Lab 3 Kasus dan Korelasi

1	# import tools
	import pandas as pd
	import seaborn as sns
	import matplotlib.pyplot as plt
	# customisasi style plotnya
	plt.style.use('bmh')

2	# import data rumah
	df = pd.read_csv('datarumah.csv')
	df.head()

3	# cek info
	df.info()

4	# amati penyebaran harga rumah
	print(df['SalePrice'].describe())
	plt.figure(figsize=(9, 8))
	sns.distplot(df['SalePrice'], color='g', bins=100, hist_kws={'alpha': 0.4});
	# terlihat kebanyakan harga berada di 100000-200000 dan harga yang mungkin sala
	h karena lebih dari 500000

5	# Mengambil data angka yg tersebar untuk diolah
	# pertama mengecek tipe data dan mengambil hanya angka saja
	list(set(df.dtypes.tolist()))
	df_angka = df.select_dtypes(include = ['float64', 'int64'])
	df_angka.head()

6	# plotkan
	df_angka.hist(figsize=(16, 20), bins=50, xlabelsize=8, ylabelsize=8);
	# terlihat pola yang mirip dengan salePrice yaitu d 1stFlrSF, TotalBsmtSF, LotF
	rontage, GrLiveArea

7	# korelasi
	# mencari fitur/kolom yang berkorelasi kuat dengan salePrice menggunakan .corr(
	)
	# disimpan di golden_features_list
	df_angka_corr = df_angka.corr()['SalePrice'][:-1] # -1 karena yang paling berko
	relasi adalah SalePrice sendiri
	golden_features_list = df_angka_corr[abs(df_angka_corr) > 0.5].sort_values(asce
	nding=False)
	print("Ada {} yang berkorelasi kuat dengan SalePrice:\n{}".format(len(golden_fe
	atures_list), golden_features_list))

8	# sudah mendapatkan daftar yang paling berkorelasi, namun bisa juga karena outl iers # mengatasinya dengan mengamati masing2 fitur/kolom # hapus dan bandingkan kembali for i in range(0, len(df_angka.columns), 5):

	sns.pairplot(data=df_angka, x_vars=df_angka.columns[i:i+5], y_vars=['SalePrice']) # terlihat beberapa hubungan, kebanyakan berhubungan linear dengan salePrice # dan terlihat data yang banyak di x=0 berarti ada beberapa fitur rumah yang ti dak ada # hilangkan 0 kemudian cari korelasinya lg

9	# menampilkan kembali skor korelasinya
	# import operator
	import operator
	
	individual_features_df = [] # membuat array
	# looping menambah item array
	for i in range(0, len(df_angka.columns) - 1): # -1 karena kolom terakhir adalah SalePrice
	tmpDf = df_angka[[df_angka.columns[i], 'SalePrice']]
	tmpDf = tmpDf[tmpDf[df_angka.columns[i]] != 0]
	individual_features_df.append(tmpDf)
	
	# menampilkan nama fitur beserta korelasi
	all_correlations = {feature.columns[0]: feature.corr()['SalePrice'][0] for feat ure in individual_features_df}
	all_correlations = sorted(all_correlations.items(), key=operator.itemgetter(1))
	for (key, value) in all_correlations:
	print("{:>15}: {:>15}".format(key, value))
	# masukan list ke golden_features_list
	golden_features_list = [key for key, value in all_correlations if abs(value) >= 0.5]
	print("Ada {} yang berkorelasi kuat dengan SalePrice:\n{}".format(len(golden_fe atures_list), golden_features_list))

10	# hubungan feature to feature
	# plotting menggunakan sns.heatmap
	corr = df_angka.drop('SalePrice', axis=1).corr() # We already examined SalePric e correlations
	plt.figure(figsize=(12, 10))
	
	sns.heatmap(corr[(corr >= 0.5) | (corr <= -0.4)],
	cmap='viridis', vmax=1.0, vmin=-1.0, linewidths=0.1,
	annot=True, annot_kws={"size": 8}, square=True);
	
	# banyak fitur yang berkorelasi, seperti YearBuid/GarageYrBlt namun hanya
	# mengindikasikan inflasi harga tahunan
	# seperti 1stFlrSF/TotalBsmtSF, ini normal karena semakin luas lt1 maka makin l uas basementnya
	# hubungan yang menarik HalfBath/2ndFlrSF mengindikasikan orang menganggap pent ing
	# mempunyai kamar mandi di lt2 supaya tidak buru2 turun saat butuh

11	# Hubungan Q-Q
	`# hubungan Quantitative to Quantitative`
	`quantitative_features_list = ['LotFrontage', 'LotArea', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'TotalBsmtSF', '1stFlrSF',`
	`'2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',`
	`'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageCars',`
	`'GarageArea', 'WoodDeckSF', 'OpenPorchSF',`
	`'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'SalePrice']`
	`df_quantitative_values = df[quantitative_features_list]`
	`df_quantitative_values.head()`

12	`# memilih fitur-fitur yang akan digunakan berdasar golden list`
	`features_to_analyse = [x for x in quantitative_features_list if x in golden_features_list]`
	`features_to_analyse.append('SalePrice')`
	`features_to_analyse`

13	`# amati penyebarannya`
	`fig, ax = plt.subplots(round(len(features_to_analyse) / 3), 3, figsize = (18, 12))`
	
	`for i, ax in enumerate(fig.axes):`
	`if i < len(features_to_analyse) - 1:`
	`sns.regplot(x=features_to_analyse[i],y='SalePrice', data=df[features_to_analyse], ax=ax)`
	`# bisa dilihat TotalBsmtSF, 1stFlrSF, GrLivArea memiliki penyebaran yang besar namun tidak bisa disimpulkan informasi apa yang bisa didapat`

14	`# hubungan C-Q (Categorical to Quantitative)`
	`# quantitative_features_list[:-1] untuk salePricenya`
	`categorical_features = [a for a in quantitative_features_list[:-1] + df.columns.tolist() if (a not in quantitative_features_list[:-1]) or (a not in df.columns.tolist())]`
	`df_categ = df[categorical_features]`
	`df_categ.head()`

15	`# jangan lupa fitur yang bukan angka/non-numerical`
	`df_not_num = df_categ.select_dtypes(include = ['O'])`
	`print('Ada {} fitur non numerical yaitu:\n{}'`
	`.format(len(df_not_num.columns), df_not_num.columns.tolist()))`

16	`# diplotkan dengan boxplot antara SaleCondition dan SalePrice`
	`plt.figure(figsize = (12, 6))`
	`ax = sns.boxplot(x='SaleCondition', y='SalePrice', data=df_categ)`
	`plt.setp(ax.artists, alpha=.5, linewidth=2, edgecolor="k")`

	plt.xticks(rotation=45)

17	# diplotkan dengan boxplot antara BsmtExposure dan SalePrice
	plt.figure(figsize = (10, 6))
	ax = sns.boxplot(x='BsmtExposure', y='SalePrice', data=df_categ)
	plt.setp(ax.artists, alpha=.5, linewidth=2, edgecolor="k")
	plt.xticks(rotation=45)

18	# lihat keseluruhan
	fig, axes = plt.subplots(round(len(df_not_num.columns) / 3), 3, figsize=(12, 30
	))
	for i, ax in enumerate(fig.axes):
	    if i < len(df_not_num.columns):
	        ax.set_xticklabels(ax.xaxis.get_majorticklabels(), rotation=45)
	        sns.countplot(x=df_not_num.columns[i], alpha=0.7, data=df_not_num, ax=a
	x)
	
	fig.tight_layout()

19	