End to end Machine Learning

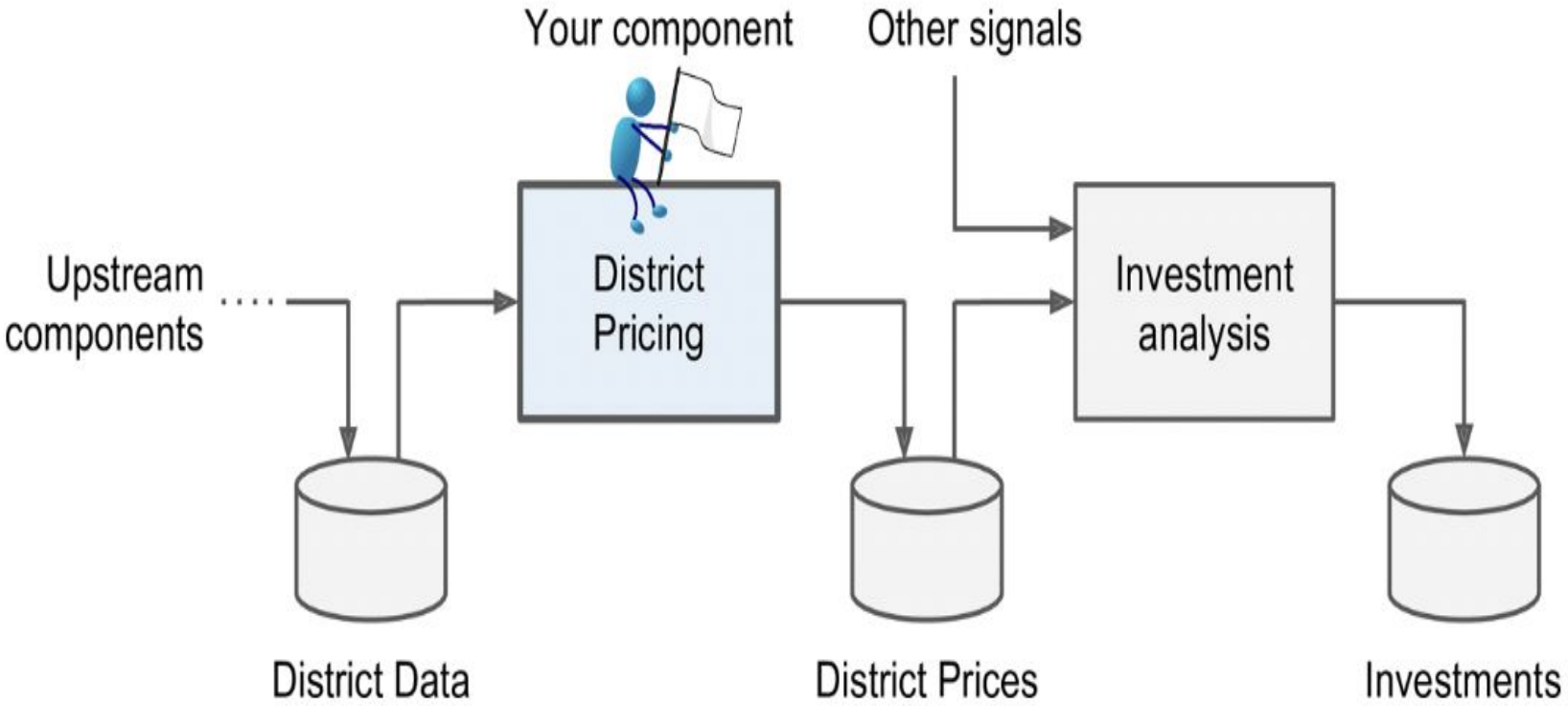
NORINA AKHTAR 19643

Table of Contents

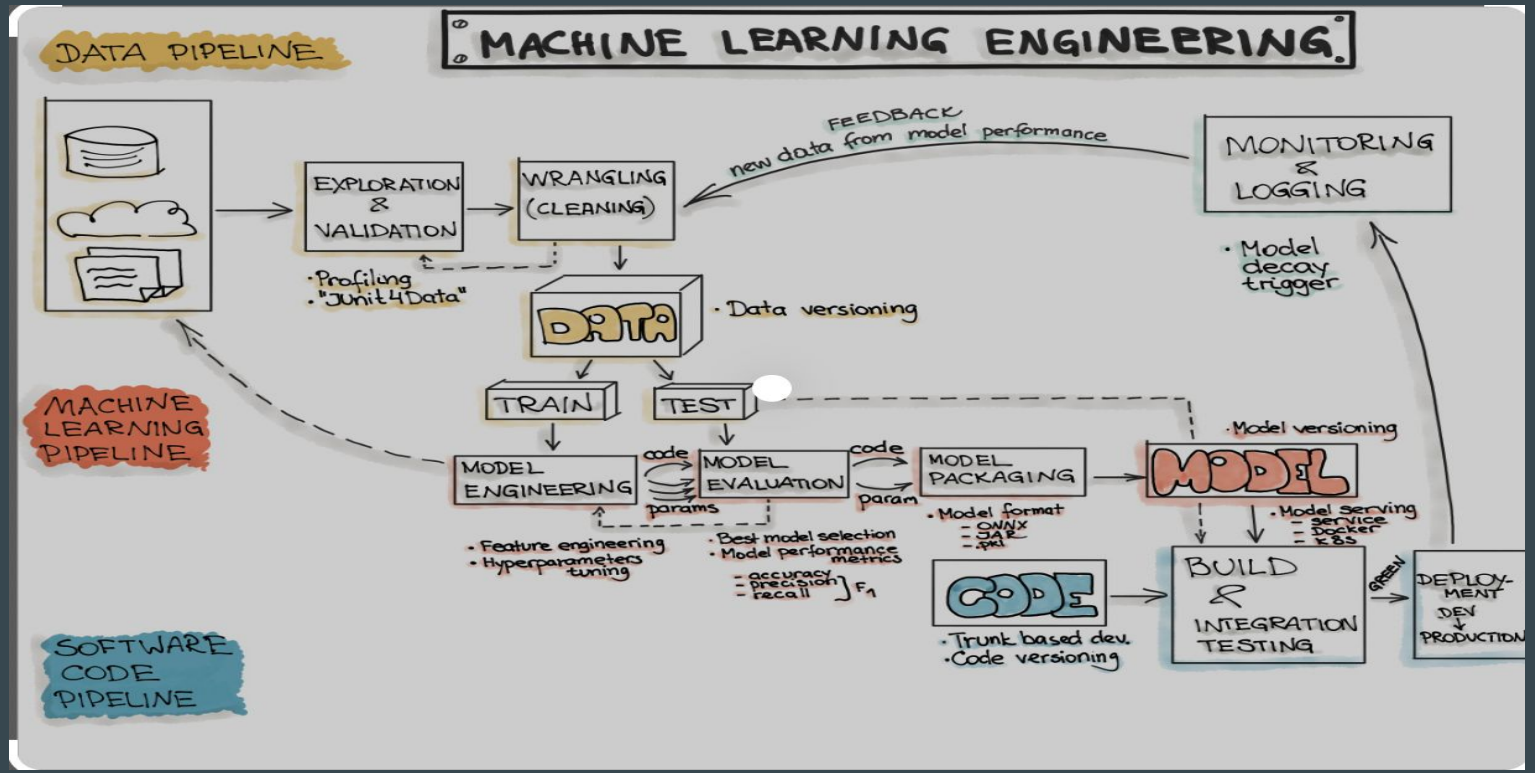
Introduction

End-to-end (E2E) learning refers to training a possibly complex learning system represented by a single model (specifically a Deep Neural Network) that represents the complete target system, bypassing the intermediate layers usually present in traditional pipeline designs.

End to End learning in the context of AI and ML is a technique where the model learns all the steps between the initial input phase and the final output result. This is a deep learning process where all of the different parts are simultaneously trained instead of sequentially.


Your component
Other signals
Upstream components
District Pricing
Investment analysis
District Data
District Prices
Investments

Design


MACHINE LEARNING ENGINEERING
DATA PIPELINE
EXPLORATION & VALIDATION
WRANGLING (CLEANING)
FEEDBACK
new data from model performance
MONITORING & LOGGING
· Model decay trigger
· Profiling
· "Junit4Data"
DATA
· Data versioning
MACHINE LEARNING PIPELINE
TRAIN
TEST
· Model versioning
MODEL ENGINEERING
code
params
MODEL EVALUATION
code
param
MODEL PACKAGING
MODEL
· Model format
- ONNX
- JAR
- pkl
· Model serving
- service
- Docker
- K8S
· Feature engineering
· Hyperparameters tuning
· Best model selection
· Model performance metrics
- accuracy
- precision
- recall
F1
CODE
· Trunk based dev.
· Code versioning
BUILD & INTEGRATION TESTING
green
DEPLOY-MENT
DEV
PRODUCTION
SOFTWARE CODE PIPELINE

Implementation

Follow the Steps:

- Look at the big picture
- Get the data
- Discover and visualize the data to gain insides
- Prepare the data for machine learning Algorithms
- Select and train a model
- Fine-Tune your model
- Launch monitor and maintain

```
[ ]  1  import os
     2  import tarfile
     3  import urllib.request
     4
     5  DOWNLOAD_ROOT = "https://raw.githubusercontent.com/ageron/handson-ml/master/"
     6  HOUSING_PATH = os.path.join("datasets", "housing")
     7  HOUSING_URL = DOWNLOAD_ROOT + "datasets/housing/housing.tgz"
     8
     9  def fetch_housing_data(housing_url=HOUSING_URL, housing_path=HOUSING_PATH):
    10      os.makedirs(housing_path, exist_ok=True)
    11      tgz_path = os.path.join(housing_path, "housing.tgz")
    12      urllib.request.urlretrieve(housing_url, tgz_path)
    13      housing_tgz = tarfile.open(tgz_path)
    14      housing_tgz.extractall(path=housing_path)
    15      housing_tgz.close()
```

```
[ ]  1  fetch_housing_data()
```

```
[ ]  1  import pandas as pd
     2
     3  def load_housing_data(housing_path=HOUSING_PATH):
     4      csv_path = os.path.join(housing_path, "housing.csv")
     5      return pd.read_csv(csv_path)
```

```
[ ]  1  housing = load_housing_data()
     2  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

```
[ ]  1  housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20640 entries, 0 to 20639
Data columns (total 10 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   longitude           20640 non-null  float64
 1   latitude            20640 non-null  float64
 2   housing_median_age  20640 non-null  float64
 3   total_rooms         20640 non-null  float64
 4   total_bedrooms      20433 non-null  float64
 5   population          20640 non-null  float64
 6   households          20640 non-null  float64
 7   median_income       20640 non-null  float64
 8   median_house_value  20640 non-null  float64
 9   ocean_proximity     20640 non-null  object
dtypes: float64(9), object(1)
memory usage: 1.6+ MB
```

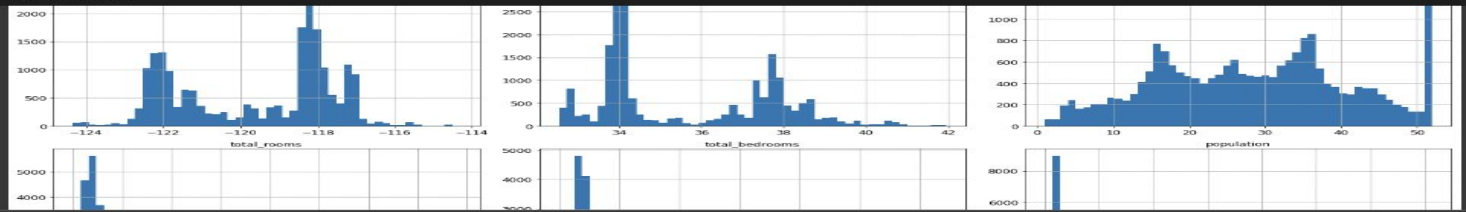
```
[ ]  1  housing["ocean_proximity"].value_counts()
```

```
<1H OCEAN      9136
INLAND         6551
NEAR OCEAN     2658
NEAR BAY       2290
ISLAND            5
Name: ocean_proximity, dtype: int64
```

```
[ ]  1  housing.describe()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
count	20640.000000	20640.000000	20640.000000	20640.000000	20433.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	-119.569704	35.631861	28.639486	2635.763081	537.870553	1425.476744	499.539680	3.870671	206855.816909
std	2.003532	2.135952	12.585558	2181.615252	421.385070	1132.462122	382.329753	1.899822	115395.615874
min	-124.350000	32.540000	1.000000	2.000000	1.000000	3.000000	1.000000	0.499900	14999.000000
25%	-121.800000	33.930000	18.000000	1447.750000	296.000000	787.000000	280.000000	2.563400	119600.000000
50%	-118.490000	34.260000	29.000000	2127.000000	435.000000	1166.000000	409.000000	3.534800	179700.000000
75%	-118.010000	37.710000	37.000000	3148.000000	647.000000	1725.000000	605.000000	4.743250	264725.000000
max	-114.310000	41.950000	52.000000	39320.000000	6445.000000	35682.000000	6082.000000	15.000100	500001.000000

```
[ ]  1  %matplotlib inline
     2  import matplotlib.pyplot as plt
     3  housing.hist(bins=50, figsize=(20,15))
     4  save_fig("attribute_histogram_plots")
     5  plt.show()
```



Test

screenshot is attached on next page,, you can refer the ipynb file given on github for more details

```
                                                  Pipeline(steps=[('imputer',
                                                                   SimpleImputer(strategy='median')),
                                                                  ('attribs_adder',
                                                                   FunctionTransformer(func=<function add_extra_features at 0x7f86bc86aca0>)),
                                                                  ('std_scaler',
                                                                   StandardScaler())]),
                                                  ['longitude',
                                                   'latitude',
                                                   'housing_median_age',
                                                   'total_rooms',
                                                   'total_be...
        5.47789150e-02, 1.07031322e-01, 4.82031213e-02, 6.79266007e-03,
        1.65706303e-01, 7.83480660e-05, 1.52473276e-03, 3.02816106e-03]),
                                                             k=5)),
                                        ('svm_reg',
                                         SVR(C=157055.10989448498,
                                             gamma=0.26497040005002437))]),
              n_jobs=4,
              param_grid=[{'feature_selection__k': [1, 2, 3, 4, 5, 6, 7, 8, 9,
                                                    10, 11, 12, 13, 14, 15, 16],
                           'preparation__num__imputer__strategy': ['mean',
                                                                   'median',
                                                                   'most_frequent']}],
              scoring='neg_mean_squared_error', verbose=2)
```

+ Code + Markdown

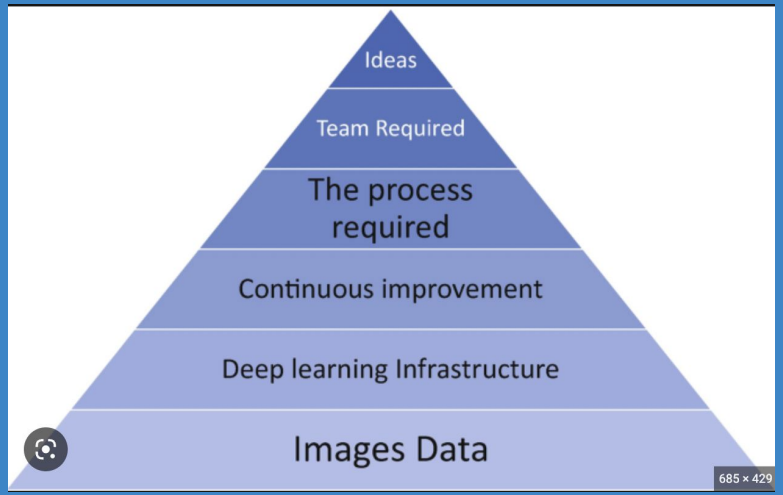
[141]
```
grid_search_prep.best_params_
```

```
{'feature_selection__k': 1, 'preparation__num__imputer__strategy': 'mean'}
```

The best imputer strategy is mean

Enhancement Ideas

- End to end can be enhanced using huge amount of data
- Improve the quality of data
- Optimize the hyperparameters
- Use better feature representation
- Experiment with different algorithms
- Use transfer learning
- Regularization techniques
- Use ensemble methods



Conclusion

End to End is indisputably a great tool for solving elaborate tasks. The idea of using a single model that can specialize to predict the outputs directly from the inputs allows the development of otherwise extremely complex systems that can be considered state-of-the-art. However, every enhancement comes with a price: while consecrated in the academic field, the industry is still reluctant to use E2E to solve its problems due to the need for a large amount of training data and the difficulty of validation.

References

https://towardsdatascience.com/e2e-the-every-purpose-ml-method-5d4f20dafee4

https://arxiv.org/abs/1704.08305

https://arxiv.org/abs/1706.05125

https://hc.labnet.sfbu.edu/~henry/sfbu/course/hands_on_ml_with_schikit_2nd/end_to_end/slide/topic_frame.html#process

https://www.google.com/search?q=enhancement+idea+for+end+to+end+learning&rlz=1C5CHFA_enUS1011US1011&source=lnms&tbm=isch&sa=X&ved=2ahUKEwieu87ChJv9AhWKITQIHQ0vC5IQ_AUoAXoECAEQAw&biw=1440&bih=789&dpr=2#imgrc=YZOhniVpnI946M