lab01

July 25, 2022

José Javier Hurtarte #19707

Diana Zaray Corado #191025

1 Análisis Exploratorio, PCA y Apriori

[ ]:
```
# Librerías a utilizar
import pandas as pd
from pandas_profiling import ProfileReport
import matplotlib.pyplot as plt
import seaborn as sns
from factor_analyzer import FactorAnalyzer
import numpy as np
from apyori import apriori
```

```
c:\Users\josej\AppData\Local\Programs\Python\Python39\lib\site-
packages\tqdm\auto.py:22: TqdmWarning: IProgress not found. Please update
jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

[ ]:
```
def calculate_frecuency(data, column, index='index'):
    data_f = pd.DataFrame({
        'frecuency': data[column].value_counts(),
        'relative_frecuency (%)': data[column].value_counts(normalize=True)*100,
        'relative_acc_frecuency': data[column].value_counts(normalize=True).
 ↪cumsum()
    })
    data_f.reset_index(level=[0], inplace=True)
    data_f.rename(columns={index:column}, inplace=True)
    left_aligned_df = data_f.style.set_properties(**{'text-align': 'center'})
    display(left_aligned_df)
    return data_f
```

2 Haga una exploración rápida de sus datos para eso haga un resumen de su dataset.

Para el análisis de los datos y puesta en práctica de los conocimientos obtenidos mediante la clases teóricas, se trabajará con un set de datos proporcionado por Kaggle denominado House Prices: Advance Regression Techniques el cual cuenta con 1460 observaciones y 80 variables, en las cuales se describen diversas características de las casas así como su precio de venta.

[ ]:
```
data = pd.read_csv('train.csv').drop(['Id'], axis = 1)
print(f'El formato de los datos es: {data.shape}')
data
```

```
El formato de los datos es: (1460, 80)
```

[ ]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
0	60	RL	65.0	8450	Pave	NaN	Reg	
1	20	RL	80.0	9600	Pave	NaN	Reg	
2	60	RL	68.0	11250	Pave	NaN	IR1	
3	70	RL	60.0	9550	Pave	NaN	IR1	
4	60	RL	84.0	14260	Pave	NaN	IR1	
…	…	…	…	…	…	…	…	
1455	60	RL	62.0	7917	Pave	NaN	Reg	
1456	20	RL	85.0	13175	Pave	NaN	Reg	
1457	70	RL	66.0	9042	Pave	NaN	Reg	
1458	20	RL	68.0	9717	Pave	NaN	Reg	
1459	20	RL	75.0	9937	Pave	NaN	Reg	

	LandContour	Utilities	LotConfig	…	PoolArea	PoolQC	Fence	MiscFeature	\
0	Lvl	AllPub	Inside	…	0	NaN	NaN	NaN	
1	Lvl	AllPub	FR2	…	0	NaN	NaN	NaN	
2	Lvl	AllPub	Inside	…	0	NaN	NaN	NaN	
3	Lvl	AllPub	Corner	…	0	NaN	NaN	NaN	
4	Lvl	AllPub	FR2	…	0	NaN	NaN	NaN	
…	…	…	…	…	…	…	…	…	
1455	Lvl	AllPub	Inside	…	0	NaN	NaN	NaN	
1456	Lvl	AllPub	Inside	…	0	NaN	MnPrv	NaN	
1457	Lvl	AllPub	Inside	…	0	NaN	GdPrv	Shed	
1458	Lvl	AllPub	Inside	…	0	NaN	NaN	NaN	
1459	Lvl	AllPub	Inside	…	0	NaN	NaN	NaN	

	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	0	2	2008	WD	Normal	208500
1	0	5	2007	WD	Normal	181500
2	0	9	2008	WD	Normal	223500
3	0	2	2006	WD	Abnorml	140000
4	0	12	2008	WD	Normal	250000
…	…	…	…	…	…	
1455	0	8	2007	WD	Normal	175000

```
1456       0      2    2010        WD         Normal     210000
1457    2500      5    2010        WD         Normal     266500
1458       0      4    2010        WD         Normal     142125
1459       0      6    2008        WD         Normal     147500

[1460 rows x 80 columns]
```

[ ]:
```
data.describe()
```

[ ]:
```
        MSSubClass  LotFrontage        LotArea  OverallQual  OverallCond  \
count  1460.000000  1201.000000    1460.000000  1460.000000  1460.000000
mean     56.897260    70.049958   10516.828082     6.099315     5.575342
std      42.300571    24.284752    9981.264932     1.382997     1.112799
min      20.000000    21.000000    1300.000000     1.000000     1.000000
25%      20.000000    59.000000    7553.500000     5.000000     5.000000
50%      50.000000    69.000000    9478.500000     6.000000     5.000000
75%      70.000000    80.000000   11601.500000     7.000000     6.000000
max     190.000000   313.000000  215245.000000    10.000000     9.000000

         YearBuilt  YearRemodAdd   MasVnrArea   BsmtFinSF1   BsmtFinSF2  …  \
count  1460.000000   1460.000000  1452.000000  1460.000000  1460.000000  …
mean   1971.267808   1984.865753   103.685262   443.639726    46.549315  …
std      30.202904     20.645407   181.066207   456.098091   161.319273  …
min    1872.000000   1950.000000     0.000000     0.000000     0.000000  …
25%    1954.000000   1967.000000     0.000000     0.000000     0.000000  …
50%    1973.000000   1994.000000     0.000000   383.500000     0.000000  …
75%    2000.000000   2004.000000   166.000000   712.250000     0.000000  …
max    2010.000000   2010.000000  1600.000000  5644.000000  1474.000000  …

        WoodDeckSF  OpenPorchSF  EnclosedPorch    3SsnPorch  ScreenPorch  \
count  1460.000000  1460.000000    1460.000000  1460.000000  1460.000000
mean     94.244521    46.660274      21.954110     3.409589    15.060959
std     125.338794    66.256028      61.119149    29.317331    55.757415
min       0.000000     0.000000       0.000000     0.000000     0.000000
25%       0.000000     0.000000       0.000000     0.000000     0.000000
50%       0.000000    25.000000       0.000000     0.000000     0.000000
75%     168.000000    68.000000       0.000000     0.000000     0.000000
max     857.000000   547.000000     552.000000   508.000000   480.000000

          PoolArea       MiscVal       MoSold       YrSold      SalePrice
count  1460.000000   1460.000000  1460.000000  1460.000000    1460.000000
mean      2.758904     43.489041     6.321918  2007.815753  180921.195890
std      40.177307    496.123024     2.703626     1.328095   79442.502883
min       0.000000      0.000000     1.000000  2006.000000   34900.000000
25%       0.000000      0.000000     5.000000  2007.000000  129975.000000
50%       0.000000      0.000000     6.000000  2008.000000  163000.000000
75%       0.000000      0.000000     8.000000  2009.000000  214000.000000
```

```
max     738.000000  15500.000000    12.000000  2010.000000  755000.000000

[8 rows x 37 columns]
```

[ ]:
```
profile = ProfileReport(data)
profile
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
File ~\AppData\Roaming\Python\Python39\site-packages\IPython\core\formatters.py
 ↪343, in BaseFormatter.__call__(self, obj)
    341     method = get_real_method(obj, self.print_method)
    342     if method is not None:
--> 343         return method()
    344     return None
    345 else:

File c:
 ↪\Users\josej\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas_ rofiling\pro
 ↪py:418, in ProfileReport._repr_html_(self)
    416 def _repr_html_(self) -> None:
    417     """The ipython notebook widgets user interface gets called by the␣
 ↪jupyter notebook."""
--> 418     self.to_notebook_iframe()

File c:
 ↪\Users\josej\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas_ rofiling\pro
 ↪py:391, in ProfileReport.to_notebook_iframe(self)
    380 """Used to output the HTML representation to a Jupyter notebook.
    381 When config.notebook.iframe.attribute is "src", this function creates a
 ↪temporary HTML file
    382 in `./tmp/profile_[hash].html` and returns an Iframe pointing to that␣
 ↪contents.
   (…)
    387     This constructions solves problems with conflicting stylesheets and
 ↪navigation links.
    388 """
    389 from IPython.core.display import display
--> 391 from pandas_profiling.report.presentation.flavours.widget.notebook␣
 ↪import (
    392     get_notebook_iframe,
    393 )
    395 # Ignore warning: https://github.com/ipython/ipython/pull/11350/files
    396 with warnings.catch_warnings():

File c:
 ↪\Users\josej\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas_ rofiling\rep
 ↪py:1, in <module>
```

```
----> 1 from pandas_profiling.report.presentation.flavours.widget.alerts import
 ↪WidgetAlerts
      2 from pandas_profiling.report.presentation.flavours.widget.collapse
 ↪import WidgetCollapse
      3 from pandas_profiling.report.presentation.flavours.widget.container
 ↪import (
      4     WidgetContainer,
      5 )

File c:
 ↪\Users\josej\AppData\Local\Programs\Python\Python39\lib\site-packages\pandas_
 ↪py:3, in <module>
      1 from typing import List
----> 3 from ipywidgets import HTML, Button, widgets
      5 from pandas_profiling.report.presentation.core import Alerts
      6 from pandas_profiling.report.presentation.flavours.html import template

ModuleNotFoundError: No module named 'ipywidgets'
```

[ ]:

2.1 Diga el tipo de cada una de las variables del dataset (cualitativa o categórica, cuantitativa continua, cuantitativa discreta)

Tal cual se puede observar en los resultados obtenidos mediante el *profiling* se cuenta con un total de 51 variables cualitativas, las cuales a continuación se separan entre cualitativas ordinales y nominales, y se cuenta con 29 variables numéricas, las cuales a su vez, se separan en continuas y discretas. A continuación se presenta una lista con las variables y lo que significan, separada por tipo de variables.

Cualitativas ordinales

- **LotShape:** general shape of the property
- **LandContour:** flatness of the property
- **LandSlope:** slope of the property
- **BldgType:** type of dwelling
- **OverallQual:** rates the overall material and finish of the house
- **OverallCond:** rates the overall condition of the house
- **ExterQual:** evaluates the quality of the material on the exterior
- **ExterCond:** evaluates the present condition of the material on the exterior
- **BsmtQual:** evaluates the height of the basement
- **BsmtCond:** evaluates the general conditions of the basement
- **BsmtExposure:** refers to walkout or garden level walls
- **BsmtFinType1:** rating of basement finished area
- **BsmtFinType2:** rating of basement finished area (if multiple types)
- **HeatingQC:** heating quality and condition
- **KitchenQual:** kitchen quality
- **FireplaceQu:** fireplace qualityt

- **GarageFinish:** interior finish of the garage
- **GarageQual:** garage quality
- **GarageCond:** garage condition
- **PoolQC:** pool quality
- **Fence:** fence quality

Cualitativas nominales

- **MSSubclass:** identifies the type of dwelling involved in the sale
- **MSZoning:** identifies the general zoning classification of hte sale
- **Street:** type of road access to property
- **Alley:** type of alley access to property
- **Utilities:** type of utilities available
- **LotConfig:** lot configuration
- **Neighborhood:** physical locations within Ames city limits
- **Condition1:** proximity to various conditions
- **Condition2:** proximity to various conditions
- **HouseStyle:** style of dwelling
- **YearBuilt:** original construction date
- **YearRemodAdd:** remodel date (same as construction if no remodeling or additions)
- **RoofStyle:** type of roof
- **RoofMatl:** roof material
- **Exterior1st:** exterior covering on house
- **Exterior2nd:** exterior covering on house
- **MasVnrType:** masonry veneer type
- **Foundation:** type of foundation
- **Heating:** type of heating
- **CentralAir:** central air conditioning
- **Electrical:** electrical system
- **Functional:** home functionality*
- **GarageType:** garage location
- **GarageYrBlt:** year garage was built
- **PavedDrive:** paved driveway*
- **MiscFeature:** miscellaneous feature not covered in other categories
- **MoSold:** month sold
- **YrSold:** year sold
- **SaleType:** type of sale
- **SaleCondition:** condition of sale

Cuantitativas Continuas

- **LotFrontage:** linear feet of street connected to property
- **LotArea:** lot size in square feet
- **MasVnArea:** Masonry veneer area in square feet
- **BsmtFinSF1:** type 1 finished square feet
- **BsmtFinSF2:** type 2 finished square feet
- **BsmtUnfSF:** unfinished square feet of basement area
- **TotalBsmtSF:** total square feet of basement area

- **1stFlrSF:** first Floor square feet
- **2ndFlrSF:** second floor square feet
- **LowQualFinSF:** low quality finished square feet (all floors)
- **GrLivArea:** above grade (ground) living area square fee
- **GarageArea:** size of garage in square feet
- **WoodDeckSF:** wood deck area in square feet
- **OpenPorchSF:** open porch area in square feet
- **EnclosedPorch:** enclosed porch area in square feet
- **3SsnPorch:** three season porch area in square feet
- **ScreenPorch:** screen porch area in square feet
- **PoolArea:** pool area in square feet
- **SalePrice:** price of the property

Cuantitativas Discretas

- **BsmtFullBath:** basement full bathrooms
- **BsmtHalfBath:** basement half bathrooms
- **FullBath:** full bathrooms above grade
- **HalfBath:** half baths above grade
- **Bedroom:** bedrooms above grade (does NOT include basement bedrooms)
- **Kitchen:** kitchens above grade
- **TotRmsAbvGrd:** total rooms above grade (does not include bathrooms)
- **Fireplaces:** number of fireplaces
- **GarageCars:** size of garage in car capacity
- **MiscVal:** value of miscellaneous feature

2.2 Incluya los gráficos exploratorios siendo consecuentes con el tipo de variable que están representando

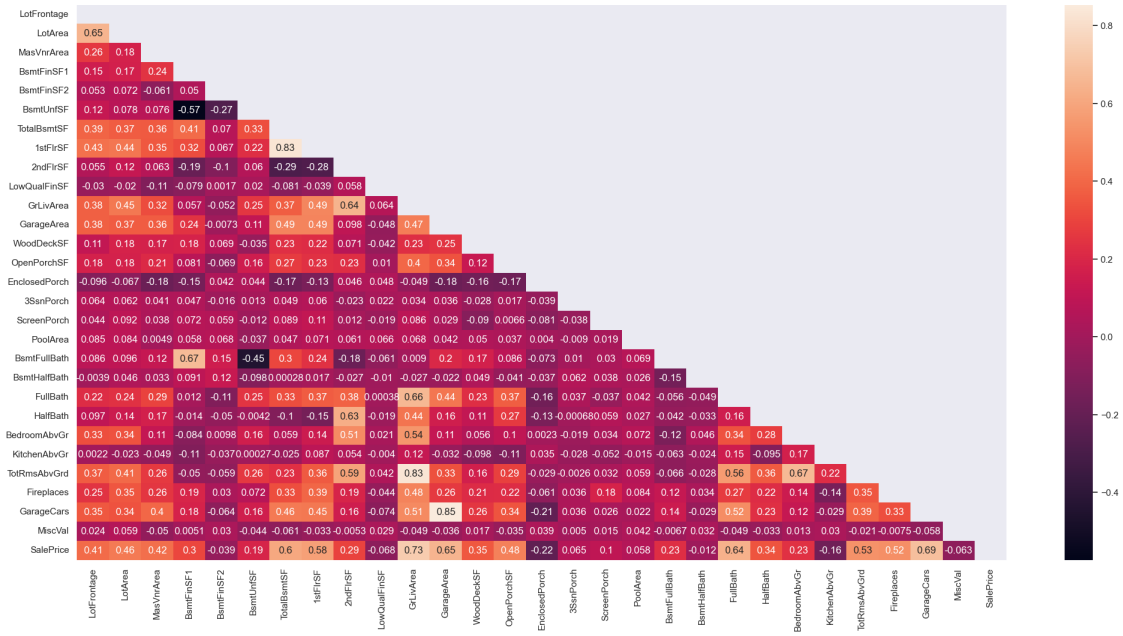
Los gráficos se pueden observar en el reporte anterior obtenido mediante la librería *profiling*

2.3 Aísle las variables numéricas de las categóricas, haga un análisis de correlación entre las mismas.

[ ]:
```
# quantitative variables
quantitative = [
    'LotFrontage',
    'LotArea',
    'MasVnrArea',
    'BsmtFinSF1',
    'BsmtFinSF2',
    'BsmtUnfSF',
    'TotalBsmtSF',
    '1stFlrSF',
    '2ndFlrSF',
    'LowQualFinSF',
    'GrLivArea',
    'GarageArea',
```

```
    'WoodDeckSF',
    'OpenPorchSF',
    'EnclosedPorch',
    '3SsnPorch',
    'ScreenPorch',
    'PoolArea',
    'BsmtFullBath',
    'BsmtHalfBath',
    'FullBath',
    'HalfBath',
    'BedroomAbvGr',
    'KitchenAbvGr',
    'TotRmsAbvGrd',
    'Fireplaces',
    'GarageCars',
    'MiscVal',
    'SalePrice'
]
```

[ ]:
```
quantitative_data = data[quantitative]
correlation = quantitative_data.corr(method = 'spearman')
plt.figure(figsize=(25,12))
matrix = np.triu(correlation)
sns.heatmap(correlation, annot=True, mask=matrix)
plt.show()

del correlation, matrix
```



Del análisis de correlación elaborado anteriormente se puede observar que 22 variables cuentan con una correlación significativa, es decir igual o por arriba de 0.5. Entre estas se tienen, el área de garage con el precio al cual se vende una casa, así como la cantidad total de cuartos con el espacio de área verde disponible.

2.4 Utilice las variables categóricas, haga tablas de frecuencia, proporción, gráficas de barras o cualquier otra técnica que le permita explorar los datos

¿Cuál es el estilo de vivienda predominante? Es estilo de vivienda predominante o el que más se ha vendido según los datos de entrenamiento son aquellas casa de 1 piso estilo 1946 y más pisos con diferentes estilos.

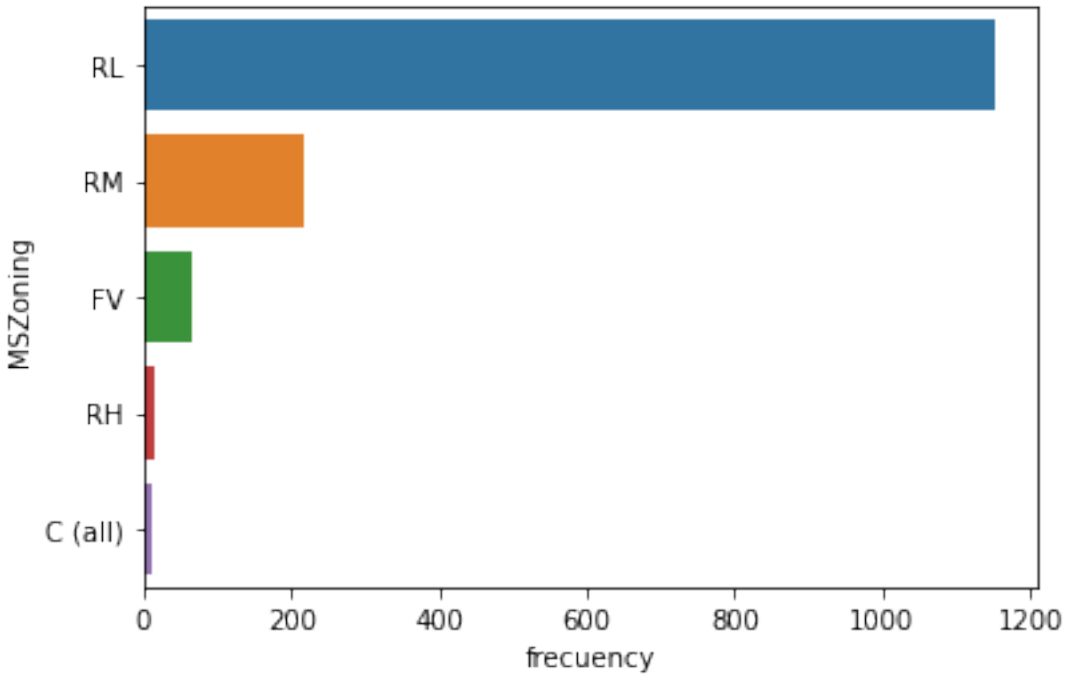
[ ]:
```
feature = calculate_frecuency(data, 'MSSubClass')
del feature
```

```
<pandas.io.formats.style.Styler at 0x7f304df30580>
```

¿En qué zona se encuentran las casas más vendidas? Las zonas en las cuales se encuentran las casas más vendidas son en residenciales con baja densidad. Algo interesante es que no existen casas en zonas industriales o de agricultura.

[ ]:
```
zoning = calculate_frecuency(data, 'MSZoning')
sns.barplot(x='frecuency', y='MSZoning', data=zoning)
del zoning
```

```
<pandas.io.formats.style.Styler at 0x7f3045efd310>
```



¿Son mayores las ventas si el tipo de vía de acceso a la propiedad es pavimentado? Sí son mayores las ventas con la vía de acceso pavimentada. De hecho, el 99.5% de las casa vendidas, en los datos de entrenamiento, tienen acceso pavimentado.

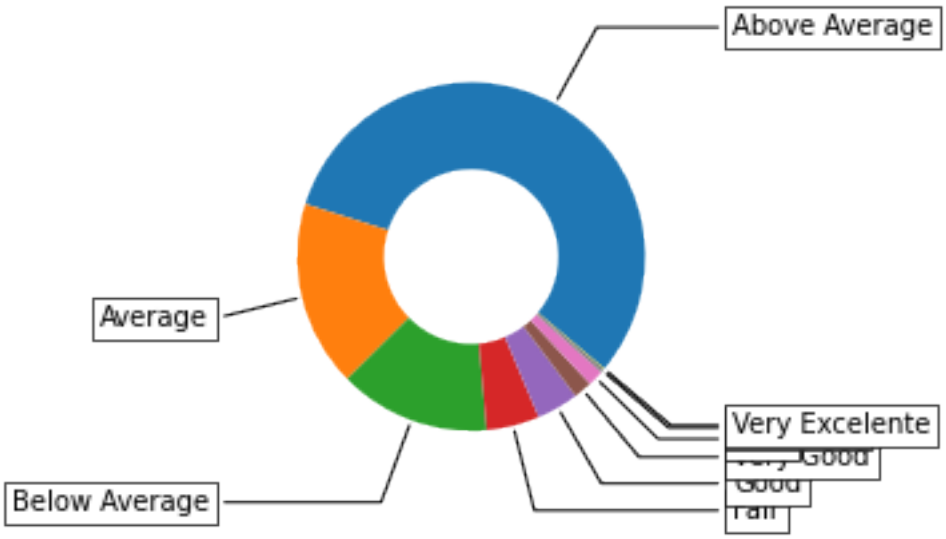
[ ]:
```
street = calculate_frecuency(data, 'Street')
del street
```

<pandas.io.formats.style.Styler at 0x7f3045f3f580>

¿La mayor cantidad de casas que se venden se encuentran en excelente estado? No, las casas que más se venden son las que se encuentran en un estado promedio, aquellas que no están ni excelente pero tampoco mal. Y estás representan un 56% de las ventas totales.

[ ]:
```
overall = calculate_frecuency(data, 'OverallCond')
# giving the numbers a cualitative meaning
overall['OverallCond'] = overall['OverallCond'].replace([1, 2, 3, 4, 5, 6, 7,␣
 ↪8, 9, 10], ['Very Excelente', 'Excelente', 'Very Good', 'Good', 'Above␣
 ↪Average', 'Average', 'Below Average', 'Fair', 'Poor', 'Very Poor'])

fig, ax = plt.subplots(figsize=(6, 3), subplot_kw=dict(aspect="equal"))

wedges, texts = ax.pie(overall['frecuency'], wedgeprops=dict(width=0.5),␣
 ↪startangle=-40)

bbox_props = dict(boxstyle="square,pad=0.3", fc="w", ec="k", lw=0.72)
kw = dict(arrowprops=dict(arrowstyle="-"),
          bbox=bbox_props, zorder=0, va="center")

for i, p in enumerate(wedges):
    ang = (p.theta2 - p.theta1)/2. + p.theta1
    y = np.sin(np.deg2rad(ang))
    x = np.cos(np.deg2rad(ang))
    horizontalalignment = {-1: "right", 1: "left"}[int(np.sign(x))]
    connectionstyle = "angle,angleA=0,angleB={}".format(ang)
    kw["arrowprops"].update({"connectionstyle": connectionstyle})
    ax.annotate(overall['OverallCond'][i], xy=(x, y), xytext=(1.5*np.sign(x), 1.
 ↪5*y),
                horizontalalignment=horizontalalignment, **kw)
plt.show()
```

<pandas.io.formats.style.Styler at 0x7f304c994610>



La mayoría de viviendas cuentan con un sistema de aire acondicionado central y calefacción en buena calidad Al menos un 96% de las viviendas cuentan con calefacción en condiciones promedio y un 93.5% cuenta con aire acondicionado central

[ ]:
```
heating = calculate_frecuency(data, 'HeatingQC')
del heating
```

```
<pandas.io.formats.style.Styler at 0x7f304c1bd310>
```

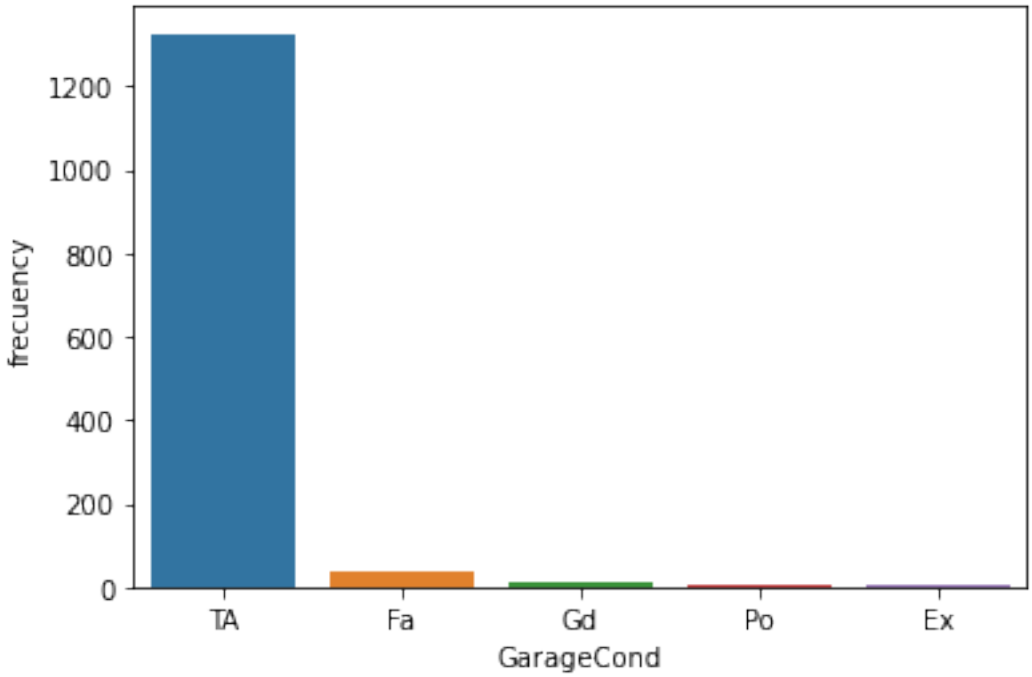
[ ]:
```
central = calculate_frecuency(data, 'CentralAir')
del central
```

```
<pandas.io.formats.style.Styler at 0x7f304c1bd970>
```

Al menos un 50% de las viviendas cuenta con garage en buenas condiciones Un 96% de los garages de las viviendas se encuentran en condiciones promedio, y solo apenas un 0.802 tiene garage en buenas condiciones

[ ]:
```
garage_cond = calculate_frecuency(data, 'GarageCond')
sns.barplot(x='GarageCond', y='frecuency', data=garage_cond)
del garage_cond
```

```
<pandas.io.formats.style.Styler at 0x7f304d332160>
```

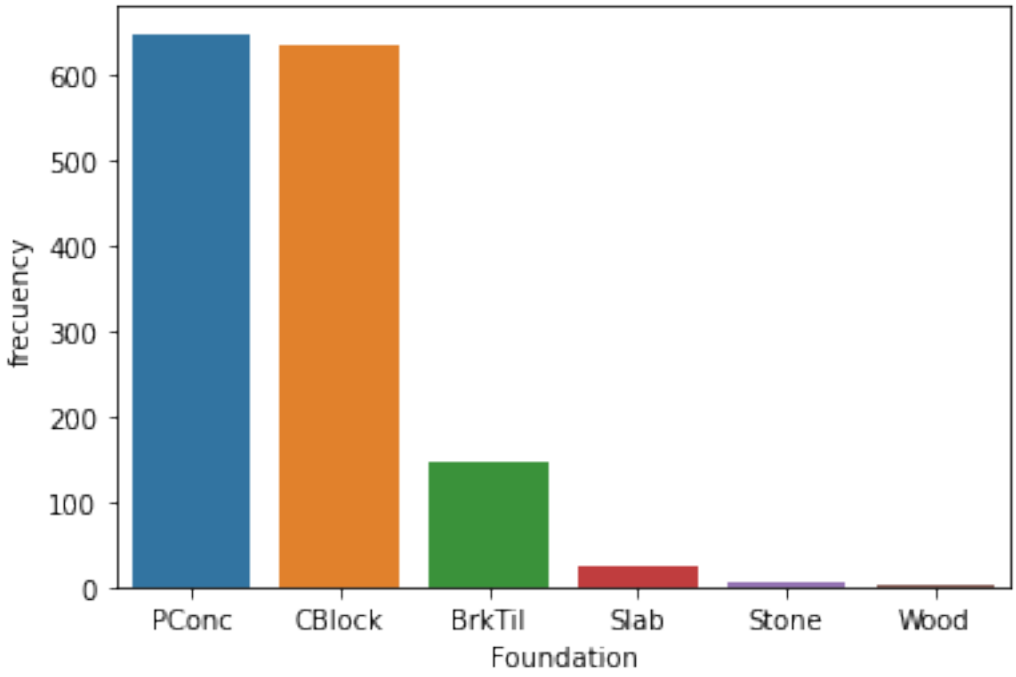


¿Las personas prefieren casas fundidas en concreto vertido? Sí, al menos un 44% de las prefieren las casas de cemento vertido, sin embargo, un 43% prefieren las casas de bloques de cemento.

¿Cuáles son los precios más altos por los cuales se ha vendido una casa? De acuerdo con los resultados obtenidos se sabe que el rango de precios en el que están las casas con los precios más altos es de 755000 hasta 485000.

[ ]:
```
foundation = calculate_frecuency(data, 'Foundation')
sns.barplot(x='Foundation', y='frecuency', data=foundation)
del foundation
```

```
<pandas.io.formats.style.Styler at 0x7f3045f8d340>
```



[ ]:
```
sale_prices  = calculate_frecuency(data, 'SalePrice')
del sale_prices
```

```
<pandas.io.formats.style.Styler at 0x7f304cc9a5b0>
```

3 Estudie si es conveniente hacer un Análisis de Componentes Principales. Recuerde que puede usar el índice KMO y el test de esfericidad de Bartlett. Haga un análisis de componentes principales con las variables numéricas, discuta los resultados e interprete los componentes.

[ ]:
```
# Deleting null values
quantitative_data = quantitative_data.dropna()
```

[ ]:
```
# KMO test
from factor_analyzer.factor_analyzer import calculate_kmo
all, model = calculate_kmo(quantitative_data)
model
```

[ ]: 0.7452578732089676

La medida de suficiencia de muestreo, conocida comúnmente como KMO test, pretende explicar el nivel de correlación entre las variables, es decir, conocer cómo un *feature* es capaz de explicar

a otro. El valor varía de 0 a 1, siendo 1 el ideal y generalmente se considera que los *datasets* con valores por debajo de 0.5 son inaceptados para factor de análisis.

Como se puede notar, el valor de KMO obtenido para el conjunto de datos de los precios de las casas es de **0.745** que si bien, es un valor por debajo de lo que se considera bueno, el cual es 0.80, aun así es un valor aceptable, ya que demuestra que existe correlación parcial entre las variables.

[ ]:
```
from factor_analyzer.factor_analyzer import calculate_bartlett_sphericity
chi_square ,p_value = calculate_bartlett_sphericity(quantitative_data)
chi_square , p_value
```

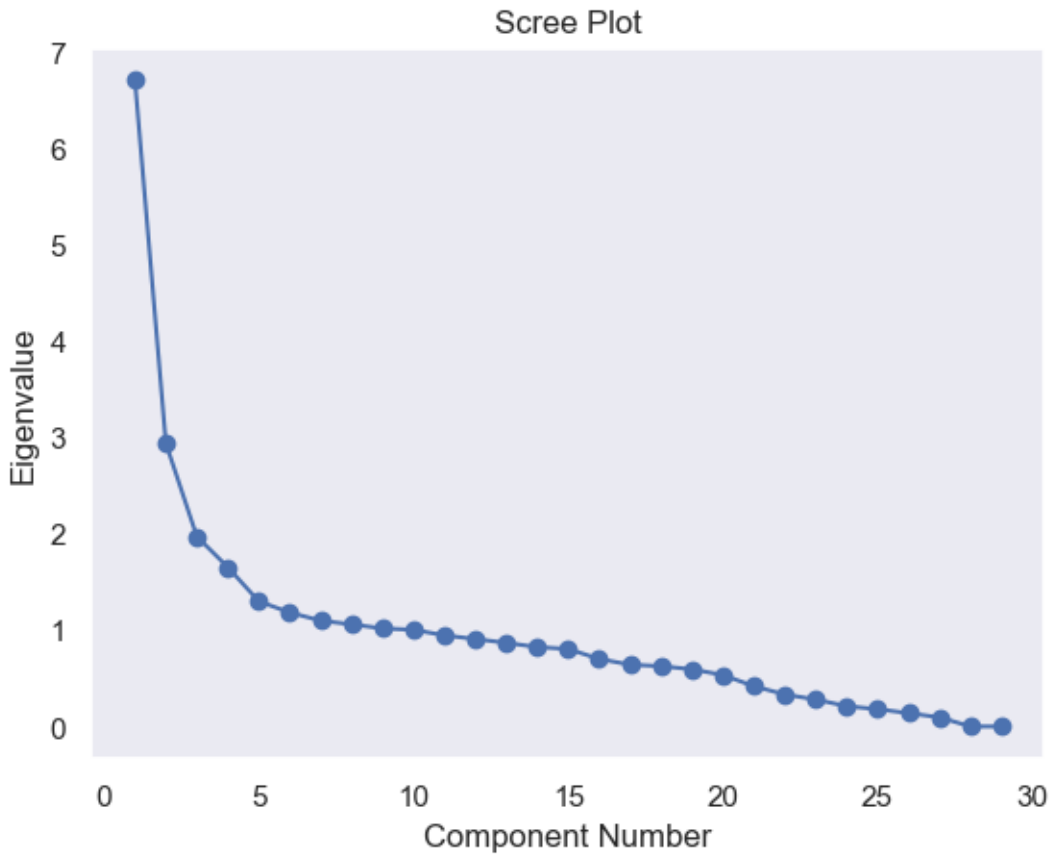
[ ]:
```
(96925.67307595348, 0.0)
```

El test de esfericidad de Barlett se basa en probar la hipótesis nula, de que la matriz de correlación de los datos es una matriz identidad. Para la validación de la hipótesis nula se utiliza un valor de significancia de 0.05, por lo tal, como se puede observar, en la prueba realizada a los datos se obtuvo un *p-value* de 0.00 rechazando así la hipótesis nula, validando que las variables sí se encuentran relacionadas.

Tanto con el KMO test como el test de esfericidad se pudo validar que los datos sí son aptos para realizar análisis de factores, ya que sí existe relación entre ellos.

[ ]:
```
# Standardize the data
from sklearn.preprocessing import StandardScaler
quantitative_data = StandardScaler().fit_transform(quantitative_data)
```

[ ]:
```
# Find the more adequate number of factors
factor = FactorAnalyzer()
factor.fit(quantitative_data)
# return the eigenvalues to know how many factors are the ideal
eigenvalues, values = factor.get_eigenvalues()
# create the scree plot to see which factors explain the more variance of the␣
 ↪data
plt.scatter(range(1,30),eigenvalues)
plt.plot(range(1,30),eigenvalues)
plt.title('Scree Plot')
plt.xlabel('Component Number')
plt.ylabel('Eigenvalue')
plt.grid()
plt.show()
```



La gráfica anterior se interpreta de manera muy similar a un gráfico de codo en el caso de *clustering*. Por lo tal, se puede observar como el número ideal de componentes en los cuales se puede simplificar el conjunto de datos es 5, esto debido a que luego de 5 componentes la variabilidad aportada a los datos no es tan significativa, es decir, el esfuerzo que se debe realizar no es compensado por la información de los nuevos componentes.

[ ]:
```
factor = FactorAnalyzer(n_factors=5)
factor.fit(quantitative_data)
loadings = pd.DataFrame(data=factor.loadings_, columns=['pc1', 'pc2', 'pc3',␣
 ↪'pc4', 'pc5'], index=quantitative)
variance = pd.DataFrame(data=factor.get_factor_variance(), columns=['pc1',␣
 ↪'pc2', 'pc3', 'pc4', 'pc5'])
display(loadings)
display(variance)
```

	pc1	pc2	pc3	pc4	pc5
LotFrontage	0.571265	-0.052240	-0.002060	-0.054373	0.081792
LotArea	0.469366	-0.018071	0.072664	-0.064055	-0.005722
MasVnrArea	0.253498	0.159298	0.060540	0.304560	-0.006409
BsmtFinSF1	0.400095	-0.120785	0.757942	0.233137	0.062087

```
BsmtFinSF2     0.138925 -0.083554  0.115790 -0.112644 -0.053654
BsmtUnfSF      0.462826 -0.248800 -0.980579  0.041246 -0.069048
TotalBsmtSF    0.922702 -0.386867 -0.067293  0.220888 -0.016124
1stFlrSF       0.981993 -0.451102 -0.007498  0.140697  0.225033
2ndFlrSF      -0.226183  1.055046 -0.035367 -0.082888 -0.017801
LowQualFinSF   0.131109  0.036924 -0.080171 -0.252907  0.040294
GrLivArea      0.568685  0.526197 -0.048896 -0.012288  0.158577
GarageArea     0.244552  0.100410  0.086459  0.699315  0.120360
WoodDeckSF     0.157775  0.119584  0.124118  0.207226 -0.011307
OpenPorchSF    0.222488  0.174312 -0.072609  0.114873 -0.086854
EnclosedPorch -0.023620  0.000281 -0.032938 -0.228086  0.009337
3SsnPorch      0.037345 -0.033945 -0.019477  0.033622 -0.016199
ScreenPorch    0.164516  0.080962 -0.033201 -0.057529 -0.157817
PoolArea       0.272500  0.019294  0.080619 -0.148493 -0.036837
BsmtFullBath   0.204262 -0.135001  0.655330  0.173181  0.082489
BsmtHalfBath   0.005635  0.011836  0.017957 -0.042108 -0.078914
FullBath       0.266943  0.252968 -0.091550  0.241660  0.289668
HalfBath      -0.173284  0.735942  0.019476  0.112161 -0.243321
BedroomAbvGr   0.229512  0.365573 -0.096881 -0.304019  0.346176
KitchenAbvGr  -0.023074 -0.083301  0.113847 -0.150517  0.583492
TotRmsAbvGrd   0.418559  0.458242 -0.037115 -0.109598  0.441385
Fireplaces     0.449369  0.200329  0.002905  0.063376 -0.202763
GarageCars     0.142521  0.180375  0.053896  0.769700  0.134300
MiscVal        0.031230  0.018976  0.027443 -0.130079  0.014826
SalePrice      0.511864  0.273369  0.037754  0.438045 -0.078284
```

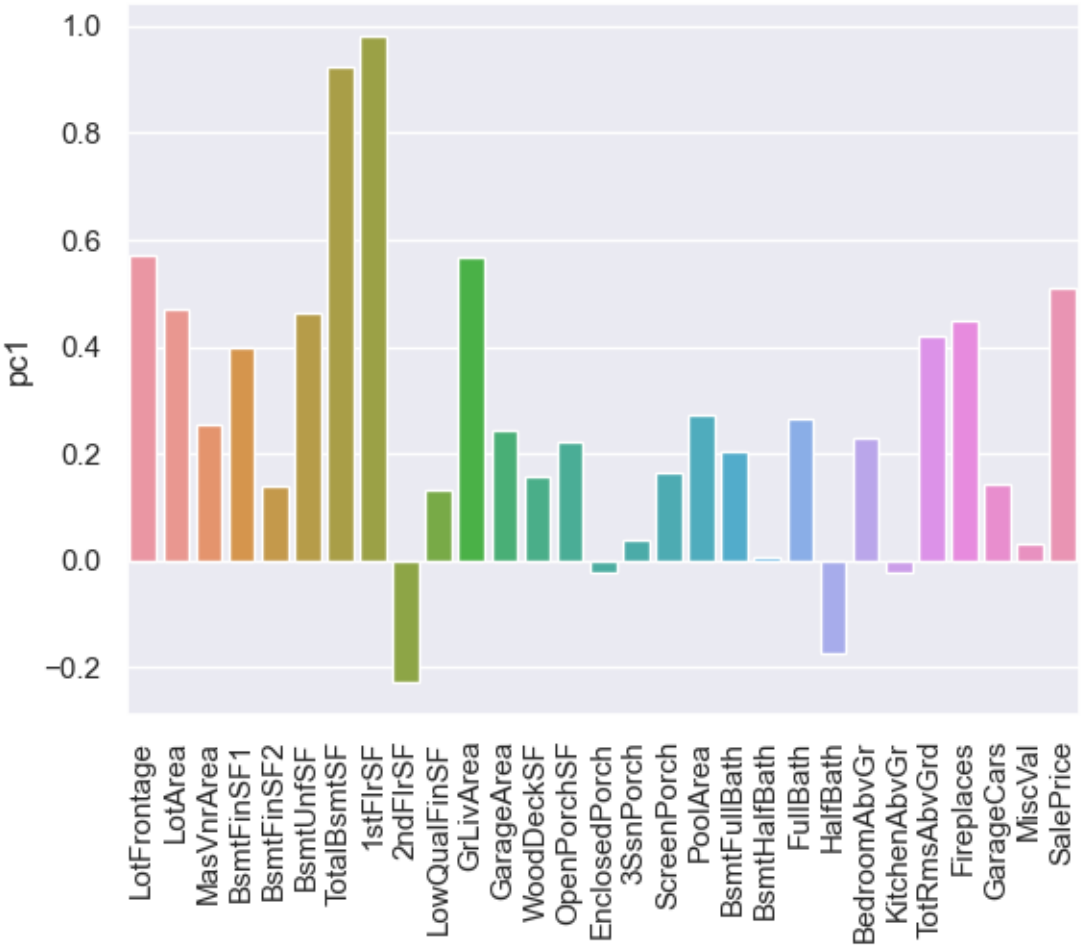
	pc1	pc2	pc3	pc4	pc5
0	4.306073	3.041572	2.077597	1.966920	1.021785
1	0.148485	0.104882	0.071641	0.067825	0.035234
2	0.148485	0.253367	0.325008	0.392833	0.428067

Mediante la varianza es posible conocer el porcentaje de información que aporta cada uno de los componentes del conjunto de datos original. Como se puede observar en la tabla anterior, el componente 1, el cual contribuye en un 15% a la variación de los datos, en conjunto con el componente 2, el cual aporta un 11% a la variación, son los dos componentes que en conjunto aportan el mayor porcentaje de información a diferencia del resto de componentes que en conjunto solamente suman un 16%. Como se puede observar, en total se cuenta con un 43% de la información de los datos generales. Esto implica que se perdió aproximadamente un 57% de la información original, sin embargo, aún es posible utilizar una diversidad de técnicas de aprendizaje para obtener información de estos, tal como el poder conocer las características que persisten en los datos para luego poder agruparlos.

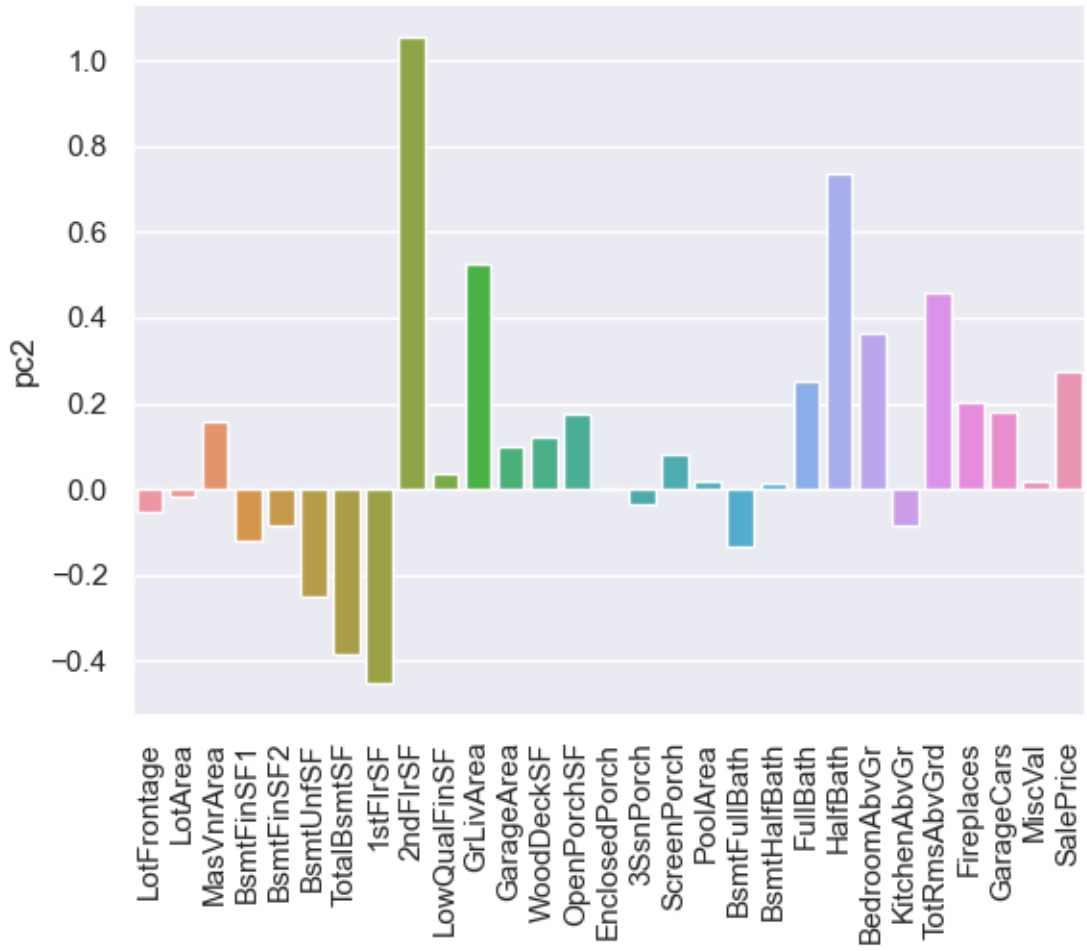
Por otro lado, una de las estrategias que se podría utilizar para analizar si con eso es posible obtener una mayor información de los datos, es antes de realizar un análisis de componentes principales utilizar una estrategia de *feature selection* con lo cual se permitan obtener aquellas variables que son las que más delimitan el valor de la variable objetivo, y ya sobre este conjunto, ya reducido, aplicar un anáisis de componentes principales. Algo interesante que se realizó al momento de desarrollar el análisis de componentes es probar también únicamente con dos componentes principales, con los cuales, no se obtenía una porción significativa de la varianza de los datos originales. Por otro lado, en

busca de poder obtener un mayor porcentaje de información se optó por utilizar 10 componentes, sin embargo, la varianza obtenida de esos últimos cinco componentes no era significativa con respecto a los primeros cinco componentes.

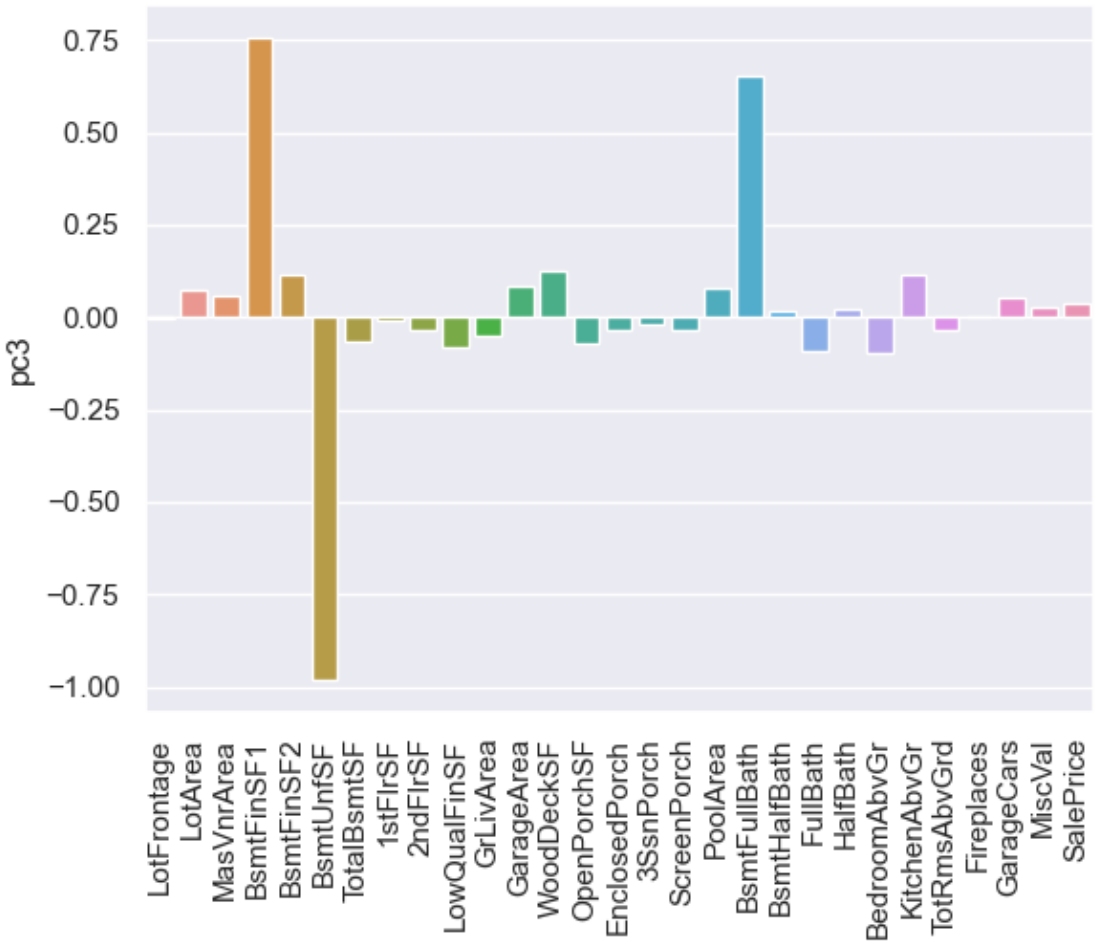
[ ]:
```
# Analyzing the loadings
ax = sns.barplot(x=loadings.index, y='pc1', data=loadings)
ax.tick_params(axis='x', rotation=90)
```



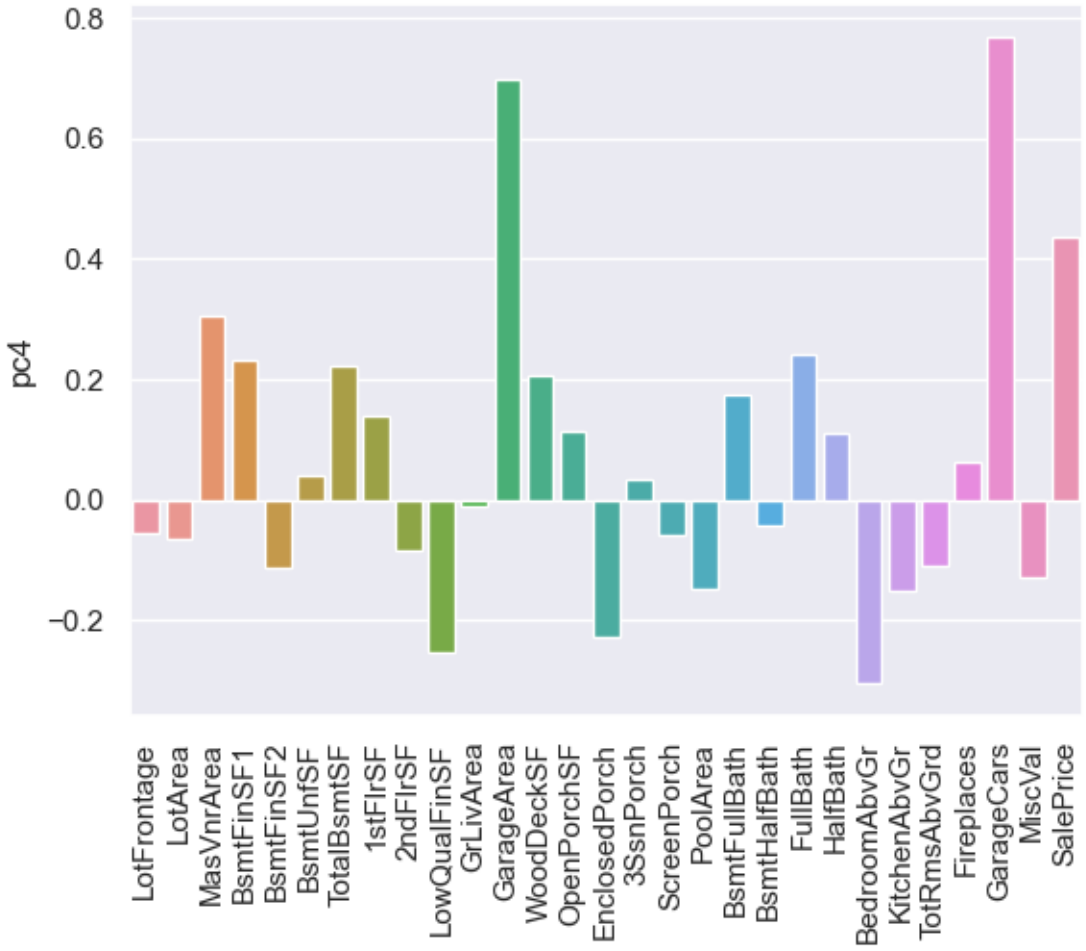
[ ]:
```
ax = sns.barplot(x=loadings.index, y='pc2', data=loadings)
ax.tick_params(axis='x', rotation=90)
```



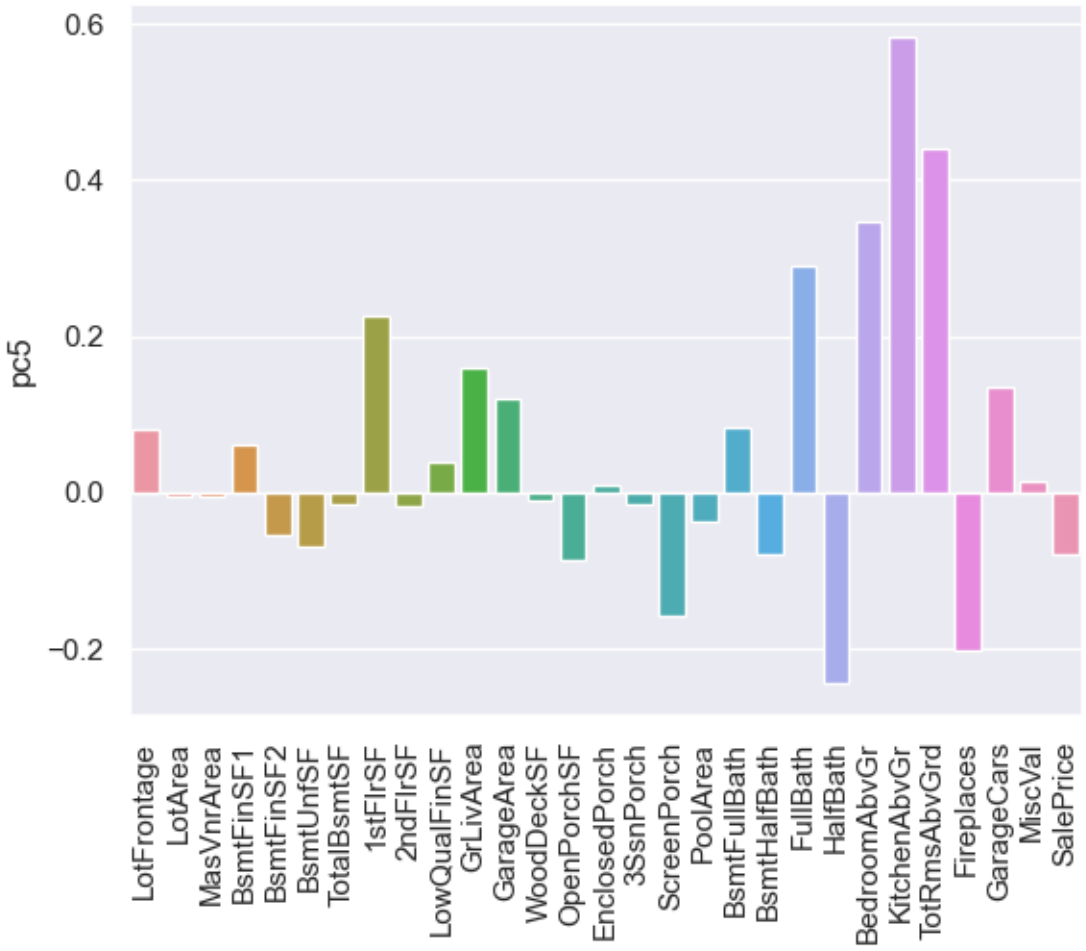
[ ]:
```
ax = sns.barplot(x=loadings.index, y='pc3', data=loadings)
ax.tick_params(axis='x', rotation=90)
```



[ ]:
```
ax = sns.barplot(x=loadings.index, y='pc4', data=loadings)
ax.tick_params(axis='x', rotation=90)
```



[ ]:
```
ax = sns.barplot(x=loadings.index, y='pc5', data=loadings)
ax.tick_params(axis='x', rotation=90)
```



Si bien, mediante la varianza es posible conocer qué tanto aporta un componente a la varianza de los datos, también es necesario saber cuál es la interpretación de cada uno de los nuevos componentes, para esto se utilizan los *loadings*. Los *loadings* permiten conocer qué variables son las que más "influencian" los componentes principales. Por lo tal, en las imágenes anteriores, se graficó cada uno de los componentes, con cada uno de los valores de *loading* correspondiente para cada variable, mediante esto es posible notar que el primer componente se encuentran estrechamente influenciado por el tamaño de la propiedad en general, desde el área libre disponible, hasta el tamaño del primer piso de construcción. Por otro lado, el segundo componente se puede interpretar como el espacio disponible del segundo piso de construcción, luego se tiene el tercer componente el cual está altamente relacionado con el espacio disponible para el sótano de la propiedad, seguido se tienen el cuarto componente, con el cual es posible describir el área de garage de la casa y finalmente se tiene el quinto componente el cual permite describir el interior de la casa, desde la cantidad de cuartos disponibles hasta la cantidad de cocinas.

4 Obtenga reglas de asociación interesantes del dataset. Discuta sobre el nivel de confianza y soporte

[ ]:
```
#Calculo de las variables cualitativas basado en remover las cuantitativas
# cualitatives = [x for x in list(data.keys()) if x not in quantitative]

cualitatives = ['MSSubClass',
  'MSZoning',
  'Street',
  'Alley',
  'LotShape',
  'LandContour',
  'Utilities',
  'LotConfig',
  'LandSlope',
  'Neighborhood',
  'Condition1',
  'Condition2',
  'BldgType',
  'HouseStyle',
  'OverallQual',
  'OverallCond',
  'YearRemodAdd',
  'RoofStyle',
  'RoofMatl',
  'Exterior1st',
  'Exterior2nd',
  'MasVnrType',
  'ExterQual',
  'ExterCond',
  'Foundation',
  'BsmtQual',
  'BsmtCond',
  'BsmtExposure',
  'BsmtFinType1',
  'BsmtFinType2',
  'Heating',
  'HeatingQC',
  'CentralAir',
  'Electrical',
  'KitchenQual',
  'Functional',
  'FireplaceQu',
  'GarageType',
  'GarageYrBlt',
  'GarageFinish',
  'GarageQual',
```

```
  'GarageCond',
  'PavedDrive',
  'PoolQC',
  'Fence',#
  'MiscFeature',
  'MoSold',
  'YrSold',
  'SaleType',
  'SaleCondition']


(cualitatives, len(cualitatives))
```

[ ]:
```
(['MSSubClass',
  'MSZoning',
  'Street',
  'Alley',
  'LotShape',
  'LandContour',
  'Utilities',
  'LotConfig',
  'LandSlope',
  'Neighborhood',
  'Condition1',
  'Condition2',
  'BldgType',
  'HouseStyle',
  'OverallQual',
  'OverallCond',
  'YearRemodAdd',
  'RoofStyle',
  'RoofMatl',
  'Exterior1st',
  'Exterior2nd',
  'MasVnrType',
  'ExterQual',
  'ExterCond',
  'Foundation',
  'BsmtQual',
  'BsmtCond',
  'BsmtExposure',
  'BsmtFinType1',
  'BsmtFinType2',
  'Heating',
  'HeatingQC',
  'CentralAir',
  'Electrical',
```

```
  'KitchenQual',
  'Functional',
  'FireplaceQu',
  'GarageType',
  'GarageYrBlt',
  'GarageFinish',
  'GarageQual',
  'GarageCond',
  'PavedDrive',
  'PoolQC',
  'Fence',
  'MiscFeature',
  'MoSold',
  'YrSold',
  'SaleType',
  'SaleCondition'],
 50)
```

[ ]:
```
print(data[cualitatives].shape)
```

```
(1460, 50)
```

Podemos observar que hay 50 variables categóricas y que estas poseen, al igual que en el dataset original, 1460 datos únicos de casas

[ ]:
```
cualitatives_df = data[cualitatives].astype(str)


for n in cualitatives:
    cualitatives_df[n] = cualitatives_df[n].apply(lambda x: n + '-' + str(x))
```

[ ]:
```
# para transformar los datos a listas
records = []
for i in range(0, len(cualitatives_df)):
    records.append([str(cualitatives_df.values[i,j]) for j in range(0,␣
 ↪len(cualitatives))])
```

[ ]:
```
#Reglas de asociación
reglas_asociacion = apriori(transactions = records, min_support = 0.003,␣
 ↪min_confidence = 0.2, min_lift = 3, min_length = 2, max_length = 2)
output = list(reglas_asociacion)
len(output)
```

[ ]:
```
1144
```

Podemos observar que encontró 1144 reglas de asociación dentro del dataset de variables cualitativas

[ ]:
```
#Función para transformación del tipo de dato apryori a un dataframe, tomado␣
 ↪del ejemplo de apriori en clase
```

```
def inspect(output):
    lhs         = [tuple(result[2][0][0])[0] for result in output]
    rhs         = [tuple(result[2][0][1])[0] for result in output]
    support    = [result[1] for result in output]
    confidence = [result[2][0][2] for result in output]
    lift       = [result[2][0][3] for result in output]
    return list(zip(lhs, rhs, support, confidence, lift))
```

[ ]:
```
output_DataFrame = pd.DataFrame(inspect(output), columns = ['Left_Hand_Side',␣
↪'Right_Hand_Side', 'Support', 'Confidence', 'Lift'])
output_DataFrame
```

[ ]:

	Left_Hand_Side	Right_Hand_Side	Support	Confidence	Lift
0	Alley-Grvl	CentralAir-N	0.010959	0.320000	4.917895
1	Alley-Grvl	Condition1-Artery	0.007534	0.220000	6.691667
2	Exterior1st-AsbShng	Alley-Grvl	0.003425	0.250000	7.300000
3	Exterior1st-Stucco	Alley-Grvl	0.004110	0.240000	7.008000
4	Exterior2nd-Stucco	Alley-Grvl	0.004110	0.230769	6.738462
…	…	…	…	…	…
1139	YearRemodAdd-2008	SaleType-New	0.011644	0.425000	5.086066
1140	YearRemodAdd-2009	SaleType-New	0.011644	0.739130	8.845331
1141	YearRemodAdd-2010	SaleType-New	0.003425	0.833333	9.972678
1142	YearRemodAdd-2009	YrSold-2009	0.011644	0.739130	3.192694
1143	YearRemodAdd-2010	YrSold-2010	0.004110	1.000000	8.342857

[1144 rows x 5 columns]

4.0.1 Datos ordenados por soporte

[ ]:
```
output_DataFrame.sort_values(by=['Support'], ascending=False).head(20)
```

[ ]:

	Left_Hand_Side	Right_Hand_Side	Support	Confidence	\
906	HouseStyle-2Story	MSSubClass-60	0.204110	0.669663	
403	Exterior1st-MetalSd	Exterior2nd-MetalSd	0.145205	0.963636	
386	Exterior1st-HdBoard	Exterior2nd-HdBoard	0.132192	0.869369	
441	Exterior1st-Wd Sdng	Exterior2nd-Wd Sdng	0.121233	0.859223	
892	HouseStyle-1.5Fin	MSSubClass-50	0.096575	0.915584	
1130	SaleCondition-Partial	SaleType-New	0.083562	0.976000	
410	Exterior1st-Plywood	Exterior2nd-Plywood	0.065753	0.888889	
1018	MSZoning-RM	Neighborhood-OldTown	0.065068	0.435780	
629	GarageFinish-nan	GarageYrBlt-nan	0.055479	1.000000	
609	GarageCond-nan	GarageQual-nan	0.055479	1.000000	
610	GarageCond-nan	GarageType-nan	0.055479	1.000000	
611	GarageCond-nan	GarageYrBlt-nan	0.055479	1.000000	
627	GarageFinish-nan	GarageQual-nan	0.055479	1.000000	
628	GarageFinish-nan	GarageType-nan	0.055479	1.000000	
608	GarageCond-nan	GarageFinish-nan	0.055479	1.000000	

```
649          GarageQual-nan        GarageType-nan  0.055479    1.000000
650          GarageQual-nan      GarageYrBlt-nan  0.055479    1.000000
681          GarageType-nan      GarageYrBlt-nan  0.055479    1.000000
68          BldgType-TwnhsE       MSSubClass-120  0.054110    0.692982
581       Foundation-BrkTil    YearRemodAdd-1950  0.052740    0.527397

           Lift
906    3.269926
403    6.574342
386    6.131784
441    6.367848
892    9.283009
1130  11.680000
410    9.139280
1018   5.630429
629   18.024691
609   18.024691
610   18.024691
611   18.024691
627   18.024691
628   18.024691
608   18.024691
649   18.024691
650   18.024691
681   18.024691
68    11.629361
581    4.325843
```

Basados en las reglas de asociación que presentan una mayor confianza podemos ver que en un 20% de los casos aparece la regla de casas de 2 niveles relacionadas con un MSSubClass-60, lo cual nos indica que la venta de la casa fue realizada con una casa intermediaria de 2 niveles construída a partir de 1946, esta tiene un lift de 3.26, lo cual nos dice que las ventas con una casa intermediaria de 2 niveles construída a partir de 1946 aumentan en un 326% cuando la casa vendida es de 2 niveles, lo cual se podría deber principalmente a que la casa intermediaria fue la casa que se compró.

Algo interesante que también se encontró fue una regla de asociación que relacióna los cimientos de ladrillo y teja de una casa con el año de remodelación o construcción en 1950, además el lift de esta regla de asociación nos indica que en un 4.32 mas de veces aparecen las casas de 1950 cuando hay cimientos de ladrillo y teja. Esto se explica ya que anteriormente los cimientos solían ser construidos con estos materiales de manera rustica, en cambio hoy en día se utiliza concreto y block y debido al deterioro muchos de estos cimientos antiguos deben ser reemplazados (Burnett and Burnett, 2013).

4.0.2 Datos ordenados por confianza

[ ]:
```
output_DataFrame.sort_values(by=['Confidence'], ascending=False).head(20)
```

[ ]:

	Left_Hand_Side	Right_Hand_Side	Support	Confidence	Lift
1143	YearRemodAdd-2010	YrSold-2010	0.004110	1.0	8.342857
609	GarageCond-nan	GarageQual-nan	0.055479	1.0	18.024691
880	Heating-Grav	HeatingQC-Fa	0.004795	1.0	29.795918
881	Heating-Grav	YearRemodAdd-1950	0.004795	1.0	8.202247
890	YearRemodAdd-1974	HeatingQC-TA	0.004795	1.0	3.411215
905	MSSubClass-160	HouseStyle-2Story	0.043151	1.0	3.280899
610	GarageCond-nan	GarageType-nan	0.055479	1.0	18.024691
908	Neighborhood-BrDale	HouseStyle-2Story	0.010959	1.0	3.280899
608	GarageCond-nan	GarageFinish-nan	0.055479	1.0	18.024691
1017	Neighborhood-MeadowV	MSZoning-RM	0.011644	1.0	6.697248
606	GarageCond-Po	GarageType-Detchd	0.004795	1.0	3.772610
557	GarageYrBlt-1922.0	Foundation-BrkTil	0.003425	1.0	10.000000
964	MSSubClass-180	MSZoning-RM	0.006849	1.0	6.697248
51	BldgType-Duplex	MSSubClass-90	0.035616	1.0	28.076923
250	Heating-Grav	CentralAir-N	0.004795	1.0	15.368421
1005	MSZoning-FV	Neighborhood-Somerst	0.044521	1.0	16.976744
91	BsmtCond-nan	BsmtExposure-nan	0.025342	1.0	38.421053
92	BsmtCond-nan	BsmtFinType1-nan	0.025342	1.0	39.459459
93	BsmtCond-nan	BsmtFinType2-nan	0.025342	1.0	38.421053
94	BsmtCond-nan	BsmtQual-nan	0.025342	1.0	39.459459

De las cosas interesantes que nos dicen este dataset ordenado por confianza es que siempre que se casa en 2010 se vendió la casa ese mismo año. Además algo que nos dice es que las casas que poseen calefacción gravitacional fue únicamente en remodelaciones y casas del 1950, lo cual es algo bastante explicable debido a que la calefacción gravitacional fue una tecnica antigua de calefacción

utilizada desde los años 1800 pero muy populares a mediados del siglo 20 (INTERNACHI, 2022). Finalmente de manera intuitiva nos dice que si la casa no tiene condición existente de el garage será siempre que no exista garage en la casa.

4.0.3 Datos ordenados por lift

[ ]:
```
output_DataFrame.sort_values(by=['Lift'], ascending=False).head(20)
```

[ ]:

	Left_Hand_Side	Right_Hand_Side	Support	Confidence	Lift
773	GarageYrBlt-1984.0	YearRemodAdd-1984	0.003425	0.625000	130.357143
465	Exterior2nd-Brk Cmn	Neighborhood-NPkVill	0.003425	0.714286	115.873016
761	GarageYrBlt-1975.0	YearRemodAdd-1975	0.004795	0.777778	113.555556
719	GarageYrBlt-1953.0	YearRemodAdd-1953	0.006164	0.750000	109.500000
897	HouseStyle-1.5Unf	MSSubClass-45	0.008219	0.857143	104.285714
1128	RoofMatl-Tar&Grv	RoofStyle-Flat	0.006849	0.909091	102.097902
757	GarageYrBlt-1973.0	YearRemodAdd-1973	0.006849	0.714286	94.805195
750	GarageYrBlt-1969.0	YearRemodAdd-1969	0.008904	0.866667	90.380952
734	GarageYrBlt-1961.0	YearRemodAdd-1961	0.004110	0.461538	84.230769
723	GarageYrBlt-1956.0	YearRemodAdd-1956	0.006164	0.562500	82.125000
746	GarageYrBlt-1967.0	YearRemodAdd-1967	0.006849	0.666667	81.111111
770	GarageYrBlt-1979.0	YearRemodAdd-1979	0.005479	0.533333	77.866667
739	GarageYrBlt-1963.0	YearRemodAdd-1963	0.007534	0.687500	77.211538
902	HouseStyle-2.5Unf	MSSubClass-75	0.006164	0.818182	74.659091
741	GarageYrBlt-1964.0	YearRemodAdd-1964	0.006849	0.555556	73.737374
771	GarageYrBlt-1980.0	YearRemodAdd-1980	0.006164	0.600000	73.000000
901	HouseStyle-2.5Fin	MSSubClass-75	0.004110	0.750000	91.125000
724	GarageYrBlt-1957.0	YearRemodAdd-1957	0.005479	0.400000	72.727273
358	Exterior1st-AsbShng	Exterior2nd-AsbShng	0.011644	0.850000	72.340426
745	GarageYrBlt-1966.0	YearRemodAdd-1966	0.008904	0.619048	72.000000

```
901    68.437500
724    64.888889
358    62.050000
745    60.253968
```

Con los datos ordenados por lift, podemos ver que en los años de 1984, 1975, 1953, 1973, 1969, 1961, 1956, 1967, 1979, 1963, 1964, 1980, 1957 y 1966 se logra apreciar que hay un lift mayor a 60 con la regla de asociación de año de construcción del garage y del año de remodelación de la casa, por lo que podemos decir que las remodelaciones aumentan cuando en el mismo año se construye el garage de la casa.

Además podemos observar una regla de asociación con el material del techo de grava y alquitrán y el tipo de techo plano, teniendo un lift de 102.09. por lo que podemos decir que los techos planos aumentan cuando el techo es de grava y alquitrán.

5 Conclusiones

Con base en el análisis de componentes principales se puede notar que si bien no se cuenta con una alta variabilidad del conjunto de datos original, sin embargo sí es posible el obtener información relevante de las características predominantes de los grupos de datos. Mediante estas características principales es posible utilizar algoritmos de *machine learning* que permitan clasificar y agrupar los datos, como *clustering*.

Con base a las reglas de asociación podemos decir que tan fuertemente asociadas están las variables con la confianza, así como se vio en el año de venta y de remodelación con confianza 1, que tanto aumentan las apariciones de una variable con respecto a otra con el lift, así como se vio en los techos de alquitran y grava que causaban que aumentaran los tipos de techo plano, y que tanto aparece una regla de asociación mediante el soporte, así como se vio que la regla de asociación del las casas vendidas de 2 niveles y la casa intermediaria de 2 niveles que aparecía en un 20% de los casos.

5.1 Referencias

Burnett, B. and Burnett, K., 2013. 100-year-old brick foundation may need to be replaced. [online] SFGATE. Available at: https://www.sfgate.com/homeandgarden/sweatequity/article/100-year-old-brick-foundation-may-need-to-be-4958112.php.

INTERNACHI, 2022. Gravity Furnace Inspection. [online] Nachi.org. Available at: https://www.nachi.org/gravity-furnace-inspection.htm.

Created in Deepnote