Numeric feature

- preprocessing
  - scale
    - tree 기반 모델의 경우는 feature간의 scale차이가 상관없지만
    - 거리기반, linear, Neural net, gradient 기반 모델들은 scale의 영향을 받는다.
      - minmax, standardization 사용한다.
      - `sklearn.preprocessing.MinMaxScaler`
  - outliers
    - linear model이 취약하다.
  - rank
    - 각 value들을 동일한 차이로 만든다.
    - scale뿐만 아니라 outlier 처리에도 좋다.
  - log / sqrt transform
    - 큰 숫자는 작게 작은 숫자는 크게 만들어주는 역할
    - outlier 처리에 좋다.
- feature generation
  - prior knowledge, EDA를 통해 아이디어를 얻는다.

Categorical, ordinal feature

- target과의 관계를 고려하여 encoding방법을 선택하면 좋다.
- Label encoding
  - tree 기반 모델에서만 사용가능하다.
  - ordinal한 상황에서 사용하기도 좋다.
  - `sklearn.preprocessing.LabelEncoder` : 알파벳순서로 1,2,3... encode
- Frequency encoding
  - 해당 라벨의 빈도수로 encoding (0~1 사이로 만든다)
  - linear, tree 모델 모두에게 도움을 줄 수 있겠다.
  - same frequency가 나올 수 있으니까 주의한다.

```
encoding = titanic.groupby('Embarked').size()
freq_encoding = encoding/len(titanic)
titanic['freq_enc'] = titanic['Embarked'].map(freq_encoding)
```

- one hot encoding
  - non tree based 모델에서 사용한다.
  - dataset이 커지는 단점이 있다.
- feature generation
  - feature간의 interaction 고려한다. (Categorical feature끼리)
    - non tree based 모델에 유용할 것이다.

Datetime and coordinates (feature generation)

- date and time
  - periodicity
  - Time since
  - Difference between dates
- coordinates
  - center of cluster

    - aggregated statistics
    - distance

Missing values

- Hidden NaNs
    - histogram그려봤더니 특이한 값이 있다면 NA값인지 의심해보자.
    - Fillna (상황에 따라 다양한 방법들중 선택)
        - -999, -1 etc.
            - linear model에서는 쓰지말자.
        - mean, median
        - reconstruct value
    - isnull feature generation
        - dataset이 커지는 단점이 있다.
        - NA인 경우를 1로해서 새로운 feature를 생성한다.
- outlier를 missing value로 다루기 가능하다. (outlier 지우기가 아깝다면)
- test에는 있고 train에는 없는 Categorical 라벨에 대해 주의하자.
- XGboost 는 NA값 handle을 해준다.
- 일반적으로 feature generation하기전에 fillna하지 않는다.

참고 링크 : https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

Feature extraction from text and image

- Bag of words
    - preprocessing
        - lowercase (대,소문자 구분없이)
        - lemmatization and stemming
        - stopwords
    - N-gram
    - TFIDF (postprocessing)
        - 각 document에서 많이 나온 빈도를 가중치(TF), 전체에서 적게 나온 단어를 더 가중치(IDF)
        - `sklearn.feature_extraction.text.TfidfVectorizer`
- Word2vec, CNN
    - Word2vec
        - word를 vector로 나타낸 것이다.
        - Words : glove, fasttext, etc
        - sentence : Doc2vec, etc
        - pretrained model이 있으니까 이용할 수 있다.
    - CNN
        - image -> vector
        - Fine tuining
            - pretrained model의 일부만 바꿔서 사용한다.
        - Augumentation
            - image data를 약간씩 바꿔서 data를 수를 늘린다.