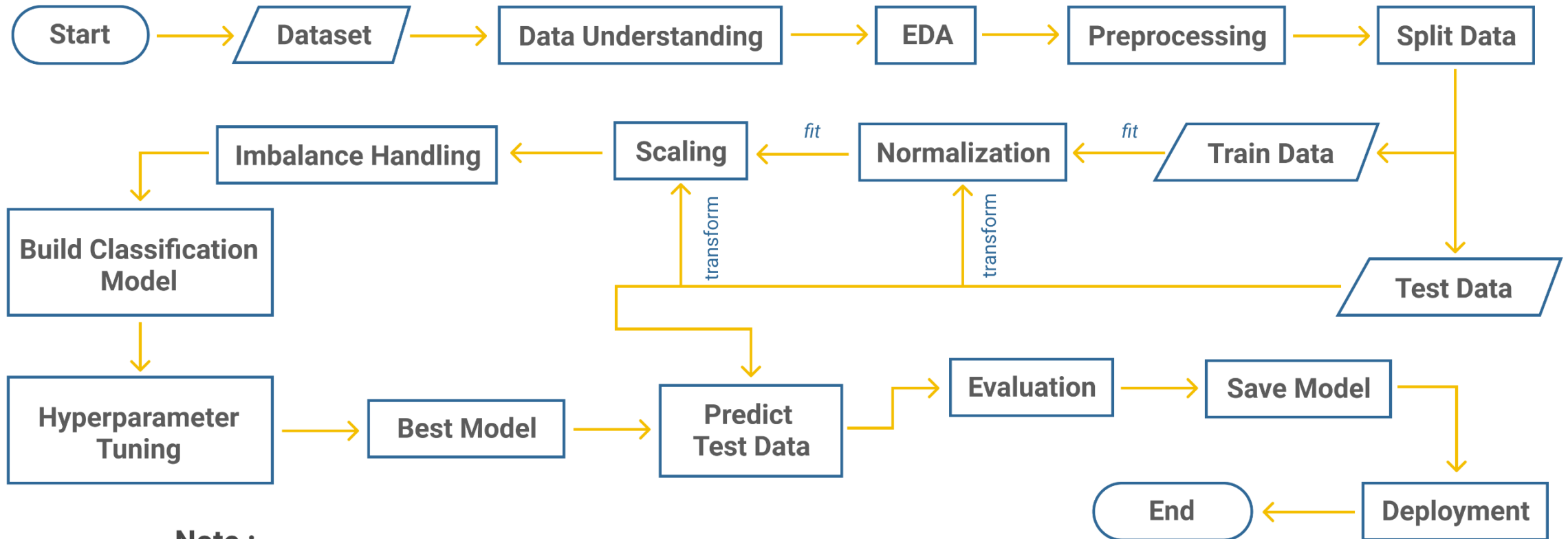


# Build Our First End-to-End Machine Learning Project



# Build Machine Learning Model



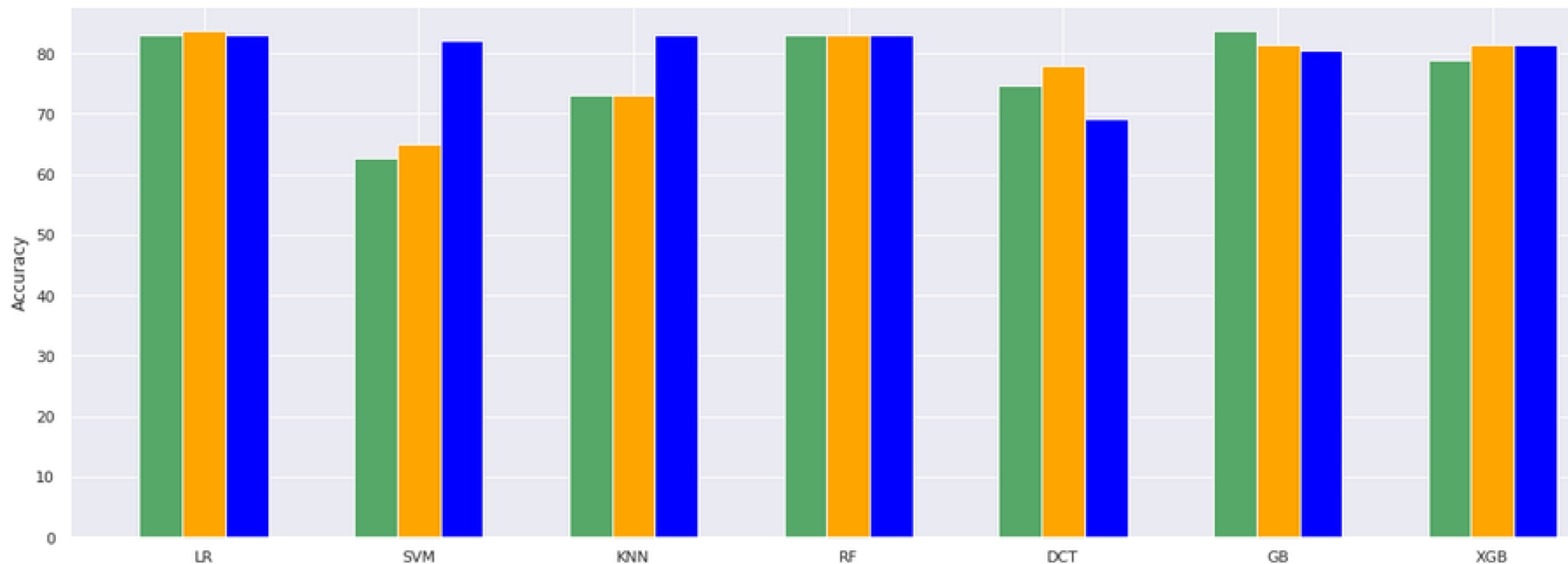
## Note :

*This is my model workflow, but it doesn't mean I apply each step to my final model. I tried some of scenarios to get the best model accuracy. For example, which one is better to use normalization and scaling or without both of it. I also tried to use several classification methods such as **Logistic Regression, SVM, KNN, Random Forest, Decision Tree, Gradient Boosting and XGBoost**. Let's see the comparison of accuracy values between the classifiers on the next slide.*

# Model Summary [1]

- ① **Data Understanding**, try to get information from the dataset such as summary statistics, data info, unnecessary feature, missing value, etc.
- ② **Exploratory Data Analysis (EDA)**, gain insights from the data by visualizing it.
- ③ **Preprocessing**, is step in which the data gets encoded, scaling, transformed, or to bring it to such a state that now the machine can easily parse it. The out put from this process is all the features value in number format.
- ④ **Split Data**, this process will separate the data into train and test. The train data will be used to build the model while test data will be used to check the model performance. **Why do we split the dataset before implementing normalization and scaling? The goal is to prevent the data leakage.** Data lakage is when information from outside the training dataset is used to create the model (it includes normalization and scaling model).
- ⑤ **Normalization**, makes the feature normally distributed. We can use PowerTransformer with method yeo-jhonson, box-cox, etc. **Scaling**, to normalize the range of feature value from the of data. We can use MinMax, standardscaler, etc.
- ⑥ **Imbalance Class Handling**, makes the ratio of each class (target) tobe equal. You can try using undersampling, oversampling or SMOTE.
- ⑦ **Build Model and Tuning it**, by tuning the parameters for each classifier. We will get the best classifier performace with its parameter.
- ⑧ **Save Mode**, by saving the model, we can use that model without retraining it. Usually the preprocessing result (normalization or scaling model) and classifier model are stored in the pickle format.
- ⑨ **Model Deployment**, in this project I tried to deploy my machine learning model on heroku server. I have used HTML5, CSS3 + Bootstrap and JS to create the user interface. While for the API, I built it using flask.

# Model Summary [2]



## Legend :

- Dataset with One Hot Encoding without drop first.
- Dataset with One Hot Encoding and drop first.
- Dataset with One Hot Encoding, drop first, scaling, and normalization.

## Conclusion :

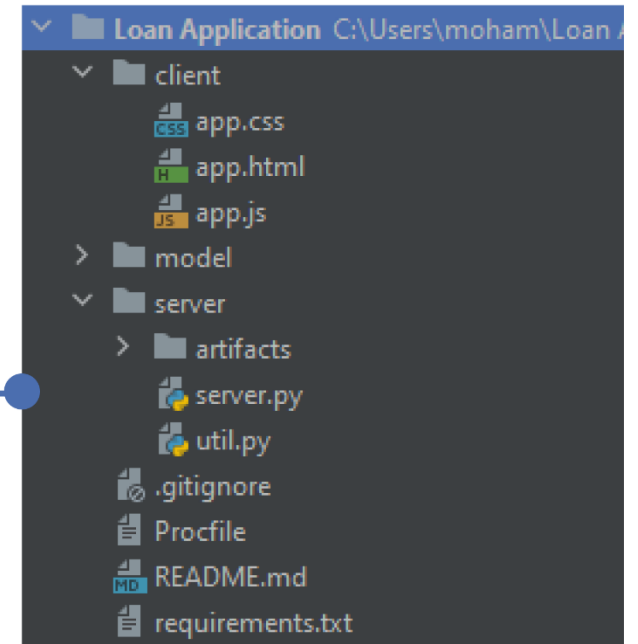
- Three models are overfitting such as XGBosst, Decision Tree and Random Forest.
- Some of models have low accuracy such as SVM and KNN.
- The best model accuracy produce by **Logistic Regression** ■ with accuracy 83.37% (submitted datathon : 87,15%).

[Visit the documentation on my github](#)

# Deploy Machine Learning Model



## Project Files



## How it works?

**Response 1**, when user accesses `website_link`, the server will send the responses such as the website interface and APIs (dropdown and classifier model) accessed at that page. Now, user can fill out the form and then press submit button to get the prediction results from the model.

**Request 1**, the data sent by the user will be preprocessed and then predicted by the model by hitting the API model.

After the API has finished predicting the data, it will return the result in json format.

**Response 2**, the results of the model will be returned to the user interface (accepted or rejected).

- **Client Folder**  
Contains files to create the frontend view (HTML, CSS and JS).
- **Model Folder**, it is used to store any scripts related to the database and put our pickle model.
- **Server Folder**, **serve.py** is used to set the website routing, while **util.py** is used for the logic from our API.
- **Procfile**, it will show the server which file should start at the first time. **Requirements.txt**, it will ask the server to install all libraries used in this project.



# Website Interface

Loan Approval

Property Area: rural

Gender: ☐ Male ☒ Female

Married: ☒ No ☐ Yes

Self Emp: ☐ No ☒ Yes

Education: Graduate

Dependents: 2

Ap. Income: 6000

C.Ap. Income: 4500

Loan Amount: 320

Loan Am.Trm: 30

Credit Hist: 1

[Reset Result](#) [Submit Loan](#)

**Loan Approved**

## Note:

Before you deploy your model on heroku server, make sure that You have changed the local settings to server and also added **Procfile** and **requirements.txt**.

## Data Information

|               |  |
|---------------|--|
| Property Area | The type of location where the applicant's property lies |
| Gender        | Gender of the applicant                                  |
| Married       | The marital status of the applicant                      |
| Self Emp      | If the applicant is self-employed or not                 |
| Education     | Education level of the applicant                         |
| Dependents    | Noumber of dependent on the applicant                    |
| Ap. Income    | The amount of income the applicant earns                 |
| C.Ap. Income  | The amount of income the co-applicant earns              |
| Loan Am       | he amount of loan the applicant has requested for        |
| Loan Am.Trm   | The number of days over which the loan will be paid      |
| Credit Hist   | A record of a borrower's responsible repayment of debts  |

Website url:

<https://dphi-loan-predict.herokuapp.com/>



## CERTIFICATE OF ACCOMPLISHMENT

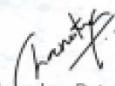


**Irwan Afandi**

has successfully completed

## **Machine Learning Advanced Bootcamp**

The bootcamp consisted of 20 days of coursework, live sessions from industry experts, quizzes, and projects/datathons on real-world datasets.



Chakraborty Patnaik  
Co-founder, DPhi

Issued on Jan. 7, 2021

Authenticity of the certificate can be verified here:

<https://dphi.tech/bootcamps/certificate/verify/347b6bdb-d82a-4575-8113-6b02334fd195>