EDA

1. 데이터 타입별

```
app_train.select_dtypes('object')
```

	NAME_CONTRACT_TYPE	CODE_GENDER	FLAG_OWN_CAR	FLAG_OWN_REALTY	NAME_TYPE_SUITE	NAME_INCOME_TYPE	NAME_E
0	Cash loans	M	N	Y	Unaccompanied	Working	Secondary special
1	Cash loans	F	N	N	Family	State servant	Higher ed
2	Revolving loans	M	Y	Y	Unaccompanied	Working	Secondary special
3	Cash loans	F	N	Y	Unaccompanied	Working	Secondary special
4	Cash loans	M	N	Y	Unaccompanied	Working	Secondary special
5	Cash loans	M	N	Y	Spouse, partner	State servant	Secondary special
6	Cash loans	F	Y	Y	Unaccompanied	Commercial associate	Higher ed
7	Cash loans	M	Y	Y	Unaccompanied	State servant	Higher ed
8	Cash loans	F	N	Y	Children	Pensioner	Secondary special
9	Revolving loans	M	N	Y	Unaccompanied	Working	Secondary

```
app_train.select_dtypes('object').apply(pd.Series.nunique, axis=0)
```

```
NAME_CONTRACT_TYPE            2
CODE_GENDER                   3
FLAG_OWN_CAR                  2
FLAG_OWN_REALTY               2
NAME_TYPE_SUITE               7
NAME_INCOME_TYPE              8
NAME_EDUCATION_TYPE           5
NAME_FAMILY_STATUS            6
NAME_HOUSING_TYPE             6
OCCUPATION_TYPE              18
WEEKDAY_APPR_PROCESS_START    7
ORGANIZATION_TYPE            58
FONDKAPREMONT_MODE            4
HOUSETYPE_MODE                3
WALLSMATERIAL_MODE            7
EMERGENCYSTATE_MODE           2
dtype: int64
```

2. Encoding Categorical Variables

- 논란이 많음 (모델에 따라 달라짐) ~~대체적으로 필자의 논점에 동의... 허진경교수님의 생각은 무조건 Label encoding이 낫다하여 과제 할 때 그렇게 했었고 해당 데이터셋에서는 큰 성능의 차이는 없었음~~

- 안전한 방법은 one-hot encoding 후 PCA

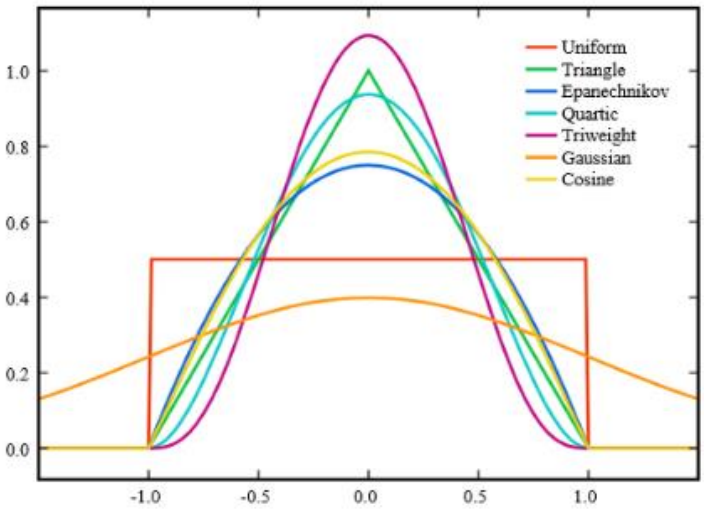
- 여기서는 2개의 categorical 변수에 대해서는 label encoding 그 외는 one-hot encoding 방식을 사용하였음

3. Kernel Density Estimation(커널 밀도 추정)

- Parametric 밀도추정: 미리 pdf(probability density function)에 대한 모델을 정해놓고 데이터들로부터의 모델의 파라미터를 추정하는 방식, 평균과 분산만 구하면 계산 바로 가능
- Non-parametric: 대부분의 경우 사전 정보나 지식 없이 관측한 데이터들로부터 확률밀도함수를 추정해야함
  - ■ Histogram: non-parametric 밀도 추정의 가장 단순한 형태
    - ➔ 문제점
      1. Bin경계의 불연속 지점
      2. bin 크기 및 시작 위치에 따라 히스토그램이 달라짐
      3. 고차원 데이터의 경우 사용이 힘듦
  - ■ KDE: histogram의 문제점을 보완하기 위해 kernel함수를 활용하여 밀도 추정하는 방법
    - ➔ 커널함수란? : 원점을 중심으로 대칭이면서 적분값이 1인 non-negative함수로 가우시안, Epanechnikov, uniform 함수등이 있다.

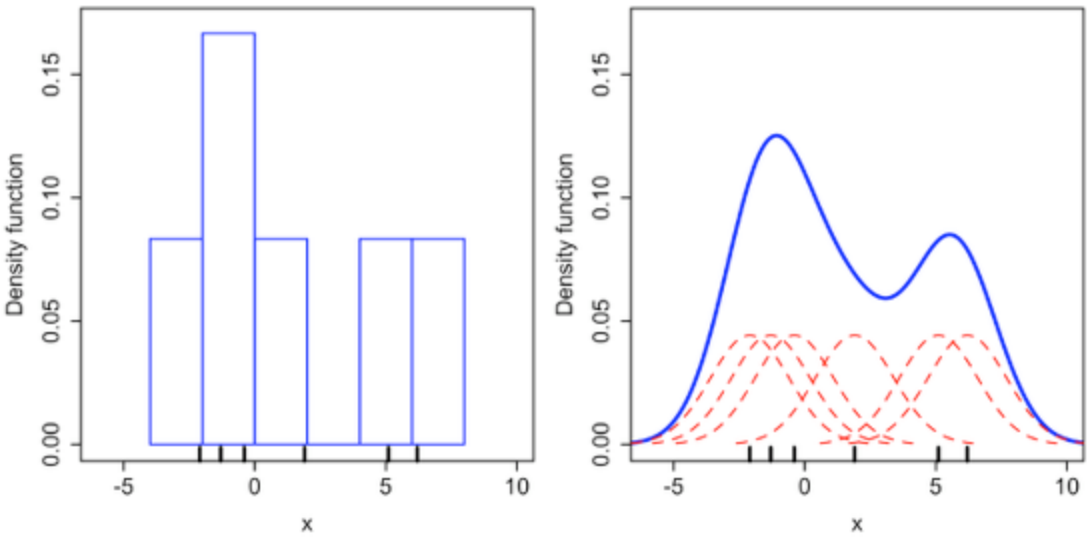
$$\int_{-\infty}^{\infty} K(u)du = 1 \quad \text{--- (2)}$$

$$K(u)=K(-u),\ K(u)\geq 0,\ \forall u \quad \text{--- (3)}$$



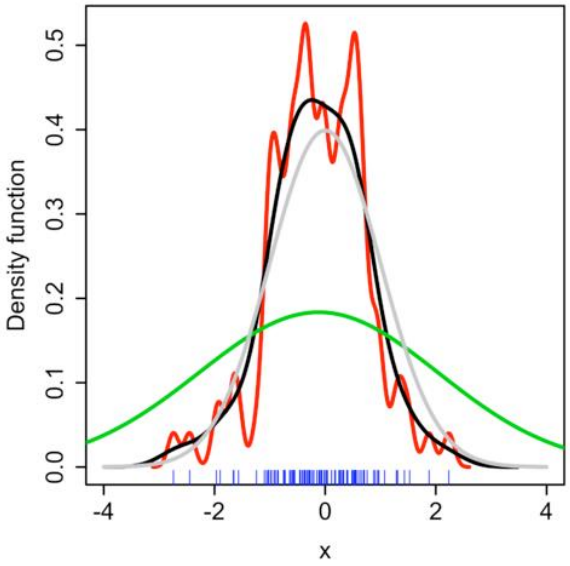
<그림 3> 다양한 커널 함수들 (출처: 위키피디아)

1. 관측된 데이터 각각마다 해당 데이터 값을 중심으로 하는 커널 함수를 생성한다: K(x-xi)
2. 이렇게 만들어진 커널 함수들을 모두 더한 후 전체 데이터 개수로 나눈다.



<그림 4> "Comparison of 1D histogram and KDE" by Drleft

➔ 즉, KDE를 통해 얻은 확률밀도함수는 히스토그램 확률밀도함수를 스무딩(smoothing)한 것으로 볼 수 있으며, 스무딩 정도는 어떤 bandwidth값의 커널 함수를 사용했는지에 따라 달라짐



출처: https://darkpgmr.tistory.com/147

- 그래프 그릴 때, kde 쓰는 이유

Bar plots can tell us what the mean of our dataset is, but they don't give us any hints as to the distribution of the dataset values. For all we know, the data could be clustered around the mean or spread out evenly across the entire range.

To find out more about each of these datasets, we'll need to examine their distributions. A common way of doing so is by plotting the data as a histogram, but histograms have their drawback as well.

Seaborn offers another option for graphing distributions: *KDE Plots*.

KDE stands for Kernel Density Estimator. A KDE plot gives us the sense of a *univariate* as a curve. A univariate dataset only has one variable and is also referred to as being *one-dimensional*, as opposed to *bivariate* or two-dimensional datasets which have two variables.

KDE plots are preferable to histograms because depending on how you group the data into bins and the width of the bins, you can draw wildly different conclusions about the shape of the data. Using a KDE plot can mitigate these issues, because they smooth the datasets, allow us to generalize over the shape of our data, and aren't beholden to specific data points