Chapter 2

12.4.2021

Thordis Thorsteins

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

1. Frame the problem

Purpose:

- Make sure we’re solving the right problem
- Make sure we’re using a sensible approach

1. Frame the problem

- What's the objective?

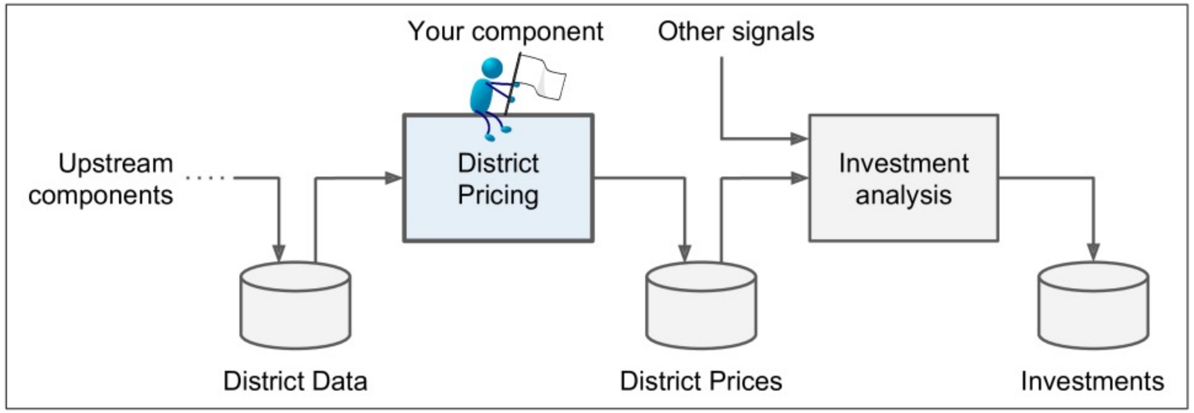

Figure 2-2. A Machine Learning pipeline for real estate investments

- What sort of problem is this?
  - Supervised/ unsupervised/ reinforcement learning?
  - Classification/ regression/ something else?
  - Batch learning/ online learning?

1. Frame the problem

- What's the objective?

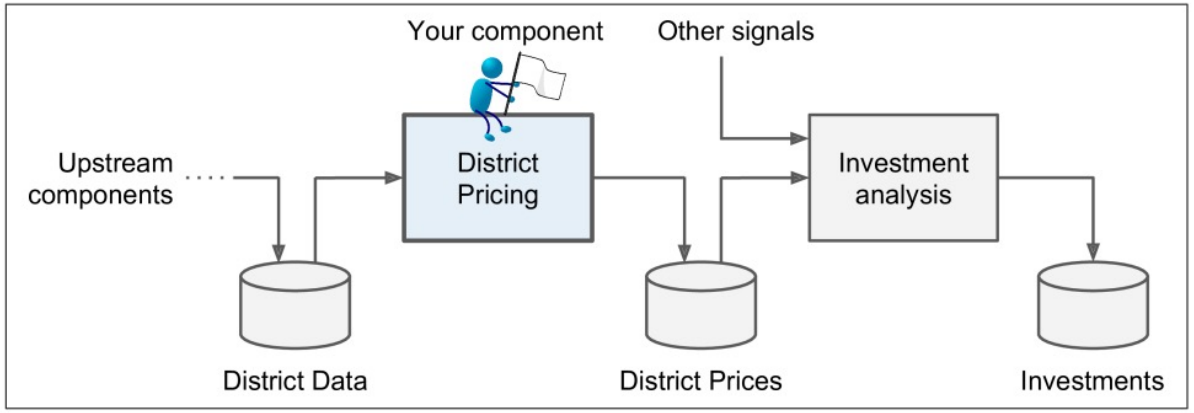

Figure 2-2. A Machine Learning pipeline for real estate investments

- What sort of problem is this?
  - Supervised/ unsupervised/ reinforcement learning?
  - Classification/ regression/ something else?
  - Batch learning/ online learning?

1. Frame the problem

- What's the objective?
- What sort of problem is this?
  - Supervised. Regression. Batch learning
- Select a performance measure

Equation 2-1. Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

1. Frame the problem

- What's the objective?
- What sort of problem is this?
    - Supervised. Regression. Batch learning
- Select a performance measure

Equation 2-1. Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathbf{X}, h) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h\left(\mathbf{x}^{(i)}\right) - y^{(i)}\right)^2}$$

- Validate assumptions

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
    a) Getting started with Jupyter notebooks
    b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

The rest of this section is in a notebook

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

3. Create a test set

- Around 20% of the dataset
- Does random sampling make sense?
  - Stratified sampling is more suitable for small datasets
    - Our dataset has around 20k instances
- Need to be able to find the same test set again in a subsequent run. Solutions include:
  - Save the test data or
  - Set the random number generator’s seed or
  - Use instance identifiers to choose test set

The rest of this section is in a notebook

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

4. Explore the data

Purpose:

- Get familiar with the data we’re trying to model
- Spot potential problems/data quirks that could impact the model training
- High level discovery into patterns of values

The rest of this section is in a notebook

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

5. Data prep

Purpose:

- Make sure that the data is in a suitable format to produce an effective model from

5. Data prep

- Attribute combinations
- Separate labels from predictors
- Data cleaning
  - Missing data
- Handling attributes that are not numeric
  - "Most ML algorithms prefer to work with numbers"
  - Options for categorical non-numeric values:
    - Ordinal encoding / One-hot encoding / Representation learning (more on this in later chapters)

5. Data prep

- Attribute combinations
- Separate labels from predictors
- Data cleaning
  - Missing data
- Handling attributes that are not numeric
  - “Most ML algorithms prefer to work with numbers”
  - Options for categorical non-numeric values:
    - Ordinal encoding / One-hot encoding / Representation learning (more on this in later chapters)

5. Data prep

- Attribute combinations
- Separate labels from predictors
- Data cleaning
  - Missing data
- Handling attributes that are not numeric
  - "Most ML algorithms prefer to work with numbers"
  - One-hot encoding
- Feature scaling
  - "With a few exceptions, ML algorithms don't perform well when the input numerical attributes have
  - Min-max scaling/ standardisation

5. Data prep

- Attribute combinations
- Separate labels from predictors
- Data cleaning
  - Missing data
- Handling attributes that are not numeric
  - "Most ML algorithms prefer to work with numbers"
  - One-hot encoding
- Feature scaling
  - "With a few exceptions, ML algorithms don't perform well when the input numerical attributes have
  - Min-max scaling/ standardisation

5. Data prep

- Attribute combinations
- Separate labels from predictors
- Data cleaning
  - Missing data
- Handling attributes that are not numeric
  - “Most ML algorithms prefer to work with numbers”
  - One-hot encoding
- Feature scaling
  - “With a few exceptions, ML algorithms don’t perform well when the input numerical attributes have
  - Standardisation
- Functions, instead of manual prep, are strongly encouraged

The rest of this section is in a notebook

5. Data prep

- Takeaway: Scikit-Learn is great

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

6. Pick and train a model

- Try models from various categories of machine learning algorithms
  - Shortlist ~ 2-5 promising models
- Start simple

The rest of this section is in a notebook

6. Pick and train a model

- Takeaway: The process for training a model is very similar regardless of the type of model we chose. Syntax is standardised

```
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()
lin_reg.fit(housing_prepared, housing_labels)
```

```
from sklearn.tree import DecisionTreeRegressor

tree_reg = DecisionTreeRegressor(random_state=42)
tree_reg.fit(housing_prepared, housing_labels)
```

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

7. Fine tune the model

- Try out different hyperparameters
  - GridSearchCV for small numbers of hyperparameter combinations
  - RandomizedSearchCV otherwise
- Once you’re happy with the parameters, evaluate the model on the test set
- Document the model and creation process
  - Assumptions
  - What worked and what didn’t
  - Limitations
  - Key predictors of the model

The rest of this section is in a notebook

End-to-end Machine Learning Project

1. Frame the problem
2. Get the data
   a) Getting started with Jupyter notebooks
   b) Download the data
3. Create a test set
4. Explore the data
5. Data prep
6. Select and train a model
7. Fine-tune the model
8. Launch, monitor and maintain

8. Launch, monitor and maintain

- Clean code, write documentation and tests
- Deploy model
  - Web service queried through a REST API
  - Cloud solution (e.g. Google Cloud AI Platform)
- Monitoring code
  - Alerts if anything fails or assumptions are broken
  - Monitor model performance and its trend
    - Watch out for data drift
  - Monitor quality and statistics of input data
- Keep backups of every model and version of dataset