Predict Future Sales

- 대회 링크 : https://www.kaggle.com/c/competitive-data-science-predict-future-sales/overview
- 대회 개요 : 러시아의 최대 소프트웨어 회사인 1C Company에서 제공하는 일상적인 영업 데이터.
- 대회 문제 : 다음달에 모든 제품과 가게의 총 매출을 예상해 줄 것에 대한 요청
- 평가 방법 : RMSE(root mean squaed error)
- 제출 형식 : 데이터 셋의 각 ID에 대해 총 판매수를 예측하기

```
ID, item_cnt_month
0, 0.5
1, 0.5
3, 0.5
```

In [1]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/competitive-data-science-predict-future-sales/items.csv
/kaggle/input/competitive-data-science-predict-future-sales/sample_submission.csv
/kaggle/input/competitive-data-science-predict-future-sales/item_categories.csv
/kaggle/input/competitive-data-science-predict-future-sales/sales_train.csv
/kaggle/input/competitive-data-science-predict-future-sales/shops.csv
/kaggle/input/competitive-data-science-predict-future-sales/test.csv
```

데이터 불러오기

In [3]:
```
train = pd.read_csv("/kaggle/input/competitive-data-science-predict-future-sales
test = pd.read_csv("/kaggle/input/competitive-data-science-predict-future-sales/
sub = pd.read_csv("/kaggle/input/competitive-data-science-predict-future-sales/s
items = pd.read_csv("/kaggle/input/competitive-data-science-predict-future-sales
items_cat = pd.read_csv("/kaggle/input/competitive-data-science-predict-future-s
shops = pd.read_csv("/kaggle/input/competitive-data-science-predict-future-sales
```

- sales_train.csv : 학습 데이터. 2013년 1월부터 2015년 10월까지의 일일 기록 데이터.
- test.csv - 테스트 데이터. 상점과 제품의 2015년 11월 매출을 예측.
- sample_submission.csv : 제출용 샘플 파일
- items.csv : 항목/제품에 대한 추가 정보
- item_categories.csv : 항목 카테고리에 대한 추가 정보
- shops.csv : 상점에 대한 추가 정보

파일명	내용	행열
sales_train.csv	학습 데이터. 2013년 1월부터 2015년 10월까지의 일일 기록 데이터	2935849행, 6열
test.csv	테스트 데이터. 상점과 제품의 2015년 11월 매출을 예측	214200행, 3열
items.csv	항목/제품에 대한 추가 정보	22170행, 3열
item_categories.csv	항목 카테고리에 대한 추가 정보	84행, 2열
shops.csv	상점에 대한 추가 정보	60행, 2열
sample_submission.csv	올바른 형식의 샘플 파일 제출	

기본 데이터 탐색

In [4]:
```
print("학습용 데이터 행열 : {}".format(train.shape))
print("제출용 데이터 행열 : {}".format(sub.shape))
print("테스트 데이터 행열 : {}".format(test.shape))
print("items 데이터 행열 : {}".format(items.shape))
print("items_categories 데이터 행열 : {}".format(items_cat.shape))
print("shops 데이터 행열 : {}".format(shops.shape))
```

```
학습용 데이터 행열 : (2935849, 6)
제출용 데이터 행열 : (214200, 2)
테스트 데이터 행열 : (214200, 3)
items 데이터 행열 : (22170, 3)
items_categories 데이터 행열 : (84, 2)
shops 데이터 행열 : (60, 2)
```

데이터 살펴보기 (head())

In [5]:
```
train.head(3)
```

Out[5]:

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
0	02.01.2013	0	59	22154	999.0	1.0
1	03.01.2013	0	25	2552	899.0	1.0
2	05.01.2013	0	25	2552	899.0	-1.0

In [6]:
```
test.head(3)
```

Out[6]:

	ID	shop_id	item_id
0	0	5	5037
1	1	5	5320
2	2	5	5233

In [7]:
```
sub.head(3)
```

Out[7]:

	ID	item_cnt_month
0	0	0.5
1	1	0.5
2	2	0.5

In [8]:
```
items.head(3)
```

Out[8]:

	item_name	item_id	item_category_id
0	! ВО ВЛАСТИ НАВАЖДЕНИЯ (ПЛАСТ.) D	0	40
1	!ABBYY FineReader 12 Professional Edition Full...	1	76
2	***В ЛУЧАХ СЛАВЫ (UNV) D	2	40

In [9]:
```
items_cat.head(3)
```

Out[9]:

	item_category_name	item_category_id
0	PC - Гарнитуры/Наушники	0
1	Аксессуары - PS2	1
2	Аксессуары - PS3	2

In [10]:
```
shops.head(3)
```

Out[10]:

	shop_name	shop_id
0	!Якутск Орджоникидзе, 56 фран	0
1	!Якутск ТЦ "Центральный" фран	1
2	Адыгея ТЦ "Мега"	2

컬럼명 확인

In [11]:
```
print("\n 학습용 데이터 : {}".format(train.columns))
print("\n 제출용 데이터 : {}".format(sub.columns))
print("\n 테스트 데이터 : {}".format(test.columns))
print("\n items 데이터 : {}".format(items.columns))
print("\n items_categories 데이터 : {}".format(items_cat.columns))
print("\n shops 데이터 : {}".format(shops.columns))
```

```
학습용 데이터 : Index(['date', 'date_block_num', 'shop_id', 'item_id', 'item_price',
       'item_cnt_day'],
      dtype='object')

제출용 데이터 : Index(['ID', 'item_cnt_month'], dtype='object')

테스트 데이터 : Index(['ID', 'shop_id', 'item_id'], dtype='object')

items 데이터 : Index(['item_name', 'item_id', 'item_category_id'], dtype='object')

items_categories 데이터 : Index(['item_category_name', 'item_category_id'], dtype='object')

shops 데이터 : Index(['shop_name', 'shop_id'], dtype='object')
```

데이터 필드 설명

구분	컬럼명	설명	값
train	date	dd / mm / yyyy 형식의 날짜	날짜 데이터
train	date_block_num	편의를 위해 사용되는 연속 월 번호입니다.	2013/01(1)~2015/10(33)
train	shop_id	상점 고유 ID	0~59
train	item_id	항목 ID	0~22169
train	item_price	상품의 현재 가격	-1~307980
train	item_cnt_day	판매 된 제품 수입니다. 이 측정 값의 월별 금액을 예측하고 있습니다.	-22~2169
test	ID	테스트 예측을 위한 ID	0~214199
test	shop_id	상점 고유 ID	2~59
test	item_id	항목 ID	30~22167
sub	ID	테스트 예측을 위한 ID	0~214199
sub	item_cnt_month	예측해야 하는 값	default:0.5
items	item_name	항목 이름	범주의 개수(22170) '! ВО ВЛАСТИ НАВАЖДЕНИЯ

구분	컬럼명	설명	값
			(ПЛАСТ.) D', '!ABBYY FineReader 12 Professional Edition Full [PC, Цифровая версия]'
items	item_id	항목 ID	0~22169
items	item_category_id	항목 카테고리의 고유 식별자	0~83
items_categories	item_category_name	항목 카테고리 이름	범주의 개수(84) 'PC - Гарнитуры/Наушники' 'Аксессуары - PS2' 'Аксессуары - PS3' 'Аксессуары - PS4'
items_categories	item_category_id	항목 카테고리의 고유 식별자	0~83
shops	shop_name	상점 이름	범주의 개수(60)
shops	shop_id	상점 고유 ID	0~59

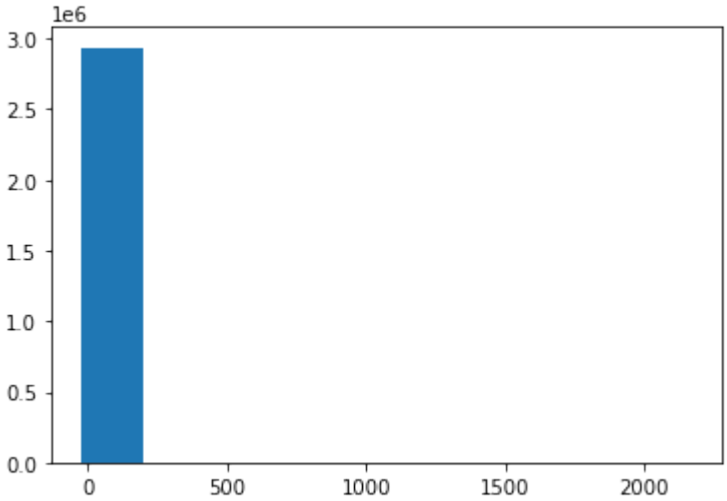
item_cnt_day 컬럼

- 판매 된 제품 수입니다.

In [13]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [14]:
```
plt.hist(train['item_cnt_day'])
```

Out[14]:
```
(array([2.93581e+06, 2.40000e+01, 1.10000e+01, 2.00000e+00, 1.00000e+00,
        0.00000e+00, 0.00000e+00, 0.00000e+00, 0.00000e+00, 1.00000e+00]),
 array([ -22. ,  197.1,  416.2,  635.3,  854.4, 1073.5, 1292.6, 1511.7,
        1730.8, 1949.9, 2169. ]),
 <a list of 10 Patch objects>)
```



In [15]:
```
train.describe()
```

Out[15]:

	date_block_num	shop_id	item_id	item_price	item_cnt_day
count	2.935849e+06	2.935849e+06	2.935849e+06	2.935849e+06	2.935849e+06
mean	1.456991e+01	3.300173e+01	1.019723e+04	8.908532e+02	1.242641e+00
std	9.422988e+00	1.622697e+01	6.324297e+03	1.729800e+03	2.618834e+00
min	0.000000e+00	0.000000e+00	0.000000e+00	-1.000000e+00	-2.200000e+01
25%	7.000000e+00	2.200000e+01	4.476000e+03	2.490000e+02	1.000000e+00
50%	1.400000e+01	3.100000e+01	9.343000e+03	3.990000e+02	1.000000e+00
75%	2.300000e+01	4.700000e+01	1.568400e+04	9.990000e+02	1.000000e+00
max	3.300000e+01	5.900000e+01	2.216900e+04	3.079800e+05	2.169000e+03

In [16]:
```
test.describe()
```

Out[16]:

	ID	shop_id	item_id
count	214200.000000	214200.000000	214200.000000
mean	107099.500000	31.642857	11019.398627
std	61834.358168	17.561933	6252.644590
min	0.000000	2.000000	30.000000
25%	53549.750000	16.000000	5381.500000
50%	107099.500000	34.500000	11203.000000
75%	160649.250000	47.000000	16071.500000
max	214199.000000	59.000000	22167.000000

In [17]:
```
sub.describe()
```

Out[17]:

	ID	item_cnt_month
count	214200.000000	214200.0
mean	107099.500000	0.5
std	61834.358168	0.0
min	0.000000	0.5
25%	53549.750000	0.5
50%	107099.500000	0.5
75%	160649.250000	0.5
max	214199.000000	0.5

In [18]:
```
print(items.describe())
print(items_cat.describe())
print(shops.describe())
```

```
            item_id  item_category_id
count  22170.00000      22170.000000
mean   11084.50000         46.290753
std     6400.07207         15.941486
min        0.00000          0.000000
25%     5542.25000         37.000000
50%    11084.50000         40.000000
75%    16626.75000         58.000000
max    22169.00000         83.000000
       item_category_id
count         84.000000
mean          41.500000
std           24.392622
min            0.000000
25%           20.750000
50%           41.500000
75%           62.250000
max           83.000000
         shop_id
count  60.000000
mean   29.500000
std    17.464249
min     0.000000
25%    14.750000
50%    29.500000
75%    44.250000
max    59.000000
```

총 카테고리 개수

In [19]:
```
def col_cat_col(col_name):
    num = len(col_name.unique() )
    print("범주의 개수 : {}".format(num) )
    print("List : ", col_name.unique() )
```

In [21]:
```
# 상점이름
col_cat_col(shops.shop_name)
```

```
범주의 개수 : 60
List :  ['!Якутск Орджоникидзе, 56 фран' '!Якутск ТЦ "Центральный" фран'
 'Адыгея ТЦ "Мега"' 'Балашиха ТРК "Октябрь-Киномир"'
 'Волжский ТЦ "Волга Молл"' 'Вологда ТРЦ "Мармелад"'
 'Воронеж (Плехановская, 13)' 'Воронеж ТРЦ "Максимир"'
 'Воронеж ТРЦ Сити-Парк "Град"' 'Выездная Торговля'
 'Жуковский ул. Чкалова 39м?' 'Жуковский ул. Чкалова 39м²'
 'Интернет-магазин ЧС' 'Казань ТЦ "Бехетле"' 'Казань ТЦ "ПаркХаус" II'
 'Калуга ТРЦ "XXI век"' 'Коломна ТЦ "Рио"' 'Красноярск ТЦ "Взлетка Плаза"'
 'Красноярск ТЦ "Июнь"' 'Курск ТЦ "Пушкинский"' 'Москва "Распродажа"'
 'Москва МТРЦ "Афи Молл"' 'Москва Магазин С21'
 'Москва ТК "Буденовский" (пав.А2)' 'Москва ТК "Буденовский" (пав.К7)'
 'Москва ТРК "Атриум"' 'Москва ТЦ "Ареал" (Беляево)'
 'Москва ТЦ "МЕГА Белая Дача II"' 'Москва ТЦ "МЕГА Теплый Стан" II'
 'Москва ТЦ "Новый век" (Новокосино)' 'Москва ТЦ "Перловский"'
 'Москва ТЦ "Семеновский"' 'Москва ТЦ "Серебряный Дом"'
 'Мытищи ТРК "XL-3"' 'Н.Новгород ТРЦ "РИО"' 'Н.Новгород ТРЦ "Фантастика"'
 'Новосибирск ТРЦ "Галерея Новосибирск"' 'Новосибирск ТЦ "Мега"'
 'Омск ТЦ "Мега"' 'РостовНаДону ТРК "Мегацентр Горизонт"'
 'РостовНаДону ТРК "Мегацентр Горизонт" Островной'
 'РостовНаДону ТЦ "Мега"' 'СПб ТК "Невский Центр"' 'СПб ТК "Сенная"'
 'Самара ТЦ "Мелодия"' 'Самара ТЦ "ПаркХаус"' 'Сергиев Посад ТЦ "7Я"'
 'Сургут ТРЦ "Сити Молл"' 'Томск ТРЦ "Изумрудный Город"'
 'Тюмень ТРЦ "Кристалл"' 'Тюмень ТЦ "Гудвин"' 'Тюмень ТЦ "Зеленый Берег"'
 'Уфа ТК "Центральный"' 'Уфа ТЦ "Семья" 2' 'Химки ТЦ "Мега"'
 'Цифровой склад 1С-Онлайн' 'Чехов ТРЦ "Карнавал"'
 'Якутск Орджоникидзе, 56' 'Якутск ТЦ "Центральный"'
 'Ярославль ТЦ "Альтаир"']
```

In [22]:
```
# 항목 카테고리 이름
col_cat_col(items_cat.item_category_name)
```

```
범주의 개수 : 84
List :  ['PC - Гарнитуры/Наушники' 'Аксессуары - PS2' 'Аксессуары - PS3'
 'Аксессуары - PS4' 'Аксессуары - PSP' 'Аксессуары - PSVita'
 'Аксессуары - XBOX 360' 'Аксессуары - XBOX ONE' 'Билеты (Цифра)'
 'Доставка товара' 'Игровые консоли - PS2' 'Игровые консоли - PS3'
 'Игровые консоли - PS4' 'Игровые консоли - PSP'
 'Игровые консоли - PSVita' 'Игровые консоли - XBOX 360'
 'Игровые консоли - XBOX ONE' 'Игровые консоли - Прочие' 'Игры - PS2'
 'Игры - PS3' 'Игры - PS4' 'Игры - PSP' 'Игры - PSVita' 'Игры - XBOX 360'
 'Игры - XBOX ONE' 'Игры - Аксессуары для игр' 'Игры Android - Цифра'
 'Игры MAC - Цифра' 'Игры PC - Дополнительные издания'
 'Игры PC - Коллекционные издания' 'Игры PC - Стандартные издания'
 'Игры PC - Цифра' 'Карты оплаты (Кино, Музыка, Игры)'
 'Карты оплаты - Live!' 'Карты оплаты - Live! (Цифра)'
 'Карты оплаты - PSN' 'Карты оплаты - Windows (Цифра)' 'Кино - Blu-Ray'
 'Кино - Blu-Ray 3D' 'Кино - Blu-Ray 4K' 'Кино - DVD'
 'Кино - Коллекционное' 'Книги - Артбуки, энциклопедии'
 'Книги - Аудиокниги' 'Книги - Аудиокниги (Цифра)' 'Книги - Аудиокниги 1С'
 'Книги - Бизнес литература' 'Книги - Комиксы, манга'
 'Книги - Компьютерная литература' 'Книги - Методические материалы 1С'
 'Книги - Открытки' 'Книги - Познавательная литература'
 'Книги - Путеводители' 'Книги - Художественная литература'
 'Книги - Цифра' 'Музыка - CD локального производства'
 'Музыка - CD фирменного производства' 'Музыка - MP3' 'Музыка - Винил'
 'Музыка - Музыкальное видео' 'Музыка - Подарочные издания'
 'Подарки - Атрибутика' 'Подарки - Гаджеты, роботы, спорт'
 'Подарки - Мягкие игрушки' 'Подарки - Настольные игры'
 'Подарки - Настольные игры (компактные)' 'Подарки - Открытки, наклейки'
 'Подарки - Развитие' 'Подарки - Сертификаты, услуги' 'Подарки - Сувениры'
 'Подарки - Сувениры (в навеску)'
 'Подарки - Сумки, Альбомы, Коврики д/мыши' 'Подарки - Фигурки'
 'Программы - 1С:Предприятие 8' 'Программы - MAC (Цифра)'
 'Программы - Для дома и офиса' 'Программы - Для дома и офиса (Цифра)'
 'Программы - Обучающие' 'Программы - Обучающие (Цифра)' 'Служебные'
 'Служебные - Билеты' 'Чистые носители (шпиль)'
 'Чистые носители (штучные)' 'Элементы питания']
```

In [23]:
```
# 항목 이름
col_cat_col(items.item_name)
```

```
범주의 개수 : 22170
List :  ['! ВО ВЛАСТИ НАВАЖДЕНИЯ (ПЛАСТ.)         D'
 '!ABBYY FineReader 12 Professional Edition Full [PC, Цифровая версия]'
 '***В ЛУЧАХ СЛАВЫ   (UNV)                     D' ...
 'Язык запросов 1С:Предприятия 8 (+CD). Хрусталева Е.Ю.'
 'Яйцо для Little Inu' 'Яйцо дракона (Игра престолов)']
```

날짜 확인 후, 열 생성

In [24]:
```
train['date'] = pd.to_datetime(train['date'],format = '%d.%m.%Y')
train['year'] = train['date'].dt.year
train['month'] = train['date'].dt.month
```

In [25]:
```
# item_cnt_day : 판매된 제품수, item_price : 총 합계 금액
# 년월별 통계
# date_block_num : 편의를 위해 사용되는 연속 월 번호
# shop_id,item_id  : 상점 ID, 아이템 ID
# item_price, item_cnt_day : 아이템 가격과 판매개수
```

```
sum_train = train.groupby( ['year', 'month'] ).sum()
sum_train
```

Out[25]:

year	month	date_block_num	shop_id	item_id	item_price	item_cnt_day
2013	1	0	3417068	1183971787	8.221187e+07	131479.0
	2	108613	3111582	1076043980	7.558019e+07	128090.0
	3	242694	4016457	1220911622	8.429831e+07	147142.0
	4	282327	3164978	971345965	6.151282e+07	107190.0
	5	367036	3093999	950372988	5.727413e+07	106970.0
	6	502015	3364700	1047351238	6.334361e+07	125381.0
	7	603288	3376156	1067060380	6.219681e+07	116966.0
	8	733404	3510787	1065970958	6.543817e+07	125291.0
	9	769096	3208314	957871641	7.270157e+07	133332.0
	10	847818	3101078	966066011	7.391497e+07	127541.0
	11	967360	3229598	974859705	7.960888e+07	130009.0
	12	1575706	4747485	1424369716	1.431799e+08	183342.0
2014	1	1192188	3317287	996727749	8.572518e+07	116899.0
	2	1167790	2982160	866201555	7.783305e+07	109687.0
	3	1298262	3117514	931084282	8.355875e+07	115297.0
	4	1168590	2608134	811885179	6.638839e+07	96556.0
	5	1256464	2649821	800026877	6.753802e+07	97790.0
	6	1400936	2774465	831994894	7.198654e+07	97429.0
	7	1417680	2671321	799658913	6.671338e+07	91280.0
	8	1645666	2948523	879409093	7.681620e+07	102721.0
	9	1463140	2462241	739754807	7.735488e+07	99208.0
	10	1666581	2621774	817071654	8.488921e+07	107422.0
	11	1901416	2899400	853857509	1.053448e+08	117845.0
	12	3008078	4339498	1330492992	1.794057e+08	168755.0
2015	1	2124528	2991797	906227293	1.005736e+08	110971.0
	2	1795200	2405142	753397832	7.255950e+07	84198.0
	3	1819402	2365258	737212776	7.145842e+07	82014.0
	4	1519398	1827819	589741486	5.897193e+07	77827.0
	5	1527344	1764840	566095394	5.874271e+07	72295.0
	6	1583893	1763694	568759443	5.573324e+07	64114.0
	7	1666470	1823463	589694328	5.370635e+07	63187.0
	8	1767899	1883430	579633738	5.425592e+07	66079.0

year	month	date_block_num	shop_id	item_id	item_price	item_cnt_day
	9	1618816	1638284	504444591	5.906161e+07	72843.0
	10	1765962	1690024	577950482	6.553190e+07	71056.0

In [26]:
```
# 3개 열을 삭제하고 남는 2개열
# item_cnt_day : 판매된 제품수, item_price : 총 합계 금액
sum_train = sum_train.drop(['date_block_num', 'shop_id', 'item_id'], axis=1)
sum_train
```

Out[26]:

year	month	item_price	item_cnt_day
2013	**1**	8.221187e+07	131479.0
	2	7.558019e+07	128090.0
	3	8.429831e+07	147142.0
	4	6.151282e+07	107190.0
	5	5.727413e+07	106970.0
	6	6.334361e+07	125381.0
	7	6.219681e+07	116966.0
	8	6.543817e+07	125291.0
	9	7.270157e+07	133332.0
	10	7.391497e+07	127541.0
	11	7.960888e+07	130009.0
	12	1.431799e+08	183342.0
2014	**1**	8.572518e+07	116899.0
	2	7.783305e+07	109687.0
	3	8.355875e+07	115297.0
	4	6.638839e+07	96556.0
	5	6.753802e+07	97790.0
	6	7.198654e+07	97429.0
	7	6.671338e+07	91280.0
	8	7.681620e+07	102721.0
	9	7.735488e+07	99208.0
	10	8.488921e+07	107422.0
	11	1.053448e+08	117845.0
	12	1.794057e+08	168755.0
2015	**1**	1.005736e+08	110971.0
	2	7.255950e+07	84198.0
	3	7.145842e+07	82014.0
	4	5.897193e+07	77827.0
	5	5.874271e+07	72295.0
	6	5.573324e+07	64114.0
	7	5.370635e+07	63187.0
	8	5.425592e+07	66079.0

		item_price	item_cnt_day
year	month		
	9	5.906161e+07	72843.0
	10	6.553190e+07	71056.0

sum_train과 원본 train 데이터합치기

In [27]:
```
train.head()
```

Out[27]:

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day	year	month
0	2013-01-02	0	59	22154	999.00	1.0	2013	1
1	2013-01-03	0	25	2552	899.00	1.0	2013	1
2	2013-01-05	0	25	2552	899.00	-1.0	2013	1
3	2013-01-06	0	25	2554	1709.05	1.0	2013	1
4	2013-01-15	0	25	2555	1099.00	1.0	2013	1

In [28]:
```
sum_train.head()
```

Out[28]:

		item_price	item_cnt_day
year	month		
2013	1	8.221187e+07	131479.0
	2	7.558019e+07	128090.0
	3	8.429831e+07	147142.0
	4	6.151282e+07	107190.0
	5	5.727413e+07	106970.0

In [29]:
```
test.head()
```

Out[29]:

	ID	shop_id	item_id
0	0	5	5037
1	1	5	5320
2	2	5	5233
3	3	5	5232
4	4	5	5268

In [32]:
```
test.tail()
```

Out[32]:

	ID	shop_id	item_id
214195	214195	45	18454
214196	214196	45	16188
214197	214197	45	15757
214198	214198	45	19648
214199	214199	45	969

year, month를 기준 열로 삼아, 두 데이터 셋을 합친다.

- train, sum_train

In [30]:
```
train_df = pd.merge(left=train,
                    right=sum_train,
                    how='left', on=['year', 'month'], sort=False)
train_df
```

Out[30]:

	date	date_block_num	shop_id	item_id	item_price_x	item_cnt_day_x	year
0	2013-01-02	0	59	22154	999.00	1.0	2013
1	2013-01-03	0	25	2552	899.00	1.0	2013
2	2013-01-05	0	25	2552	899.00	-1.0	2013
3	2013-01-06	0	25	2554	1709.05	1.0	2013
4	2013-01-15	0	25	2555	1099.00	1.0	2013
...	...	...	...	...	...	...	...
2935844	2015-10-10	33	25	7409	299.00	1.0	2015
2935845	2015-10-09	33	25	7460	299.00	1.0	2015
2935846	2015-10-14	33	25	7459	349.00	1.0	2015
2935847	2015-10-22	33	25	7440	299.00	1.0	2015
2935848	2015-10-03	33	25	7460	299.00	1.0	2015

2935849 rows × 10 columns

모델 선택, 학습 및 예측

In [33]:
```
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
```

입력, 출력 열 선택

In [34]:
```
# 변수 선택 및 데이터 지정
sel = ['shop_id', 'item_id']
X_tr_all = train_df[sel]
X_test_all = test[sel]
```

In [36]:
```
label = "item_cnt_day_y"
y_tr_all = train_df[label]
```

In [37]:
```
X_train, X_test, y_train, y_test = train_test_split(X_tr_all, y_tr_all,
                                                    random_state=77)
```

In [38]:
```
model = LinearRegression()  # 모델 생성
model.fit(X_train, y_train)  # 모델 훈련

model.score(X_test, y_test)
```

Out[38]:
```
0.00028839456379425865
```

In [39]:
```
# %%time

model_rf = RandomForestRegressor(max_depth=4)  # 모델 생성
model_rf.fit(X_train, y_train)  # 모델 훈련

model_rf.score(X_test, y_test)
```

Out[39]:
```
0.007747604469701019
```

In [41]:
```
model = RandomForestRegressor(max_depth=4)  # 모델 생성
model.fit(X_train, y_train)  # 모델 훈련
pred = model.predict(X_test_all) # 모델로 예측

sub['item_cnt_month'] = pred
sub
```

Out[41]:

	ID	item_cnt_month
0	0	115062.513818
1	1	115062.513818
2	2	115062.513818
3	3	115062.513818
4	4	115062.513818
...	...	...
214195	214195	113599.930714
214196	214196	113599.930714
214197	214197	113599.930714
214198	214198	115731.648109
214199	214199	114911.549108

214200 rows × 2 columns

제출

In [43]:
```
sub.to_csv("firstSub.csv", index=False)
```