CS550 - Machine Learning and Business Intelligence

End-to-End Machine Learning Project

By : Prince Rajasekaran
Instructor: Dr. Chang, Henry

Table of content

- **Introduction**
- **Theory**
- **Implementation**
- **Conclusion**
- **References**

Introduction

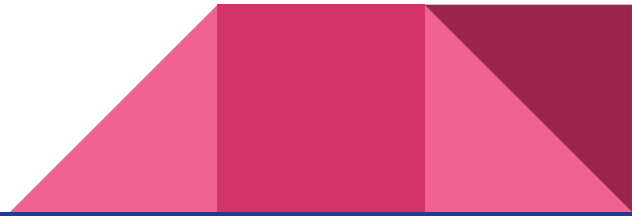
An end-to-end machine learning project entails creating a comprehensive pipeline that uses raw data as input, trains a machine learning model on it, and then applies the trained model to new, unforeseen data to make predictions. These projects typically consist of several steps, including model selection and training, feature engineering, data collection and preprocessing, and deployment of the trained model.

A project that uses end-to-end machine learning aims to produce a trustworthy and accurate model that can be applied to a particular issue, such as image classification, natural language processing, or predictive analytics. This entails not only choosing the right algorithms and methods for the job at hand but also comprehending the demands and constraints that are unique to the domain and maximizing the performance of the model.

Theory

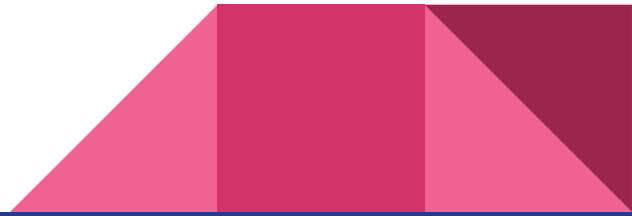
7 steps involved in Machine learning :

1. **Data collection**
2. **Data preparation**
3. **Choose a model**
4. **Train the model**
5. **Evaluate the model**
6. **Parameter tuning**
7. **Make predictions**



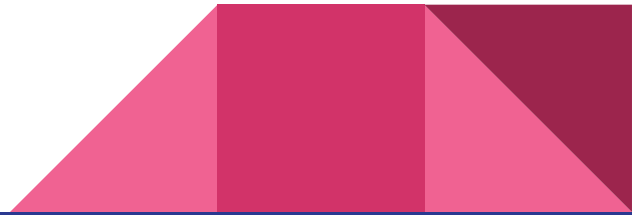
1. Data collection

- The quantity & quality of your data dictate how accurate our model is
- The outcome of this step is generally a representation of data which we will use for training
- Using pre-collected data



2. Data preparation

- Wrangle data and prepare it for training
- Clean that which may require it Ex : remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.
- Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data
- Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis
- Split into training and evaluation sets



3. Choose a model

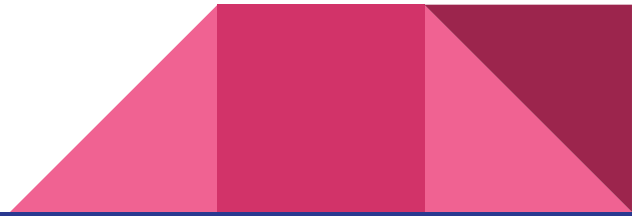
Different algorithms are for different tasks; choose the right one

Example :

1. **Linear Regression**
2. **Logistic Regression**
3. **Decision Tree**
4. **SVM (Support Vector Machine)**
5. **Naive Bayes**
6. **kNN (k- Nearest Neighbors)**
7. **K-Means**

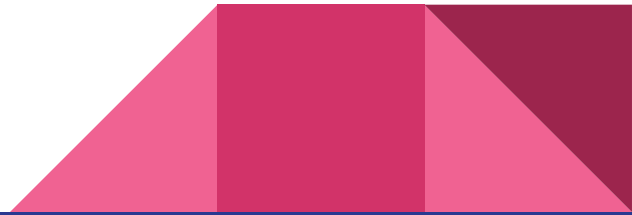
4. Train the model

- The goal of training is to answer a question or make a prediction correctly as often as possible
- Linear regression example: algorithm would need to learn values for *m* (or *W*) and *b* (*x* is input, *y* is output)
- Each iteration of process is a training step



5. Evaluate the model

- Uses some metric or combination of metrics to "measure" objective performance of model
- Test the model against previously unseen data
- This unseen data is meant to be somewhat representative of model performance in the real world, but still helps tune the model (as opposed to test data, which does not)
- Good train/eval split? 80/20, 70/30, or similar, depending on domain, data availability, dataset particulars, etc.

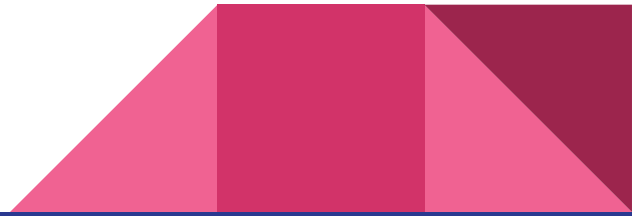


6. Parameter Tuning

- This step refers to *hyperparameter* tuning, which is an "artform" as opposed to a science

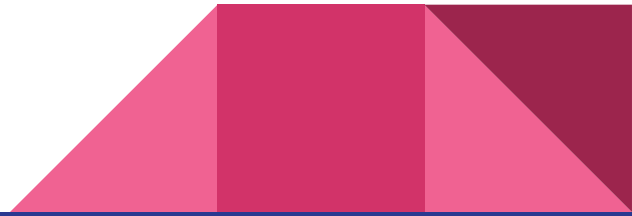
  Tune model parameters for improved performance

  Simple model hyperparameters may include: number of training steps, learning rate, initialization values and distribution, etc.



7. Make predictions

Using further (test set) data which have, until this point, been withheld from the model (and for which class labels are known), are used to test the model, a better approximation of how the model will perform in the real world



Implementation (using Colab)

Environment : Colab, Tensorflow 2

Programming Language: Python

Library :

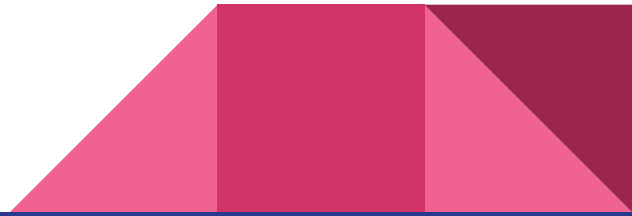
import sys

import numpy as np

import pandas as pd

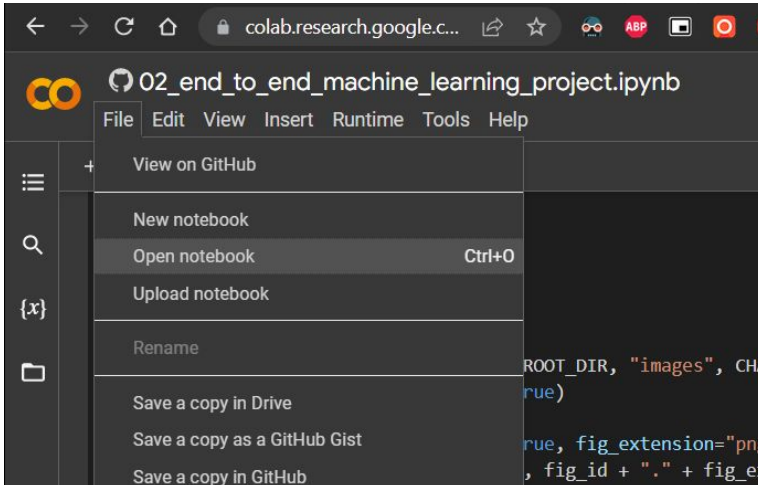
import os import matplotlib as mpl

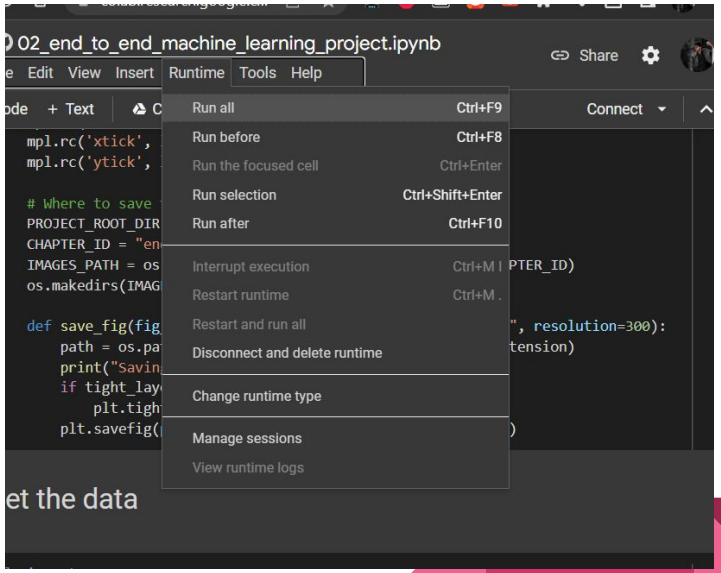
import matplotlib.pyplot as plt



Implementation

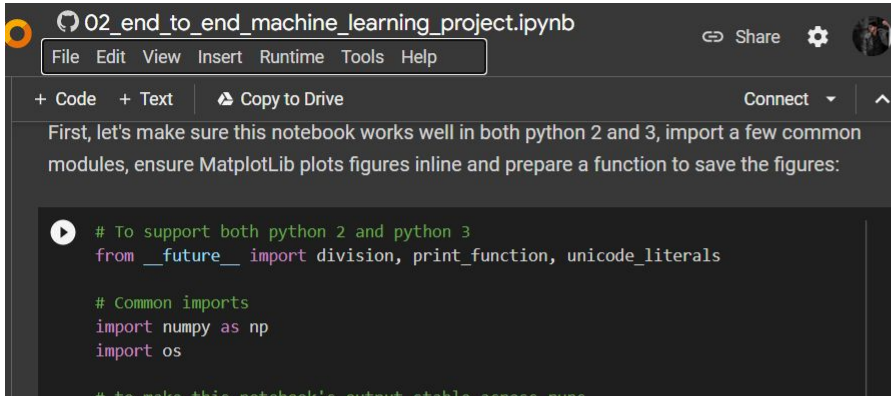
Import and run the code





Implementation

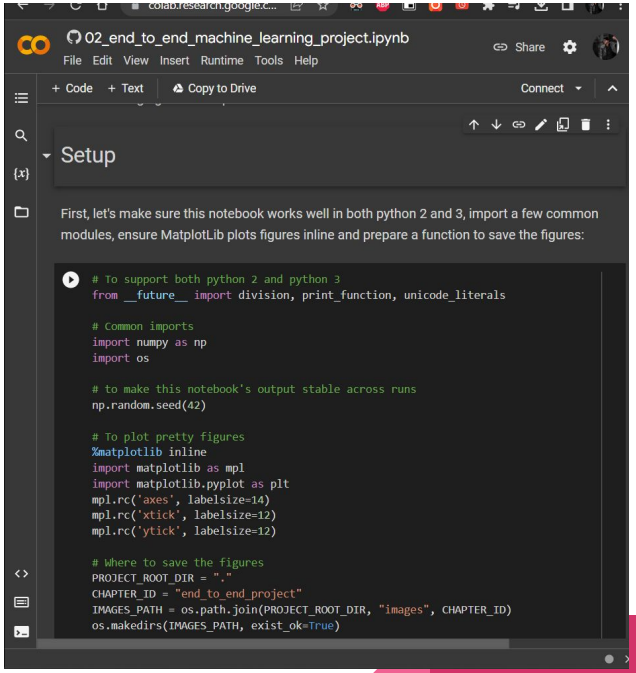
Setting up the Colab notebook for the desired version (python 2 or 3) by importing some modules



Implementation

Getting the data We have housing data residing in California that contains properties of a house like

longitude,latitude,housing_median_age,total_rooms,total_bedrooms, population,households, median_income, median_house_value and ocean_proximit



Implementation

After Fetching the data, we can have the loaded data on our Colab, as shown below.

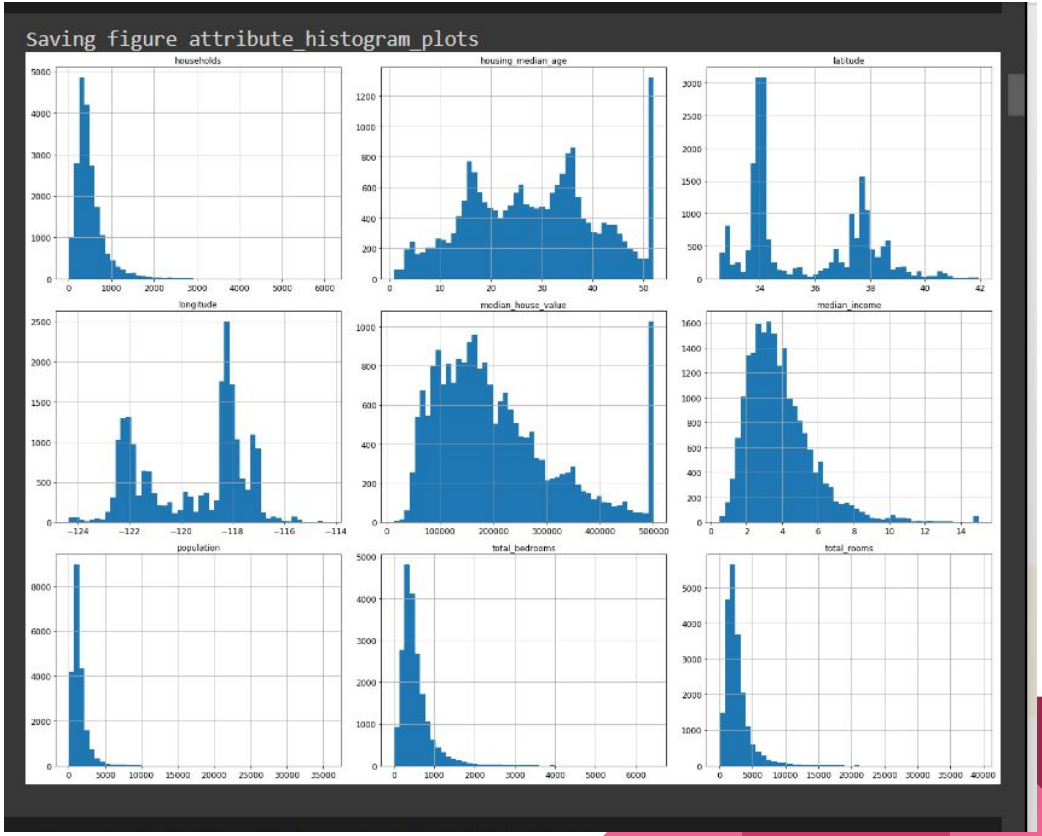
```
housing.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value	ocean_proximity
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600.0	NEAR BAY
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500.0	NEAR BAY
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100.0	NEAR BAY
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300.0	NEAR BAY
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200.0	NEAR BAY

housing.info()

Implementation

We have the generated figure for the histogram information of the house.

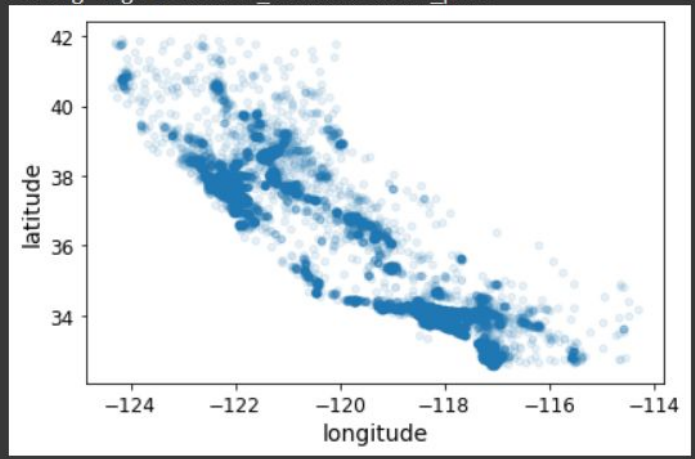


Implementation

Discover and visualize the data to gain insights

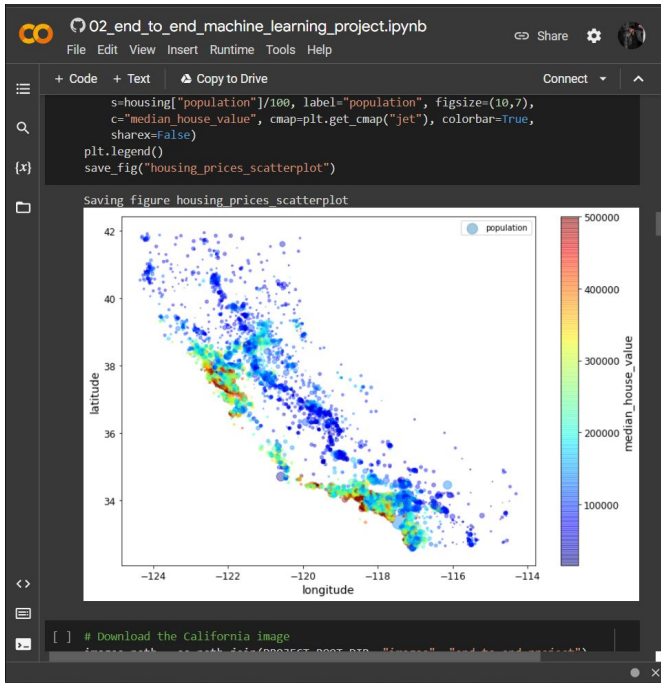
```
housing.plot(kind="scatter", x="longitude", y="latitude", alpha=0
save_fig("better_visualization_plot")
```

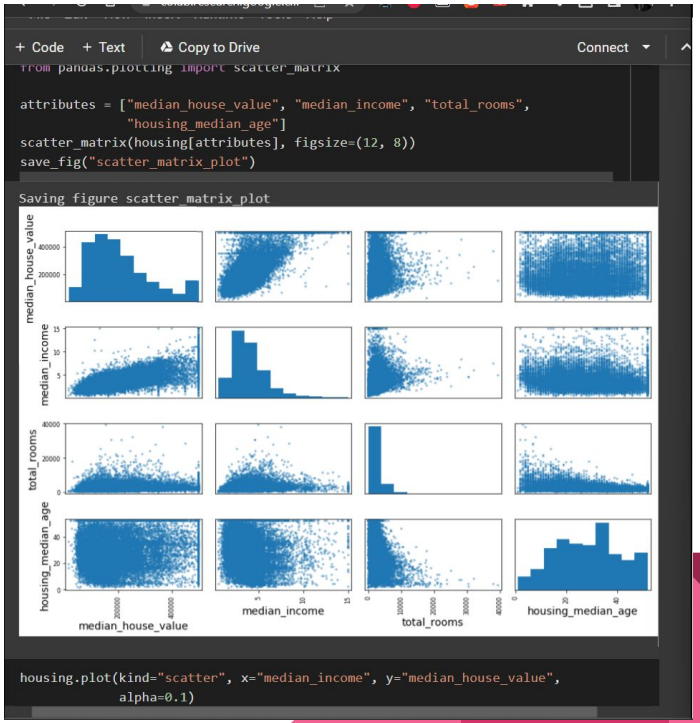
```
Saving figure better_visualization_plot
```



e argument `sharex=False` fixes a display bug (the x-axis values and legen

Implementation





Implementation

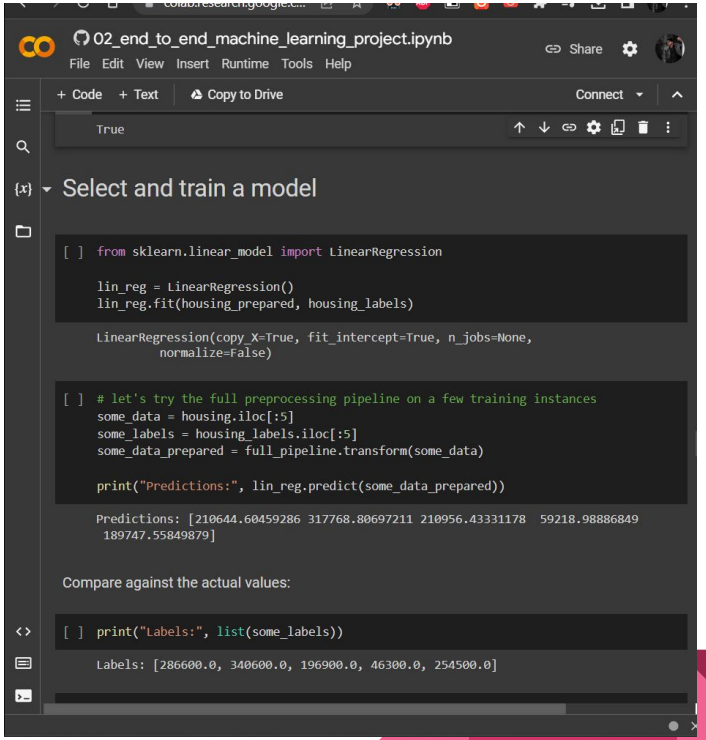
Prepare the data for Machine Learning algorithms

- While cleaning the data, we have the preprocessing phase, where we clean the data to create the machine learning model.

- The preprocessing steps include cleaning the data that is filling missing information or dropping unnecessary columns or rows that are not important later for the model to be created

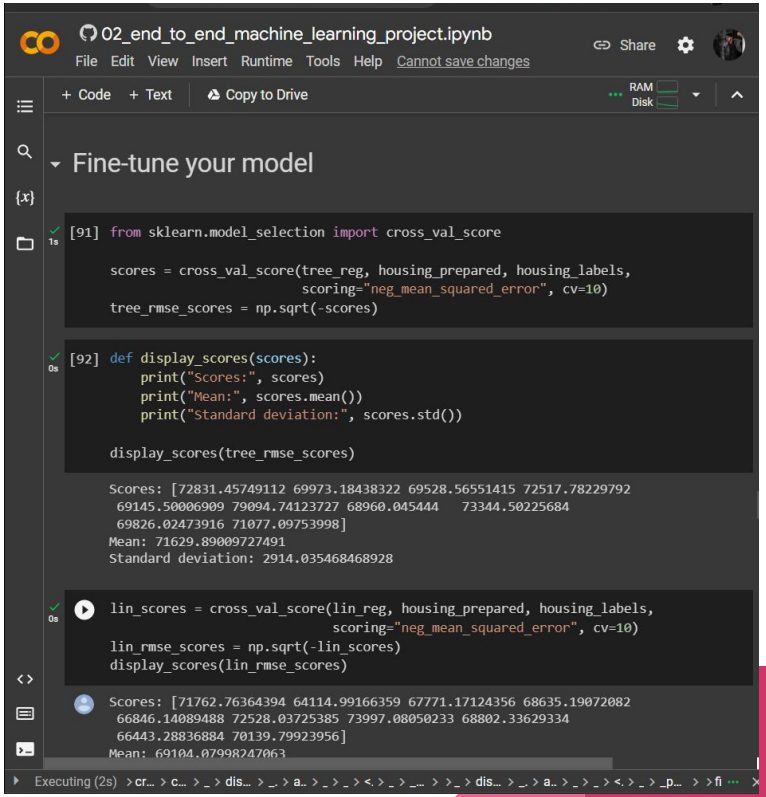
Implementation

Select and train a model



Implementation

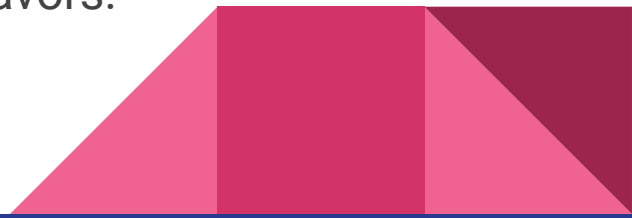
Fine tune your model



Conclusion

Finally, successful End-to-End Machine Learning projects necessitate a blend of technical proficiency, subject-matter knowledge, and project management abilities. Collaboration between data scientists, software engineers, subject matter experts, and stakeholders is required for them.

To make sure the project achieves its goals and adds value for the company or organization. The ability to harness the power of machine learning to gain insights, make predictions, and spur innovation is what makes End-to-End Machine Learning projects potentially difficult but rewarding endeavors.



References

Banoula, M. (2023, February 16).

Machine learning steps: A complete guide: Simplilearn. Simplilearn.com. Retrieved February 21, 2023, from https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps

ML-ops.org. End-to-end Machine Learning Workflow. (2022, July 22).

Retrieved February 21, 2023, from https://ml-ops.org/content/end-to-end-ml-workflow