```
{"metadata":{"kernelspec":{"language":"python","display_name":"Python
3","name":"python3"},"language_info":{"name":"python","version":"3.6.6","
mimetype":"text/x-
python","codemirror_mode":{"name":"ipython","version":3},"pygments_lexer"
:"ipython3","nbconvert_exporter":"python","file_extension":".py"}},"nbfor
mat_minor":4,"nbformat":4,"cells":[{"cell_type":"markdown","source":"# 1.
Importing Packages and Collecting Data, Defining Evaluation
","metadata":{}},{"cell_type":"markdown","source":"## 1.1 Importing
Packages","metadata":{}},{"cell_type":"code","source":"'''Importing Data
Manipulattion Modules'''\nimport re\nimport numpy as np\nimport pandas as
pd\nfrom pandas.core.dtypes.dtypes import CategoricalDtype\nfrom
scipy.stats import norm, skew\nfrom scipy.special import
boxcox1p\npd.set_option('display.float_format', lambda x:
'{:.3f}'.format(x))\npd.set_option(\"display.max_columns\",
81)\npd.set_option(\"display.max_rows\",
101)\npd.set_option(\"display.max_colwidth\", 100)\n\n'''Seaborn and
Matplotlib Visualization'''\nimport matplotlib.pyplot as plt\nimport
seaborn as sns\nplt.style.use('bmh')
\nsns.set_style({'axes.grid':False})
\nsns.set_style('whitegrid')\n%matplotlib
inline\n\n'''Validation'''\nfrom sklearn.model_selection import KFold,
cross_val_score\n\n'''Ignore deprecation and future, and user
warnings.'''\nimport warnings as wrn\nwrn.filterwarnings('ignore',
category = DeprecationWarning) \nwrn.filterwarnings('ignore', category =
FutureWarning) \nwrn.filterwarnings('ignore', category = UserWarning)
","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:26.198114Z","iopub.execute_input":"2023-03-
28T02:52:26.198494Z","iopub.status.idle":"2023-03-
28T02:52:28.030117Z","shell.execute_reply.started":"2023-03-
28T02:52:26.198436Z","shell.execute_reply":"2023-03-
28T02:52:28.029025Z"},"trusted":true},"execution_count":1,"outputs":[]},{
"cell_type":"markdown","source":"## 1.2 Collecting
Data","metadata":{}},{"cell_type":"code","source":"'''Check the
files'''\nfrom subprocess import
check_output\nprint(check_output([\"ls\", \"../input/grupo-bimbo-
inventory-
demand/\"]).decode(\"utf8\"))","metadata":{"execution":{"iopub.status.bus
y":"2023-03-28T02:52:28.034234Z","iopub.execute_input":"2023-03-
28T02:52:28.034585Z","iopub.status.idle":"2023-03-
28T02:52:28.057301Z","shell.execute_reply.started":"2023-03-
28T02:52:28.034528Z","shell.execute_reply":"2023-03-
28T02:52:28.056198Z"},"trusted":true},"execution_count":2,"outputs":[{"na
me":"stdout","text":"cliente_tabla.csv.zip\nproducto_tabla.csv.zip\nsampl
e_submission.csv.zip\ntest.csv.zip\ntown_state.csv.zip\ntrain.csv.zip\n\n
","output_type":"stream"}]},{"cell_type":"markdown","source":"###
town_state","metadata":{}},{"cell_type":"code","source":"dtype =
{'Agencia_ID': 'int16'}\ntown_state_df = pd.read_csv(\"../input/grupo-
bimbo-inventory-demand/town_state.csv.zip\",
dtype=dtype)\n\ntown_state_df['Town_ID'] =
town_state_df.Town.apply(lambda x: x.split('
')[0]).astype('int16')\ntown_state_df['Town_name'] =
town_state_df.Town.apply(lambda x: ' '.join(x.split('
')[1:]))\n\nprint(f'town_state\\'s shape:
{town_state_df.shape}')\ntown_state_df.head(5)","metadata":{"execution":{
"iopub.status.busy":"2023-03-
28T02:52:28.059013Z","iopub.execute_input":"2023-03-
28T02:52:28.059371Z","iopub.status.idle":"2023-03-
28T02:52:28.111853Z","shell.execute_reply.started":"2023-03-
28T02:52:28.059302Z","shell.execute_reply":"2023-03-
28T02:52:28.110857Z"},"trusted":true},"execution_count":3,"outputs":[{"na
me":"stdout","text":"town_state's shape: (790,
5)\n","output_type":"stream"},{"execution_count":3,"output_type":"execute
_result","data":{"text/plain":"   Agencia_ID                  Town
State  Town_ID  \\\n0        1110     2008 AG. LAGO FILT      MÉXICO,
D.F.     2008   \n1        1111  2002 AG. AZCAPOTZALCO      MÉXICO, D.F.
2002   \n2        1112    2004 AG. CUAUTITLAN  ESTADO DE MÉXICO     2004
\n3        1113     2008 AG. LAGO FILT      MÉXICO, D.F.     2008   \n4
1114   2029 AG.IZTAPALAPA 2      MÉXICO, D.F.     2029   \n\n
Town_name  \n0     AG. LAGO FILT  \n1  AG. AZCAPOTZALCO  \n2    AG.
CUAUTITLAN  \n3     AG. LAGO FILT  \n4   AG.IZTAPALAPA 2
","text/html":"<div>\n<style scoped>\n    .dataframe tbody tr th:only-of-
type {\n        vertical-align: middle;\n    }\n\n    .dataframe tbody tr
th {\n        vertical-align: top;\n    }\n\n    .dataframe thead th {\n
text-align: right;\n    }\n</style>\n<table border=\"1\"
class=\"dataframe\">\n  <thead>\n    <tr style=\"text-align: right;\">\n
<th></th>\n      <th>Agencia_ID</th>\n      <th>Town</th>\n
<th>State</th>\n      <th>Town_ID</th>\n      <th>Town_name</th>\n
</tr>\n  </thead>\n  <tbody>\n    <tr>\n      <th>0</th>\n
<td>1110</td>\n      <td>2008 AG. LAGO FILT</td>\n      <td>MÉXICO,
D.F.</td>\n      <td>2008</td>\n      <td>AG. LAGO FILT</td>\n    </tr>\n
<tr>\n      <th>1</th>\n      <td>1111</td>\n      <td>2002 AG.
AZCAPOTZALCO</td>\n      <td>MÉXICO, D.F.</td>\n      <td>2002</td>\n
<td>AG. AZCAPOTZALCO</td>\n    </tr>\n    <tr>\n      <th>2</th>\n
<td>1112</td>\n      <td>2004 AG. CUAUTITLAN</td>\n      <td>ESTADO DE
MÉXICO</td>\n      <td>2004</td>\n      <td>AG. CUAUTITLAN</td>\n
</tr>\n    <tr>\n      <th>3</th>\n      <td>1113</td>\n      <td>2008
AG. LAGO FILT</td>\n      <td>MÉXICO, D.F.</td>\n      <td>2008</td>\n
<td>AG. LAGO FILT</td>\n    </tr>\n    <tr>\n      <th>4</th>\n
<td>1114</td>\n      <td>2029 AG.IZTAPALAPA 2</td>\n      <td>MÉXICO,
D.F.</td>\n      <td>2029</td>\n      <td>AG.IZTAPALAPA 2</td>\n
</tr>\n
</tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell_type":"code","source
":"town_state_df.info()","metadata":{"execution":{"iopub.status.busy":"20
23-03-28T02:52:28.113452Z","iopub.execute_input":"2023-03-
28T02:52:28.114035Z","iopub.status.idle":"2023-03-
28T02:52:28.128319Z","shell.execute_reply.started":"2023-03-
28T02:52:28.113971Z","shell.execute_reply":"2023-03-
28T02:52:28.127230Z"},"trusted":true},"execution_count":4,"outputs":[{"na
me":"stdout","text":"<class 'pandas.core.frame.DataFrame'>\nRangeIndex:
790 entries, 0 to 789\nData columns (total 5 columns):\nAgencia_ID    790
non-null int16\nTown          790 non-null object\nState         790 non-
null object\nTown_ID       790 non-null int16\nTown_name     790 non-null
object\ndtypes: int16(2), object(3)\nmemory usage: 21.7+
KB\n","output_type":"stream"}]},{"cell_type":"markdown","source":"###
product","metadata":{}},{"cell_type":"code","source":"dtype =
{'Producto_ID': 'int32'}\nproduct_df = pd.read_csv(\"../input/grupo-
bimbo-inventory-demand/producto_tabla.csv.zip\",
dtype=dtype)\n\nproduct_df['popular_name'] =
product_df.NombreProducto.str.extract(r'^(.*?)(\\d*\\s\\d+(kg|Kg|g|G|ml|
ml|p|Reb)\\s)', expand=False)[0]\nproduct_df['property'] =
product_df.NombreProducto.str.extract(r'^.*\\d+(kg|Kg|g|G|ml|
ml|p|Reb)\\s(.*?)\\s\\d+$', expand=False)[1]\nproduct_df['unit'] =
product_df.NombreProducto.str.extract(r'(\\d*\\s\\d+(kg|Kg|g|G|ml| ml))',
expand=False)[0]\nproduct_df['pieces'] =
product_df.NombreProducto.str.extract('(\\d+(p|Reb)) ',
expand=False)[0]\n\nprint(f'product\\'s shape:
```

```
{product_df.shape}')\nproduct_df.head(5)","metadata":{"execution":{"iopub
.status.busy":"2023-03-28T02:52:28.131436Z","iopub.execute_input":"2023-
03-28T02:52:28.131887Z","iopub.status.idle":"2023-03-
28T02:52:28.208607Z","shell.execute_reply.started":"2023-03-
28T02:52:28.131823Z","shell.execute_reply":"2023-03-
28T02:52:28.207456Z"},"trusted":true},"execution_count":5,"outputs":[{"na
me":"stdout","text":"product's shape: (2592,
6)\n","output_type":"stream"},{"execution_count":5,"output_type":"execute
_result","data":{"text/plain":"   Producto_ID
NombreProducto  \\\n0            0                      NO IDENTIFICADO
0   \n1           9            Capuccino Moka 750g NES 9   \n2
41  Bimbollos Ext sAjonjoli 6p 480g BIM 41   \n3           53
Burritos Sincro 170g CU LON 53   \n4           72     Div Tira Mini
Doradita 4p 45g TR 72   \n\n              popular_name property   unit
pieces  \n0                        NaN      NaN    NaN    NaN  \n1
Capuccino Moka      NES   750g    NaN  \n2  Bimbollos Ext sAjonjoli
BIM   480g     6p  \n3          Burritos Sincro   CU LON   170g    NaN
\n4   Div Tira Mini Doradita       TR    45g     4p
","text/html":"<div>\n<style scoped>\n    .dataframe tbody tr th:only-of-
type {\n        vertical-align: middle;\n    }\n\n    .dataframe tbody tr
th {\n        vertical-align: top;\n    }\n\n    .dataframe thead th {\n
text-align: right;\n    }\n</style>\n<table border=\"1\"
class=\"dataframe\">\n  <thead>\n    <tr style=\"text-align: right;\">\n
<th></th>\n      <th>Producto_ID</th>\n      <th>NombreProducto</th>\n
<th>popular_name</th>\n      <th>property</th>\n      <th>unit</th>\n
<th>pieces</th>\n    </tr>\n  </thead>\n  <tbody>\n    <tr>\n
<th>0</th>\n      <td>0</td>\n      <td>NO IDENTIFICADO 0</td>\n
<td>NaN</td>\n      <td>NaN</td>\n      <td>NaN</td>\n
<td>NaN</td>\n    </tr>\n    <tr>\n      <th>1</th>\n      <td>9</td>\n
<td>Capuccino Moka 750g NES 9</td>\n      <td>Capuccino Moka</td>\n
<td>NES</td>\n      <td>750g</td>\n      <td>NaN</td>\n    </tr>\n
<tr>\n      <th>2</th>\n      <td>41</td>\n      <td>Bimbollos Ext
sAjonjoli 6p 480g BIM 41</td>\n      <td>Bimbollos Ext sAjonjoli</td>\n
<td>BIM</td>\n      <td>480g</td>\n      <td>6p</td>\n    </tr>\n
<tr>\n      <th>3</th>\n      <td>53</td>\n      <td>Burritos Sincro 170g
CU LON 53</td>\n      <td>Burritos Sincro</td>\n      <td>CU LON</td>\n
<td>170g</td>\n      <td>NaN</td>\n    </tr>\n    <tr>\n
<th>4</th>\n      <td>72</td>\n      <td>Div Tira Mini Doradita 4p 45g TR
72</td>\n      <td>Div Tira Mini Doradita</td>\n      <td>TR</td>\n
<td>45g</td>\n      <td>4p</td>\n    </tr>\n
</tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell_type":"code","source
":"product_df.info()","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:28.212254Z","iopub.execute_input":"2023-03-
28T02:52:28.212673Z","iopub.status.idle":"2023-03-
28T02:52:28.226406Z","shell.execute_reply.started":"2023-03-
28T02:52:28.212600Z","shell.execute_reply":"2023-03-
28T02:52:28.225093Z"},"trusted":true},"execution_count":6,"outputs":[{"na
me":"stdout","text":"<class 'pandas.core.frame.DataFrame'>\nRangeIndex:
2592 entries, 0 to 2591\nData columns (total 6 columns):\nProducto_ID
2592 non-null int32\nNombreProducto    2592 non-null object\npopular_name
2565 non-null object\nproperty         2571 non-null object\nunit
2532 non-null object\npieces           1132 non-null object\ndtypes:
int32(1), object(5)\nmemory usage: 111.5+
KB\n","output_type":"stream"}]},{"cell_type":"markdown","source":"###
client","metadata":{}},{"cell_type":"code","source":"dtype =
{'Cliente_ID': 'int32'}\nclient_df = pd.read_csv(\"../input/grupo-bimbo-
inventory-demand/cliente_tabla.csv.zip\", dtype=dtype)\n\ndup_sr =
client_df.groupby('Cliente_ID')['Cliente_ID'].count().astype('int8')\ndup
```

```
_sr.name = 'dup_num'\nclient_df = pd.merge(client_df,
pd.DataFrame(dup_sr).reset_index())\ndel dup_sr\n\nprint(f'client\\'s
shape:
{client_df.shape}')\nclient_df.head(5)","metadata":{"execution":{"iopub.s
tatus.busy":"2023-03-28T02:52:28.228101Z","iopub.execute_input":"2023-03-
28T02:52:28.228484Z","iopub.status.idle":"2023-03-
28T02:52:29.715406Z","shell.execute_reply.started":"2023-03-
28T02:52:28.228426Z","shell.execute_reply":"2023-03-
28T02:52:29.713935Z"},"trusted":true},"execution_count":7,"outputs":[{"na
me":"stdout","text":"client's shape: (935362,
3)\n","output_type":"stream"},{"execution_count":7,"output_type":"execute
_result","data":{"text/plain":"   Cliente_ID
NombreCliente  dup_num\n0           0                              SIN
NOMBRE        1\n1           1                         OXXO XINANTECATL
1\n2           2                           SIN NOMBRE        1\n3
3                              EL MORENO        1\n4           4  SDN
SER  DE ALIM  CUERPO SA CIA  DE INT        2","text/html":"<div>\n<style
scoped>\n    .dataframe tbody tr th:only-of-type {\n        vertical-
align: middle;\n    }\n\n    .dataframe tbody tr th {\n        vertical-
align: top;\n    }\n\n    .dataframe thead th {\n        text-align:
right;\n    }\n</style>\n<table border=\"1\" class=\"dataframe\">\n
<thead>\n    <tr style=\"text-align: right;\">\n      <th></th>\n
<th>Cliente_ID</th>\n      <th>NombreCliente</th>\n
<th>dup_num</th>\n    </tr>\n  </thead>\n  <tbody>\n    <tr>\n
<th>0</th>\n      <td>0</td>\n      <td>SIN NOMBRE</td>\n
<td>1</td>\n    </tr>\n    <tr>\n      <th>1</th>\n      <td>1</td>\n
<td>OXXO XINANTECATL</td>\n      <td>1</td>\n    </tr>\n    <tr>\n
<th>2</th>\n      <td>2</td>\n      <td>SIN NOMBRE</td>\n
<td>1</td>\n    </tr>\n    <tr>\n      <th>3</th>\n      <td>3</td>\n
<td>EL MORENO</td>\n      <td>1</td>\n    </tr>\n    <tr>\n
<th>4</th>\n      <td>4</td>\n      <td>SDN SER  DE ALIM  CUERPO SA CIA
DE INT</td>\n      <td>2</td>\n    </tr>\n
</tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell_type":"code","source
":"client_df.info()","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:29.717971Z","iopub.execute_input":"2023-03-
28T02:52:29.718437Z","iopub.status.idle":"2023-03-
28T02:52:29.781111Z","shell.execute_reply.started":"2023-03-
28T02:52:29.718370Z","shell.execute_reply":"2023-03-
28T02:52:29.780050Z"},"trusted":true},"execution_count":8,"outputs":[{"na
me":"stdout","text":"<class 'pandas.core.frame.DataFrame'>\nInt64Index:
935362 entries, 0 to 935361\nData columns (total 3 columns):\nCliente_ID
935362 non-null int32\nNombreCliente    935362 non-null object\ndup_num
935362 non-null int8\ndtypes: int32(1), int8(1), object(1)\nmemory usage:
18.7+
MB\n","output_type":"stream"}]},{"cell_type":"markdown","source":"###
test","metadata":{}},{"cell_type":"code","source":"dtype = {\n    'id':
'int32',\n    'Semana': 'int8',\n    'Agencia_ID': 'int16',\n
'Canal_ID': 'int8',\n    'Ruta_SAK': 'int16',\n    'Cliente_ID':
'int32',\n    'Producto_ID': 'int32',\n}\ntest_df =
pd.read_csv(\"../input/grupo-bimbo-inventory-demand/test.csv.zip\",
dtype=dtype)\n\nprint(f'test\\'s shape:
{test_df.shape}')\ntest_df.head()","metadata":{"execution":{"iopub.status
.busy":"2023-03-28T02:52:29.782468Z","iopub.execute_input":"2023-03-
28T02:52:29.782748Z","iopub.status.idle":"2023-03-
28T02:52:35.855339Z","shell.execute_reply.started":"2023-03-
28T02:52:29.782701Z","shell.execute_reply":"2023-03-
28T02:52:35.854520Z"},"trusted":true},"execution_count":9,"outputs":[{"na
me":"stdout","text":"test's shape: (6999251,
7)\n","output_type":"stream"},{"execution_count":9,"output_type":"execute
_result","data":{"text/plain":"   id  Semana  Agencia_ID  Canal_ID
Ruta_SAK  Cliente_ID  Producto_ID\n0   0      11        4037         1
2209     4639078        35305\n1   1      11        2237         1
1226     4705135         1238\n2   2      10        2045         1
2831     4549769        32940\n3   3      11        1227         1
4448     4717855        43066\n4   4      11        1219         1
1130      966351         1277","text/html":"<div>\n<style scoped>\n
.dataframe tbody tr th:only-of-type {\n        vertical-align: middle;\n
}\n\n    .dataframe tbody tr th {\n        vertical-align: top;\n
}\n\n    .dataframe thead th {\n        text-align: right;\n
}\n</style>\n<table border=\"1\" class=\"dataframe\">\n  <thead>\n    <tr
style=\"text-align: right;\">\n      <th></th>\n      <th>id</th>\n
<th>Semana</th>\n      <th>Agencia_ID</th>\n      <th>Canal_ID</th>\n
<th>Ruta_SAK</th>\n      <th>Cliente_ID</th>\n
<th>Producto_ID</th>\n    </tr>\n  </thead>\n  <tbody>\n    <tr>\n
<th>0</th>\n      <td>0</td>\n      <td>11</td>\n      <td>4037</td>\n
<td>1</td>\n      <td>2209</td>\n      <td>4639078</td>\n
<td>35305</td>\n    </tr>\n    <tr>\n      <th>1</th>\n      <td>1</td>\n
<td>11</td>\n      <td>2237</td>\n      <td>1</td>\n      <td>1226</td>\n
<td>4705135</td>\n      <td>1238</td>\n    </tr>\n    <tr>\n
<th>2</th>\n      <td>2</td>\n      <td>10</td>\n      <td>2045</td>\n
<td>1</td>\n      <td>2831</td>\n      <td>4549769</td>\n
<td>32940</td>\n    </tr>\n    <tr>\n      <th>3</th>\n      <td>3</td>\n
<td>11</td>\n      <td>1227</td>\n      <td>1</td>\n      <td>4448</td>\n
<td>4717855</td>\n      <td>43066</td>\n    </tr>\n    <tr>\n
<th>4</th>\n      <td>4</td>\n      <td>11</td>\n      <td>1219</td>\n
<td>1</td>\n      <td>1130</td>\n      <td>966351</td>\n
<td>1277</td>\n    </tr>\n
</tbody>\n</table>\n</div>"},"metadata":{}}]},{"cell_type":"code","source
":"test_df.info()","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:35.857081Z","iopub.execute_input":"2023-03-
28T02:52:35.857395Z","iopub.status.idle":"2023-03-
28T02:52:35.865803Z","shell.execute_reply.started":"2023-03-
28T02:52:35.857334Z","shell.execute_reply":"2023-03-
28T02:52:35.864691Z"},"trusted":true},"execution_count":10,"outputs":[{"n
ame":"stdout","text":"<class 'pandas.core.frame.DataFrame'>\nRangeIndex:
6999251 entries, 0 to 6999250\nData columns (total 7 columns):\nid
int32\nSemana         int8\nAgencia_ID     int16\nCanal_ID
int8\nRuta_SAK       int16\nCliente_ID     int32\nProducto_ID
int32\ndtypes: int16(2), int32(3), int8(2)\nmemory usage: 120.2
MB\n","output_type":"stream"}]},{"cell_type":"markdown","source":"###
train","metadata":{}},{"cell_type":"code","source":"dtype = {\n
'Semana': 'int8',\n    'Agencia_ID': 'int16',\n    'Canal_ID': 'int8',\n
'Ruta_SAK': 'int16',\n    'Cliente_ID': 'int32',\n    'Producto_ID':
'int32',\n    'Venta_uni_hoy': 'int16',\n    'Venta_hoy': 'float32',\n
'Dev_uni_proxima': 'int32',\n    'Dev_proxima': 'float32',\n
'Demanda_uni_equil': 'int16',\n}\ntrain_df =
pd.read_csv(\"../input/grupo-bimbo-inventory-demand/train.csv.zip\",
dtype=dtype)\n\nprint(f'train\\'s shape:
{train_df.shape}')\ntrain_df.head()","metadata":{"execution":{"iopub.stat
us.busy":"2023-03-28T02:52:35.867620Z","iopub.execute_input":"2023-03-
28T02:52:35.868007Z","iopub.status.idle":"2023-03-
28T02:52:45.453004Z","shell.execute_reply.started":"2023-03-
28T02:52:35.867941Z","shell.execute_reply":"2023-03-
28T02:52:45.447328Z"},"trusted":true},"execution_count":11,"outputs":[{"t
raceback":["\u001b[0;31m---------------------------------------------------
-------------------------\u001b[0m","\u001b[0;31mParserError\u001b[0m
```

```
Traceback (most recent call last)","\u001b[0;32m<ipython-input-11-
55d3f5ead550>\u001b[0m in
\u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[1;32m     12\u001b[0m
\u001b[0;34m'Demanda_uni_equil'\u001b[0m\u001b[0;34m:\u001b[0m
\u001b[0;34m'int16'\u001b[0m\u001b[0;34m,\u001b[0m\u001b[0;34m\u001b[0m\u
001b[0m\n\u001b[1;32m     13\u001b[0m }\n\u001b[0;32m---> 14\u001b[0;31m
\u001b[0mtrain_df\u001b[0m \u001b[0;34m=\u001b[0m
\u001b[0mpd\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mread_csv\u001b[0m\u001
b[0;34m(\u001b[0m\u001b[0;34m\"../input/grupo-bimbo-inventory-
demand/train.csv.zip\"\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mdtype\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0mdtype\u001b[0m\u001
b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m
15\u001b[0m \u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m     16\u001b[0m
\u001b[0mprint\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34mf'train\\'s
shape:
{train_df.shape}'\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u00
1b[0m\n","\u001b[0;32m/opt/conda/lib/python3.6/site-
packages/pandas/io/parsers.py\u001b[0m in
\u001b[0;36mparser_f\u001b[0;34m(filepath_or_buffer, sep, delimiter,
header, names, index_col, usecols, squeeze, prefix, mangle_dupe_cols,
dtype, engine, converters, true_values, false_values, skipinitialspace,
skiprows, skipfooter, nrows, na_values, keep_default_na, na_filter,
verbose, skip_blank_lines, parse_dates, infer_datetime_format,
keep_date_col, date_parser, dayfirst, cache_dates, iterator, chunksize,
compression, thousands, decimal, lineterminator, quotechar, quoting,
doublequote, escapechar, comment, encoding, dialect, error_bad_lines,
warn_bad_lines, delim_whitespace, low_memory, memory_map,
float_precision)\u001b[0m\n\u001b[1;32m    683\u001b[0m
)\n\u001b[1;32m    684\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[0;32m--> 685\u001b[0;31m
\u001b[0;32mreturn\u001b[0m
\u001b[0m_read\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mfilepath_or_buffer\
u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mkwds\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0
m\n\u001b[0m\u001b[1;32m    686\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    687\u001b[0m
\u001b[0mparser_f\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m__name__\u001b[0
m \u001b[0;34m=\u001b[0m
\u001b[0mname\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n","\u001b[0;32m/opt
/conda/lib/python3.6/site-packages/pandas/io/parsers.py\u001b[0m in
\u001b[0;36m_read\u001b[0;34m(filepath_or_buffer,
kwds)\u001b[0m\n\u001b[1;32m    461\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m    462\u001b[0m
\u001b[0;32mtry\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b
[0m\n\u001b[0;32m--> 463\u001b[0;31m         \u001b[0mdata\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mparser\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mread\u001b[0m\u001
b[0;34m(\u001b[0m\u001b[0mnrows\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34
m\u001b[0m\u001b[0m\n\u001b[0m\u001b[1;32m    464\u001b[0m
\u001b[0;32mfinally\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u
001b[0m\n\u001b[1;32m    465\u001b[0m
\u001b[0mparser\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0mclose\u001b[0m\u00
1b[0;34m(\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n"
,"\u001b[0;32m/opt/conda/lib/python3.6/site-
packages/pandas/io/parsers.py\u001b[0m in
\u001b[0;36mread\u001b[0;34m(self, nrows)\u001b[0m\n\u001b[1;32m
1152\u001b[0m     \u001b[0;32mdef\u001b[0m
\u001b[0mread\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[0m\u001b[
0;34m,\u001b[0m
\u001b[0mnrows\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mNone\u001b[0m\u0
01b[0;34m)\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n
\u001b[1;32m   1153\u001b[0m         \u001b[0mnrows\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0m_validate_integer\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0;34m\"nr
ows\"\u001b[0m\u001b[0;34m,\u001b[0m
\u001b[0mnrows\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[
0m\n\u001b[0;32m-> 1154\u001b[0;31m         \u001b[0mret\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_engine\u001b[0m\u00
1b[0;34m.\u001b[0m\u001b[0mread\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnr
ows\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[
0m\u001b[1;32m   1155\u001b[0m
\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[1;32m   1156\u001b[0m
\u001b[0;31m# May alter columns /
col_dict\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n","
\u001b[0;32m/opt/conda/lib/python3.6/site-
packages/pandas/io/parsers.py\u001b[0m in
\u001b[0;36mread\u001b[0;34m(self, nrows)\u001b[0m\n\u001b[1;32m
2046\u001b[0m     \u001b[0;32mdef\u001b[0m
\u001b[0mread\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mself\u001b[0m\u001b[
0;34m,\u001b[0m
\u001b[0mnrows\u001b[0m\u001b[0;34m=\u001b[0m\u001b[0;32mNone\u001b[0m\u0
01b[0;34m)\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n
\u001b[1;32m   2047\u001b[0m
\u001b[0;32mtry\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b
[0m\n\u001b[0;32m-> 2048\u001b[0;31m           \u001b[0mdata\u001b[0m
\u001b[0;34m=\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_reader\u001b[0m\u00
1b[0;34m.\u001b[0m\u001b[0mread\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mnr
ows\u001b[0m\u001b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u001b[
0m\u001b[1;32m   2049\u001b[0m         \u001b[0;32mexcept\u001b[0m
\u001b[0mStopIteration\u001b[0m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0
m\u001b[0m\n\u001b[1;32m   2050\u001b[0m
\u001b[0;32mif\u001b[0m
\u001b[0mself\u001b[0m\u001b[0;34m.\u001b[0m\u001b[0m_first_chunk\u001b[0
m\u001b[0;34m:\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n","\u001b[0;32mpan
das/_libs/parsers.pyx\u001b[0m in
\u001b[0;36mpandas._libs.parsers.TextReader.read\u001b[0;34m()\u001b[0m\n
","\u001b[0;32mpandas/_libs/parsers.pyx\u001b[0m in
\u001b[0;36mpandas._libs.parsers.TextReader._read_low_memory\u001b[0;34m(
)\u001b[0m\n","\u001b[0;32mpandas/_libs/parsers.pyx\u001b[0m in
\u001b[0;36mpandas._libs.parsers.TextReader._read_rows\u001b[0;34m()\u001
b[0m\n","\u001b[0;32mpandas/_libs/parsers.pyx\u001b[0m in
\u001b[0;36mpandas._libs.parsers.TextReader._tokenize_rows\u001b[0;34m()\
u001b[0m\n","\u001b[0;32mpandas/_libs/parsers.pyx\u001b[0m in
\u001b[0;36mpandas._libs.parsers.raise_parser_error\u001b[0;34m()\u001b[0
m\n","\u001b[0;31mParserError\u001b[0m: Error tokenizing data. C error:
Calling read(nbytes) on source failed. Try
engine='python'."],"ename":"ParserError","evalue":"Error tokenizing data.
C error: Calling read(nbytes) on source failed. Try
engine='python'.","output_type":"error"}]},{"cell_type":"code","source":"
train_df.info()","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.454220Z","iopub.status.idle":"2023-03-
28T02:52:45.454757Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
product_price","metadata":{}},{"cell_type":"code","source":"sale_price_sr
```

```
= (train_df.Venta_hoy / train_df.Venta_uni_hoy)\nreturn_price_sr =
(train_df.Dev_proxima / train_df.Dev_uni_proxima)\nproduct_price_df =
pd.DataFrame({'Producto_ID': train_df.Producto_ID, 'sale_price':
sale_price_sr, 'return_price': return_price_sr})\n\ndel
sale_price_sr\ndel return_price_sr\n\nprint(f'product price\\'s shape:
{product_price_df.shape}')\nproduct_price_df.head(5)","metadata":{"execut
ion":{"iopub.status.busy":"2023-03-
28T02:52:45.456019Z","iopub.status.idle":"2023-03-
28T02:52:45.456653Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_price_df.info()","metadata":{"exe
cution":{"iopub.status.busy":"2023-03-
28T02:52:45.457906Z","iopub.status.idle":"2023-03-
28T02:52:45.458466Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 1.3 Defining
Evaluation","metadata":{}},{"cell_type":"code","source":"'''KFold for
cross validation'''\nkf = KFold(n_splits=3, shuffle=True,
random_state=2)\n\n'''Define the validation function'''\ndef
rmsle_cv(model, X, y, cv=kf):\n    rmsle = np.sqrt(\n        -
cross_val_score(\n            model,\n            X, y,\n
scoring=\"neg_mean_squared_log_error\",\n            cv=cv,\n        )\n
)\n    return(rmsle)","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.459565Z","iopub.status.idle":"2023-03-
28T02:52:45.460038Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"# 2. Adjusting
Data","metadata":{}},{"cell_type":"markdown","source":"## 2.1
Cleansing","metadata":{}},{"cell_type":"markdown","source":"###
town_state","metadata":{}},{"cell_type":"code","source":"town_state_df['T
own'] = town_state_df['Town'].str.upper()\ntown_state_df['Town_name'] =
town_state_df['Town_name'].str.upper()\ntown_state_df['State'] =
town_state_df['State'].str.upper()","metadata":{"execution":{"iopub.statu
s.busy":"2023-03-28T02:52:45.461674Z","iopub.status.idle":"2023-03-
28T02:52:45.462275Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df.groupby(['Town_name'])['Tow
n_ID'].nunique().sort_values(ascending=False)[:6]","metadata":{"execution
":{"iopub.status.busy":"2023-03-
28T02:52:45.463325Z","iopub.status.idle":"2023-03-
28T02:52:45.464045Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df[(town_state_df['Town_name']
.isin(['LOS MOCHIS',
'PINOTEPA']))].sort_values(by='Town_name')","metadata":{"execution":{"iop
ub.status.busy":"2023-03-28T02:52:45.465099Z","iopub.status.idle":"2023-
03-
28T02:52:45.465629Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df.loc[498, 'Town_ID'] =
2561","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.466953Z","iopub.status.idle":"2023-03-
28T02:52:45.467529Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df.head(5)","metadata":{"execu
tion":{"iopub.status.busy":"2023-03-
28T02:52:45.468757Z","iopub.status.idle":"2023-03-
28T02:52:45.469212Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df.groupby('Town_ID')['Town'].
nunique().sort_values(ascending=False)","metadata":{"execution":{"iopub.s
tatus.busy":"2023-03-28T02:52:45.470314Z","iopub.status.idle":"2023-03-
28T02:52:45.470820Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df[town_state_df.Town_ID.isin(
[2561, 2169,
2152])].sort_values(by='Town_ID')","metadata":{"execution":{"iopub.status
.busy":"2023-03-28T02:52:45.472131Z","iopub.status.idle":"2023-03-
28T02:52:45.472655Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df['Town_ID'].max()","metadata
":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.473885Z","iopub.status.idle":"2023-03-
28T02:52:45.474335Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df.loc[199, 'Town_ID'] =
3217\ntown_state_df.loc[311, 'Town_ID'] =
3218","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.475487Z","iopub.status.idle":"2023-03-
28T02:52:45.476009Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df['Town_ID'].nunique()","meta
data":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.477072Z","iopub.status.idle":"2023-03-
28T02:52:45.477585Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
product_price","metadata":{}},{"cell_type":"code","source":"sale_prices_d
f = product_price_df.drop('return_price',
axis=1).dropna().rename(columns={'sale_price':
'price'})\nreturn_prices_df = product_price_df.drop('sale_price',
axis=1).dropna().rename(columns={'return_price': 'price'})\nprices_df =
pd.concat([sale_prices_df, return_prices_df])\nprices_df =
prices_df.groupby('Producto_ID')['price'].median().reset_index()\nprices_
df.head(5)","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.478942Z","iopub.status.idle":"2023-03-
28T02:52:45.479475Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"prices_df.shape","metadata":{"execution":
{"iopub.status.busy":"2023-03-
28T02:52:45.480540Z","iopub.status.idle":"2023-03-
28T02:52:45.481048Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
product","metadata":{}},{"cell_type":"code","source":"product_df['in_trai
n'] = 0\nproduct_df['in_test'] =
0\nproduct_df.loc[product_df['Producto_ID'].isin(test_df['Producto_ID'].u
nique()), 'in_test'] =
1\nproduct_df.loc[product_df['Producto_ID'].isin(train_df['Producto_ID'].
unique()), 'in_train'] = 1\nproduct_df =
product_df[(product_df['in_test'] == 1) | (product_df['in_train'] ==
1)]","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.482162Z","iopub.status.idle":"2023-03-
28T02:52:45.482759Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df[product_df['property'].isnull(
)]","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.483734Z","iopub.status.idle":"2023-03-
28T02:52:45.484185Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df.loc[117, 'popular_name'] =
'Donas'\nproduct_df.loc[117, 'property'] = 'Prom
BIM'\nproduct_df.loc[117, 'unit'] = None\nproduct_df.loc[117, 'pieces'] =
'6p'\n\nproduct_df.loc[190, 'popular_name'] = 'Paletina para
Cafe'\nproduct_df.loc[190, 'property'] = 'NES'\nproduct_df.loc[190,
'unit'] = None\nproduct_df.loc[190, 'pieces'] =
None\n\nproduct_df.loc[381, 'popular_name'] = 'Camioncitos
Bimbo'\nproduct_df.loc[381, 'property'] = 'BIM'\nproduct_df.loc[381,
'unit'] = None\nproduct_df.loc[381, 'pieces'] =
None\n\nproduct_df.loc[1152, 'popular_name'] = 'Burrito Vaporero
FrijolChorizo'\nproduct_df.loc[1152, 'property'] = 'CU
LON'\nproduct_df.loc[1152, 'unit'] = '90g'\nproduct_df.loc[1152,
'pieces'] = None\n\nproduct_df.loc[1677, 'popular_name'] = 'Tarima Twin
```

```
Pack Thins Multig'\nproduct_df.loc[1677, 'property'] = 'CU
ORO'\nproduct_df.loc[1677, 'unit'] = None\nproduct_df.loc[1677, 'pieces']
= None\n\nproduct_df.loc[1888, 'popular_name'] = 'Deliciosas
Chochochispas'\nproduct_df.loc[1888, 'property'] = 'Prom MTA
LAR'\nproduct_df.loc[1888, 'unit'] = '204g'\nproduct_df.loc[1888,
'pieces'] = None\n\nproduct_df.loc[1889, 'popular_name'] = 'Deliciosas
Chochochispas'\nproduct_df.loc[1889, 'property'] = 'Prom
LAR'\nproduct_df.loc[1889, 'unit'] = '204g'\nproduct_df.loc[1889,
'pieces'] = None\n\nproduct_df.loc[2449, 'popular_name'] = 'Galleta
Granel Classics Chocolate'\nproduct_df.loc[2449, 'property'] =
'GBI'\nproduct_df.loc[2449, 'unit'] = None\nproduct_df.loc[2449,
'pieces'] = None","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.485419Z","iopub.status.idle":"2023-03-
28T02:52:45.485936Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df[product_df['popular_name'].isn
ull()]","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.487402Z","iopub.status.idle":"2023-03-
28T02:52:45.488093Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df.loc[877, 'popular_name'] =
'Tortilla Hna Chihuahua'\nproduct_df.loc[877, 'unit'] =
'535g'\nproduct_df.loc[877, 'pieces'] = '10p'\n\nproduct_df.loc[1585,
'popular_name'] = 'Principe Cho Bco MG'\nproduct_df.loc[1585, 'unit'] =
'110g'\nproduct_df.loc[1585, 'pieces'] = '10p'\n\nproduct_df.loc[1748,
'popular_name'] = 'Combo Salma mas Levite'\nproduct_df.loc[1748, 'unit']
= '1360g'\nproduct_df.loc[1748, 'pieces'] =
None","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.489377Z","iopub.status.idle":"2023-03-
28T02:52:45.489897Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df['pieces'] =
product_df['pieces'].str.extract(r'(\\d+)(p|Reb)')[0]","metadata":{"execu
tion":{"iopub.status.busy":"2023-03-
28T02:52:45.491136Z","iopub.status.idle":"2023-03-
28T02:52:45.491663Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df['weight'] =
product_df['unit'].str.strip()\nproduct_df['weight'] =
product_df['weight'].str.replace(' ', '.')\nproduct_df['weight'] =
product_df['weight'].str.upper()\nw =
product_df['weight'].str.extract('(.+?)(KG|G|ML)',
expand=True)\nproduct_df['weight'] = w[0].astype('float') *
w[1].map({'KG':1000, 'G':1,
'ML':1})","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.492616Z","iopub.status.idle":"2023-03-
28T02:52:45.493031Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df = pd.merge(product_df,
prices_df,
how='left')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.494000Z","iopub.status.idle":"2023-03-
28T02:52:45.494519Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.2 Imputing Missing
Data","metadata":{}},{"cell_type":"markdown","source":"###
product","metadata":{}},{"cell_type":"code","source":"product_df['pieces'
] =
product_df['pieces'].fillna(1)","metadata":{"execution":{"iopub.status.bu
sy":"2023-03-28T02:52:45.495552Z","iopub.status.idle":"2023-03-
28T02:52:45.495962Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df[product_df['weight'].isnull()]
['price'].max()","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.497034Z","iopub.status.idle":"2023-03-
28T02:52:45.497764Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"product_df[product_df['price'].isnull()][
'weight'].max()","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.498695Z","iopub.status.idle":"2023-03-
28T02:52:45.499133Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"df = product_df.dropna()\ndf =
df[(df['price'] <= 311) & (df['weight'] <=
1880)]\nplt.figure(figsize=(16,8))\nsns.scatterplot(x='weight',
y='price', data=df)\ndel
df","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.499978Z","iopub.status.idle":"2023-03-
28T02:52:45.500495Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"from sklearn.linear_model import
LinearRegression\n\ndf = product_df.dropna()\ndf = df[(df['price'] <=
100) & (df['weight'] <= 1880)]\n\n# predict missing prices\nlf =
LinearRegression()\nlf.fit(df['weight'].values.reshape(-1, 1),
df['price'])\n\nprices =
lf.predict(product_df[product_df['price'].isnull()]['weight'].values.resh
ape(-1, 1))\n\nproduct_df.loc[product_df['price'].isnull(), 'price'] =
prices\n\n# predict missing weights\nlf =
LinearRegression()\nlf.fit(df['price'].values.reshape(-1, 1),
df['weight'])\n\nweights =
lf.predict(product_df[product_df['weight'].isnull()]['price'].values.resh
ape(-1, 1))\n\nproduct_df.loc[product_df['weight'].isnull(), 'weight'] =
weights\n\ndel df\ndel prices\ndel
weights","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.501546Z","iopub.status.idle":"2023-03-
28T02:52:45.502199Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"(product_df.drop(['unit'],
axis=1).isnull().sum() ==
0).all()","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.503318Z","iopub.status.idle":"2023-03-
28T02:52:45.503814Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.3 Transforming Data
Type","metadata":{}},{"cell_type":"code","source":"product_df['pieces'] =
product_df['pieces'].astype('int16')\nproduct_df['in_train'] =
product_df['in_train'].astype('bool')\nproduct_df['in_test'] =
product_df['in_test'].astype('bool')\nproduct_df['weight'] =
product_df['weight'].astype('float32')\nproduct_df['price'] =
product_df['price'].astype('float32')","metadata":{"execution":{"iopub.st
atus.busy":"2023-03-28T02:52:45.504621Z","iopub.status.idle":"2023-03-
28T02:52:45.505027Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"town_state_df['State'] =
town_state_df['State'].astype('category')","metadata":{"execution":{"iopu
b.status.busy":"2023-03-28T02:52:45.505883Z","iopub.status.idle":"2023-
03-
28T02:52:45.506262Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df['Canal_ID'] =
train_df['Canal_ID'].astype('category')\ntest_df['Canal_ID'] =
test_df['Canal_ID'].astype('category')","metadata":{"execution":{"iopub.s
tatus.busy":"2023-03-28T02:52:45.507232Z","iopub.status.idle":"2023-03-
28T02:52:45.507768Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.4 Dropping
Features","metadata":{}},{"cell_type":"code","source":"train_df.drop(['Ve
nta_uni_hoy', 'Venta_hoy', 'Dev_uni_proxima', 'Dev_proxima'], axis=1,
inplace=True)","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.508739Z","iopub.status.idle":"2023-03-
28T02:52:45.509248Z"},"trusted":true},"execution_count":null,"outputs":[]
```

```
},{"cell_type":"markdown","source":"## 2.5 Merging
Data","metadata":{}},{"cell_type":"markdown","source":"###
town_state","metadata":{}},{"cell_type":"code","source":"train_df =
pd.merge(train_df, town_state_df[['Agencia_ID', 'Town_ID']],
how='left')\ntest_df = pd.merge(test_df, town_state_df[['Agencia_ID',
'Town_ID']], how='left')\ntrain_df.drop('Agencia_ID', axis=1,
inplace=True)\ntest_df.drop('Agencia_ID', axis=1,
inplace=True)","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.510560Z","iopub.status.idle":"2023-03-
28T02:52:45.511151Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
product","metadata":{}},{"cell_type":"code","source":"train_df =
pd.merge(\n    train_df,\n    product_df[[\n        'Producto_ID',
'popular_name', 'property',\n        'pieces', 'weight', 'price'\n    ]],
how='left')\ntest_df = pd.merge(\n    test_df,\n    product_df[[\n
'Producto_ID', 'popular_name', 'property',\n        'pieces', 'weight',
'price'\n    ]],
how='left')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.512234Z","iopub.status.idle":"2023-03-
28T02:52:45.512720Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df.head(5)","metadata":{"execution"
:{"iopub.status.busy":"2023-03-
28T02:52:45.513710Z","iopub.status.idle":"2023-03-
28T02:52:45.514230Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.6 Bin-
Counting","metadata":{}},{"cell_type":"markdown","source":"###
Semana","metadata":{}},{"cell_type":"code","source":"semana_med_s =
train_df.groupby('Semana')['Demanda_uni_equil'].median()","metadata":{"ex
ecution":{"iopub.status.busy":"2023-03-
28T02:52:45.515165Z","iopub.status.idle":"2023-03-
28T02:52:45.515781Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"semana_med_s","metadata":{"execution":{"i
opub.status.busy":"2023-03-
28T02:52:45.516774Z","iopub.status.idle":"2023-03-
28T02:52:45.517310Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df.drop('Semana', axis=1,
inplace=True)\ntest_df.drop('Semana', axis=1, inplace=True)\ndel
semana_med_s","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.518344Z","iopub.status.idle":"2023-03-
28T02:52:45.518809Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
Cliente_ID","metadata":{}},{"cell_type":"code","source":"client_med_s =
train_df.groupby('Cliente_ID')['Demanda_uni_equil'].median().astype('int1
6')\nclient_med_s.name =
'client_med'","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.519794Z","iopub.status.idle":"2023-03-
28T02:52:45.520337Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
popular_name","metadata":{}},{"cell_type":"code","source":"popular_name_m
ed_s =
train_df.groupby('popular_name')['Demanda_uni_equil'].median().astype('in
t16')\npopular_name_med_s.name =
'popular_name_med'","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.521250Z","iopub.status.idle":"2023-03-
28T02:52:45.521751Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
Town_ID","metadata":{}},{"cell_type":"code","source":"town_id_med_s =
train_df.groupby('Town_ID')['Demanda_uni_equil'].median().astype('int16')
```

\ntown_id_med_s.name =
'town_id_med'","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.522880Z","iopub.status.idle":"2023-03-
28T02:52:45.523423Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
Ruta_SAK","metadata":{}},{"cell_type":"code","source":"ruta_id_med_s =
train_df.groupby('Ruta_SAK')['Demanda_uni_equil'].median().astype('int16'
)\nruta_id_med_s.name =
'ruta_id_med'","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.524402Z","iopub.status.idle":"2023-03-
28T02:52:45.524825Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.7 Merging Bin-Counting
Data","metadata":{}},{"cell_type":"code","source":"test_df =
pd.merge(test_df, client_med_s.reset_index(), how='left')\ntest_df =
pd.merge(test_df, popular_name_med_s.reset_index(), how='left')\ntest_df
= pd.merge(test_df, town_id_med_s.reset_index(), how='left')\ntest_df =
pd.merge(test_df, ruta_id_med_s.reset_index(),
how='left')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.525700Z","iopub.status.idle":"2023-03-
28T02:52:45.526157Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df = pd.merge(train_df,
client_med_s.reset_index(), how='left')\ntrain_df = pd.merge(train_df,
popular_name_med_s.reset_index(), how='left')\ntrain_df =
pd.merge(train_df, town_id_med_s.reset_index(), how='left')\ntrain_df =
pd.merge(train_df, ruta_id_med_s.reset_index(),
how='left')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.526993Z","iopub.status.idle":"2023-03-
28T02:52:45.527505Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.8 Imputing Test Missing
Data","metadata":{}},{"cell_type":"code","source":"test_df['popular_name_
med'] =
test_df['popular_name_med'].fillna(test_df['popular_name_med'].mean())\nt
est_df['client_med'] =
test_df['client_med'].fillna(test_df['client_med'].mean())\ntest_df['ruta
_id_med'] =
test_df['ruta_id_med'].fillna(test_df['ruta_id_med'].mean())","metadata":
{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.528362Z","iopub.status.idle":"2023-03-
28T02:52:45.528840Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.9 Transforming Data
Type","metadata":{}},{"cell_type":"code","source":"train_df['client_med']
= train_df['client_med'].astype('int16')\ntrain_df['popular_name_med'] =
train_df['popular_name_med'].astype('int16')\ntrain_df['town_id_med'] =
train_df['town_id_med'].astype('int16')\ntrain_df['ruta_id_med'] =
train_df['ruta_id_med'].astype('int16')","metadata":{"execution":{"iopub.
status.busy":"2023-03-28T02:52:45.530086Z","iopub.status.idle":"2023-03-
28T02:52:45.530471Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"test_df['client_med'] =
test_df['client_med'].astype('int16')\ntest_df['popular_name_med'] =
test_df['popular_name_med'].astype('int16')\ntest_df['town_id_med'] =
test_df['town_id_med'].astype('int16')\ntest_df['ruta_id_med'] =
test_df['ruta_id_med'].astype('int16')","metadata":{"execution":{"iopub.s
tatus.busy":"2023-03-28T02:52:45.531482Z","iopub.status.idle":"2023-03-
28T02:52:45.531916Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 2.10 Dropping
Features","metadata":{}},{"cell_type":"code","source":"train_df.drop(\n
['Ruta_SAK', 'Cliente_ID', 'Producto_ID', 'Town_ID', 'popular_name',
'property', 'pieces'],\n   axis=1, inplace=True)\ntest_df.drop(\n

```
['Ruta_SAK', 'Cliente_ID', 'Producto_ID', 'Town_ID', 'popular_name',
'property', 'pieces'],\n    axis=1,
inplace=True)","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.532840Z","iopub.status.idle":"2023-03-
28T02:52:45.533399Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df.head(3)","metadata":{"execution"
:{"iopub.status.busy":"2023-03-
28T02:52:45.534710Z","iopub.status.idle":"2023-03-
28T02:52:45.535295Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"test_df.head(3)","metadata":{"execution":
{"iopub.status.busy":"2023-03-
28T02:52:45.536227Z","iopub.status.idle":"2023-03-
28T02:52:45.536848Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"# 3. Data
Preprocessing","metadata":{}},{"cell_type":"markdown","source":"## 3.1
Take a glance at all
variables","metadata":{}},{"cell_type":"code","source":"train_df.info()",
"metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.538205Z","iopub.status.idle":"2023-03-
28T02:52:45.538659Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"test_df.info()","metadata":{"execution":{
"iopub.status.busy":"2023-03-
28T02:52:45.539620Z","iopub.status.idle":"2023-03-
28T02:52:45.540029Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df.describe()","metadata":{"executi
on":{"iopub.status.busy":"2023-03-
28T02:52:45.544361Z","iopub.status.idle":"2023-03-
28T02:52:45.545016Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"test_df.describe()","metadata":{"executio
n":{"iopub.status.busy":"2023-03-
28T02:52:45.546213Z","iopub.status.idle":"2023-03-
28T02:52:45.546859Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''Plot histogram of numerical variables
to validate pandas intuition.'''\ndef draw_histograms(df, variables,
n_rows, n_cols, size):\n    fig=plt.figure()\n    for i, var_name in
enumerate(variables):\n        ax=fig.add_subplot(n_rows, n_cols, i+1)\n
df[var_name].hist(bins=40, ax=ax, color='skyblue', alpha=0.8,
figsize=size)\n        ax.set_title(var_name, fontsize=43)\n
ax.tick_params(axis='both', which='major', labelsize=35)\n
ax.tick_params(axis='both', which='minor', labelsize=35)\n
ax.set_xlabel('')\n    fig.tight_layout(rect=[0, 0.03, 1, 0.95])\n
plt.show()","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.548433Z","iopub.status.idle":"2023-03-
28T02:52:45.548967Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"skewness =
train_df.select_dtypes(include=['int8', 'int16', 'int32', 'int64',
'float32', 'float64']).apply(lambda x: skew(x))\nskew_index =
skewness[abs(skewness) >=
0.75].index\nskewness[skew_index].sort_values(ascending=False)","metadata
":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.550419Z","iopub.status.idle":"2023-03-
28T02:52:45.550952Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"skewness =
test_df.select_dtypes(include=['int8', 'int16', 'int32', 'int64',
'float32', 'float64']).apply(lambda x: skew(x))\nskew_index =
skewness[abs(skewness) >=
0.75].index\nskewness[skew_index].sort_values(ascending=False)","metadata
":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.552987Z","iopub.status.idle":"2023-03-
28T02:52:45.553661Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 3.2 BoxCox
Transform","metadata":{}},{"cell_type":"code","source":"'''BoxCox
Transform'''\nlam = 0.01\nfor column in skew_index:\n    train_df[column]
= boxcox1p(train_df[column], lam)\n    test_df[column] =
boxcox1p(test_df[column],
lam)","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.555188Z","iopub.status.idle":"2023-03-
28T02:52:45.555742Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"train_df.info()","metadata":{"execution":
{"iopub.status.busy":"2023-03-
28T02:52:45.557011Z","iopub.status.idle":"2023-03-
28T02:52:45.557517Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"test_df.info()","metadata":{"execution":{
"iopub.status.busy":"2023-03-
28T02:52:45.558906Z","iopub.status.idle":"2023-03-
28T02:52:45.559414Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 3.3 Transforming Data
Type","metadata":{}},{"cell_type":"code","source":"train_df['client_med']
= train_df['client_med'].astype('float32')\ntrain_df['popular_name_med']
= train_df['popular_name_med'].astype('float32')\ntrain_df['town_id_med']
= train_df['town_id_med'].astype('float32')\ntrain_df['ruta_id_med'] =
train_df['ruta_id_med'].astype('float32')","metadata":{"execution":{"iopu
b.status.busy":"2023-03-28T02:52:45.560694Z","iopub.status.idle":"2023-
03-
28T02:52:45.561283Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"test_df['client_med'] =
test_df['client_med'].astype('float32')\ntest_df['popular_name_med'] =
test_df['popular_name_med'].astype('float32')\ntest_df['town_id_med'] =
test_df['town_id_med'].astype('float32')\ntest_df['ruta_id_med'] =
test_df['ruta_id_med'].astype('float32')","metadata":{"execution":{"iopub
.status.busy":"2023-03-28T02:52:45.562404Z","iopub.status.idle":"2023-03-
28T02:52:45.563054Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"# 4. Exploratory Data Analysis
","metadata":{}},{"cell_type":"markdown","source":"## 4.1 Analyzing
Target ","metadata":{}},{"cell_type":"code","source":"sample_train_df =
train_df.sample(n=10000)\nsample_train_df['log_target'] =
np.log1p(sample_train_df['Demanda_uni_equil'])","metadata":{"execution":{
"iopub.status.busy":"2023-03-
28T02:52:45.564076Z","iopub.status.idle":"2023-03-
28T02:52:45.564804Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"sample_train_df.head(5)","metadata":{"exe
cution":{"iopub.status.busy":"2023-03-
28T02:52:45.565917Z","iopub.status.idle":"2023-03-
28T02:52:45.566457Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''correlation
matrix'''\nplt.subplots(figsize=(20, 16))\nk = 20 #number of variables
for heatmap\ncorrmat = sample_train_df.corr()\ncols = corrmat.nlargest(k,
'log_target')['log_target'].index\n\ncm =
np.corrcoef(sample_train_df[cols].values.T)\nhm = sns.heatmap(cm,
cbar=True, annot=True, square=True,\n                 fmt='.2f',
annot_kws={'size': 10}, cmap='Blues',\n
yticklabels=cols.values,
xticklabels=cols.values)\nplt.show()","metadata":{"execution":{"iopub.sta
tus.busy":"2023-03-28T02:52:45.567786Z","iopub.status.idle":"2023-03-
28T02:52:45.568381Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''Check feature inportance by applying
```

LightGBM'''\nimport lightgbm as lgb\nmodel_lgb =
lgb.LGBMRegressor(num_leaves=1000,\n
max_depth=5,\n   learning_rate=0.1,\n
random_state=2)\nmodel_lgb.fit(sample_train_df.drop(['Demanda_uni_equil',
'log_target'], axis=1),
sample_train_df['log_target'])","metadata":{"execution":{"iopub.status.bu
sy":"2023-03-28T02:52:45.569552Z","iopub.status.idle":"2023-03-
28T02:52:45.570069Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"df =
pd.DataFrame(model_lgb.feature_importances_,\n
index=sample_train_df.drop(['Demanda_uni_equil', 'log_target'],
axis=1).columns,\n
columns=['importance']).sort_values('importance',
ascending=False)\ndf[df.importance >
10]","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.571294Z","iopub.status.idle":"2023-03-
28T02:52:45.571813Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"# weight\nplt.figure(figsize=(16,
8))\nsns.scatterplot(x='weight', y='log_target', data=sample_train_df,
palette='Blues_d')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.572974Z","iopub.status.idle":"2023-03-
28T02:52:45.573622Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"# price\nplt.figure(figsize=(16,
8))\nsns.scatterplot(x='price', y='log_target', data=sample_train_df,
palette='Blues_d')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.574843Z","iopub.status.idle":"2023-03-
28T02:52:45.575371Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"# client_med\nplt.figure(figsize=(16,
8))\nsns.scatterplot(x='client_med', y='log_target',
data=sample_train_df,
palette='Blues_d')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.576705Z","iopub.status.idle":"2023-03-
28T02:52:45.577171Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"#
popular_name_med\nplt.figure(figsize=(16,
8))\nsns.scatterplot(x='popular_name_med', y='log_target',
data=sample_train_df,
palette='Blues_d')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.578518Z","iopub.status.idle":"2023-03-
28T02:52:45.579131Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"# town_id_med\nplt.figure(figsize=(16,
8))\nsns.scatterplot(x='town_id_med', y='log_target',
data=sample_train_df,
palette='Blues_d')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.580247Z","iopub.status.idle":"2023-03-
28T02:52:45.580903Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"# ruta_id_med\nplt.figure(figsize=(16,
8))\nsns.scatterplot(x='ruta_id_med', y='log_target',
data=sample_train_df,
palette='Blues_d')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.582056Z","iopub.status.idle":"2023-03-
28T02:52:45.582532Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"# 5. Model Building and
Evaluation","metadata":{}},{"cell_type":"markdown","source":"## 5.1
Importing
Packages","metadata":{}},{"cell_type":"code","source":"'''Importing
Modeling Interested Modules'''\nfrom sklearn.base import
BaseEstimator\nfrom sklearn.pipeline import make_pipeline\nfrom

```
sklearn.preprocessing import StandardScaler, RobustScaler\nfrom
sklearn.model_selection import GridSearchCV\nfrom sklearn.linear_model
import LinearRegression, LassoCV, RidgeCV, ElasticNetCV\nfrom sklearn.svm
import SVR\nfrom sklearn.kernel_ridge import KernelRidge\nfrom lightgbm
import LGBMRegressor","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.583740Z","iopub.status.idle":"2023-03-
28T02:52:45.584169Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 5.2 Preparation before Building
Models ","metadata":{}},{"cell_type":"code","source":"'''Adjust dataframe
for modeling'''\ntrain_y =
train_df['Demanda_uni_equil']\ntrain_df.drop(['Demanda_uni_equil'],
axis=1, inplace=True)\ntrain_X = train_df\ntest_X = test_df.drop('id',
axis=1)\n\n'''Transform categorical features to dummy
variables'''\ntrain_X = pd.get_dummies(train_X)\ntest_X =
pd.get_dummies(test_X)","metadata":{"execution":{"iopub.status.busy":"202
3-03-28T02:52:45.585285Z","iopub.status.idle":"2023-03-
28T02:52:45.585918Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"sample_train_df['Demanda_uni_equil'] =
np.expm1(sample_train_df['log_target']).astype('int32')","metadata":{"exe
cution":{"iopub.status.busy":"2023-03-
28T02:52:45.587519Z","iopub.status.idle":"2023-03-
28T02:52:45.588101Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"sample_train_df.head(5)","metadata":{"exe
cution":{"iopub.status.busy":"2023-03-
28T02:52:45.589598Z","iopub.status.idle":"2023-03-
28T02:52:45.590147Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''Prepare sample train for the fast
training'''\nsample_train_y =
sample_train_df['Demanda_uni_equil']\nsample_train_df.drop(['Demanda_uni_
equil', 'log_target'], axis=1, inplace=True)\nsample_train_X =
sample_train_df\n\nsample_train_X =
pd.get_dummies(sample_train_X)","metadata":{"execution":{"iopub.status.bu
sy":"2023-03-28T02:52:45.591466Z","iopub.status.idle":"2023-03-
28T02:52:45.592023Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''We should use the log transform of the
target value'''\nclass MyEstimator(BaseEstimator):\n    def
__init__(self, model):\n        self.model = model\n        \n    def
fit(self, X, y):\n        self.model.fit(X, np.log1p(y))\n        return
self \n\n    def predict(self, X):\n        predicts =
np.expm1(self.model.predict(X))\n        mask = (predicts <= 0)\n
predicts[mask] = 0\n        return
predicts","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.593378Z","iopub.status.idle":"2023-03-
28T02:52:45.594294Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''Define evaluation function for
Convienience'''\ndef evaluation_model(model, train_X, train_y, test_X):\n
cv = rmsle_cv(model, train_X, train_y)\n    cv_mean = np.round(cv.mean(),
5)\n    cv_std = np.round(cv.std(), 5)\n    sample_prediction =
model.predict(test_X.loc[:3, :])\n    return {'cv_mean': cv_mean,
'cv_std': cv_std, 'sample_prediction':
sample_prediction}","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.595536Z","iopub.status.idle":"2023-03-
28T02:52:45.596087Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''Define Hyperparameters Tuning
Function'''\ndef tune_hyperparameters(model, param_grid, train_X,
train_y):\n    grid = GridSearchCV(\n        model, param_grid, \n
scoring='neg_mean_squared_log_error',\n        cv=3, n_jobs=-1,\n    )\n
grid.fit(train_X, train_y)\n    best_params = grid.best_params_ \n
```

```
best_score = np.round(np.sqrt(-1 * grid.best_score_), 5)\n    return
best_params,
best_score","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.597310Z","iopub.status.idle":"2023-03-
28T02:52:45.598205Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 5.3 Building
Models","metadata":{}},{"cell_type":"markdown","source":"###
LinearRegression","metadata":{}},{"cell_type":"code","source":"model =
make_pipeline(\n    RobustScaler(),\n    LinearRegression(),\n)\nlr_model
= MyEstimator(model)\nlr_model.fit(sample_train_X,
sample_train_y)\nlr_eval = evaluation_model(lr_model, sample_train_X,
sample_train_y,
test_X)\nprint(lr_eval)","metadata":{"execution":{"iopub.status.busy":"20
23-03-28T02:52:45.599498Z","iopub.status.idle":"2023-03-
28T02:52:45.600172Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
LassoCV","metadata":{}},{"cell_type":"code","source":"model =
make_pipeline(\n    RobustScaler(),\n    LassoCV(\n
alphas=(0.00001, 0.0001, 0.0005, 0.001, 0.01, 0.05, 0.1, 0.3, 1, 3, 5,
10),\n    ),\n)\nlasso_cv_model =
MyEstimator(model)\nlasso_cv_model.fit(sample_train_X,
sample_train_y)\nlasso_cv_eval = evaluation_model(lasso_cv_model,
sample_train_X, sample_train_y,
test_X)\nprint(lasso_cv_eval)\n\nopt_alpha =
lasso_cv_model.model.steps[1][1].alpha_\nprint(f'\\nopt_alpha:
{opt_alpha}')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.601564Z","iopub.status.idle":"2023-03-
28T02:52:45.602189Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
RidgeCV","metadata":{}},{"cell_type":"code","source":"model =
make_pipeline(\n    RobustScaler(),\n    RidgeCV(\n
alphas=(0.0001, 0.0005, 0.001, 0.01, 0.05, 0.1, 0.3, 1, 3, 5, 10),\n
),\n)\nridge_cv_model =
MyEstimator(model)\nridge_cv_model.fit(sample_train_X,
sample_train_y)\nridge_cv_eval = evaluation_model(ridge_cv_model,
sample_train_X, sample_train_y,
test_X)\nprint(ridge_cv_eval)\n\nopt_alpha =
ridge_cv_model.model.steps[1][1].alpha_\nprint(f'\\nopt_alpha:
{opt_alpha}')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.603543Z","iopub.status.idle":"2023-03-
28T02:52:45.603904Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
ElasticNetCV","metadata":{}},{"cell_type":"code","source":"model =
make_pipeline(\n    RobustScaler(),\n    ElasticNetCV(\n
alphas=(0.00001, 0.0001, 0.0002, 0.0003), \n        l1_ratio=(0.001,
0.01, 0.1, 0.2, 0.3, 0.4, 0.5),\n    ),\n)\nelsnt_cv_model =
MyEstimator(model)\nelsnt_cv_model.fit(sample_train_X,
sample_train_y)\nelsnt_cv_eval = evaluation_model(elsnt_cv_model,
sample_train_X, sample_train_y,
test_X)\nprint(elsnt_cv_eval)\n\nopt_alpha =
elsnt_cv_model.model.steps[1][1].alpha_\nopt_l1_ratio =
elsnt_cv_model.model.steps[1][1].l1_ratio_\nprint(f'\\nopt_alpha:
{opt_alpha} opt_l1_ratio:
{opt_l1_ratio}')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.604694Z","iopub.status.idle":"2023-03-
28T02:52:45.605018Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
SVR","metadata":{}},{"cell_type":"code","source":"# {'cv_mean': 0.57025,
'cv_std': 0.00775, 'sample_prediction': array([3.85650165, 1.24517075,
3.45866592])}\n\n# grid best_params: {'model__svr__C': 10,
'model__svr__epsilon': 0.1, 'model__svr__gamma': 0.01}\n\n# ### build
basemodel\n# model = make_pipeline(\n#     RobustScaler(),\n#
SVR(),\n# )\n# svr_model = MyEstimator(model)\n# \n# ### optimize
hyperparameters\n# param_grid = {'model__svr__C': [1, 10, 20],\n#
'model__svr__epsilon': [0.001, 0.01, 0.1],\n#
'model__svr__gamma': [0.0001, 0.001, 0.01]}\n# best_params, best_score =
\\\n#     tune_hyperparameters(svr_model, param_grid, sample_train_X,
sample_train_y)\n# \n# ### fit using best_params\n#
svr_model.set_params(**best_params)\n# svr_model.fit(sample_train_X,
sample_train_y)\n# svr_eval = evaluation_model(svr_model, sample_train_X,
sample_train_y, test_X)\n# print(svr_eval)\n# \n# print(f'\\ngrid
best_params:
{best_params}')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.605824Z","iopub.status.idle":"2023-03-
28T02:52:45.606156Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
KernelRidge","metadata":{}},{"cell_type":"code","source":"# {'cv_mean':
0.56925, 'cv_std': 0.00546, 'sample_prediction': array([4.15061605,
1.21544493, 3.65678685, 1.81862787])}\n\n# grid best_params:
{'model__kernelridge__alpha': 0.5, 'model__kernelridge__coef0': 3,
'model__kernelridge__degree': 2, 'model__kernelridge__kernel':
'polynomial'}\n\n# ### build basemodel\n# model = make_pipeline(\n#
RobustScaler(),\n#     KernelRidge(),\n# )\n# kr_model =
MyEstimator(model)\n# \n# ### optimize hyperparameters\n# param_grid =
{'model__kernelridge__alpha': [0.01, 0.1, 0.5, 1],\n#
'model__kernelridge__kernel': ['linear', 'polynomial'],\n#
'model__kernelridge__degree': [1, 1.5, 2, 3],\n#
'model__kernelridge__coef0': [3, 4, 5]}\n# best_params, best_score =
\\\n#     tune_hyperparameters(kr_model, param_grid, sample_train_X,
sample_train_y)\n# \n# ### fit using best_params\n#
kr_model.set_params(**best_params)\n# kr_model.fit(sample_train_X,
sample_train_y)\n# kr_eval = evaluation_model(kr_model, sample_train_X,
sample_train_y, test_X)\n# print(kr_eval)\n# \n# print(f'\\ngrid
best_params:
{best_params}')","metadata":{"execution":{"iopub.status.busy":"2023-03-
28T02:52:45.607096Z","iopub.status.idle":"2023-03-
28T02:52:45.607614Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"###
LightGBM","metadata":{}},{"cell_type":"code","source":"model =
LGBMRegressor(learning_rate=0.01, n_estimators=3000,\n
num_leaves=5,\n                      objective='regression',\n
max_bin=55, bagging_fraction=0.8,\n                      bagging_freq=5,
feature_fraction=0.2319,\n                      feature_fraction_seed=9,
bagging_seed=9,\n                      min_data_in_leaf=6,
min_sum_hessian_in_leaf=11)\nlgb_model =
MyEstimator(model)\nlgb_model.fit(sample_train_X,
sample_train_y)\nlgb_eval = evaluation_model(lgb_model, sample_train_X,
sample_train_y,
test_X)\nprint(lgb_eval)","metadata":{"execution":{"iopub.status.busy":"2
023-03-28T02:52:45.608742Z","iopub.status.idle":"2023-03-
28T02:52:45.609432Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"markdown","source":"## 5.4
Submission","metadata":{}},{"cell_type":"code","source":"def
output_submission_file(model, test_X, filename='submission.csv'):\n
prediction = model.predict(test_X)\n    df = pd.DataFrame({'id':
test_df['id'], 'Demanda_uni_equil': prediction})\n
```

```
print(f'{df.shape}')\n    print(f'{df.head(5)}')\n    df.to_csv(filename,
index=False)\n    df.to_csv(filename + '.gz', index=False,
compression='gzip')","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.610677Z","iopub.status.idle":"2023-03-
28T02:52:45.611166Z"},"trusted":true},"execution_count":null,"outputs":[]
},{"cell_type":"code","source":"'''Submission'''\noutput_submission_file(
lgb_model, test_X)","metadata":{"execution":{"iopub.status.busy":"2023-
03-28T02:52:45.612678Z","iopub.status.idle":"2023-03-
28T02:52:45.613290Z"},"trusted":true},"execution_count":null,"outputs":[]
}]}
```