Exploratory Data Analysis for Machine Learning

1) Brief description of the data set and a summary of its attributes:

The Ames Housing dataset was compiled by Dean De Cock for use in data science education.
http://jse.amstat.org/v19n3/decock.pdf (http://jse.amstat.org/v19n3/decock.pdf)
Altogether the data set is made of 2920 rows and 81 columns.
The data set was already splited into Train and Test sets since it comes directly from Kaggle Competition.
Both the Test and Train data sets consists of 1460 rows each.
For the needs of Kaggle competition the Test split was stripped from "SalePrice" column.
Hence to above the performance of any model build on Ames Housing dataset is tested on a hold out split stored on a Kaggle server.
The data set exhibits 19 variables with NULL values needed to be dealt with.

More details on GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

2) Initial plan for data exploration:

My initial plan for data exploration looks as follows:

- Reading the documentation about the Ames Housing dataset - getting an initial understanding of inputs.
- Printing the informatation about the Train dataset:
  - Checking number of rows.
  - Checking data types of variables.
  - Getting first inside about the number of missing records.
- Counting the number of missing values among all columns:
  - Looking into the fraction of missing entries within each one of variables.
- Looking into summary statistics of both numeric variables and characteristics - getting an initial view about distributions of variables.

3) Actions taken for data cleaning and feature engineering:

The first challenge of the Ames Housing data set was to deal with missing values.
The challenge has been tackled with a use of methods provided by scikit-learn imputation algorithms.
Variables with imputed values have been visualised before and after imputation to assess the meaningfullness of chosen methods.
For categorical variables the SimpleImputer have been chosen as adequate method of data imputation.
Appart from mentioned SimpleImputer in case of numerical values the IterativeImputer and KNNImputer were applied to compare distributions after and before imputation.
As always before jumping straight forward into development of a model polynomial features and interactions between variables were derived from original numeric columns to fullfil a need for feature engineering.
Above step was continued by computation of multiple variables expressing the sizes of deviations within each one of characteristics groups.
The deviation was defined as a substraction of a group mean from each one of indiviudual record being a member of that group, divided by a standard deviation of that group.
In case of variables showing skewness larger than 0.75 the np.Log1p transformation was applied to centre the distributions of variables towards it's modes.
The transformation returns the natural logarithm of one plus the input.
Numeric variables were scaled with a use of Robust Scaler from scikit-learn.
Pandas get_dummies method was used to derive dummy variables from categorical variables.

More details on GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

4) Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner:

- Target "SalesPrice" is not normally distributed. It suffers from presence of outliers:
  - variable is positively skewed
  - the mean is 180 921
  - maximum is 755 000
- There were 3 numeric variables with NULL values for which "distance" weights were used as most practical imputation strategy
- There were 16 characteristics with NULL values for which "most frequent" strategy of imputation proved to be the most useful one
- Top 5 correlated variables with SalePrice before any tranformations applied are:
  - OverallQual - Rates the overall material and finish of the house
  - GrLivArea - Above grade (ground) living area square feet
  - GarageCars - Size of garage in car capacity
  - GarageArea - GarageArea
  - TotalBsmtSF - Total square feet of basement area
- Top correlated variable "OverallQual" is a quality ranking of houses starting from 1 to 10
- Deviations were calculated to include discriminatory power of characteristics groupings

More details on GitHub (https://github.com/KonuTech/house-prices-advanced-regression-techniques/blob/main/Exploratory%20Data%20Analysis%20for%20Machine%20Learning.ipynb)

5) Formulating at least 3 hypothesis about this data:

- Distribution of numeric variable "LotFrontage" after imputation is equal to distribution of that variable before imputation
- Distribution of characteristics "Alley" after imputation is equal to distribution of that variable before imputation
- Residuals of train target are normally distributed

6) Conducting a formal significance test for one of the hypotheses and discuss the results

H0: Distribution of Residuals for "SalePrice" from Training data set is normal.
H1: Distribution of Residuals for "SalePrice" from Training data set is not normal.

In [188]:

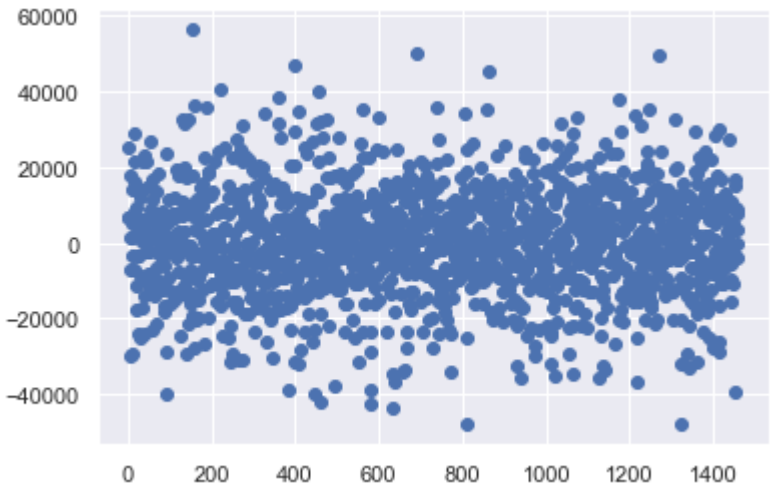
```
residuals = y_train - predicted_gbr_train
```

In [189]:

```
ax = plt.axes()
ax.scatter(y_train.index, residuals)
```

Out[189]:

```
<matplotlib.collections.PathCollection at 0x2c28a2c0908>
```

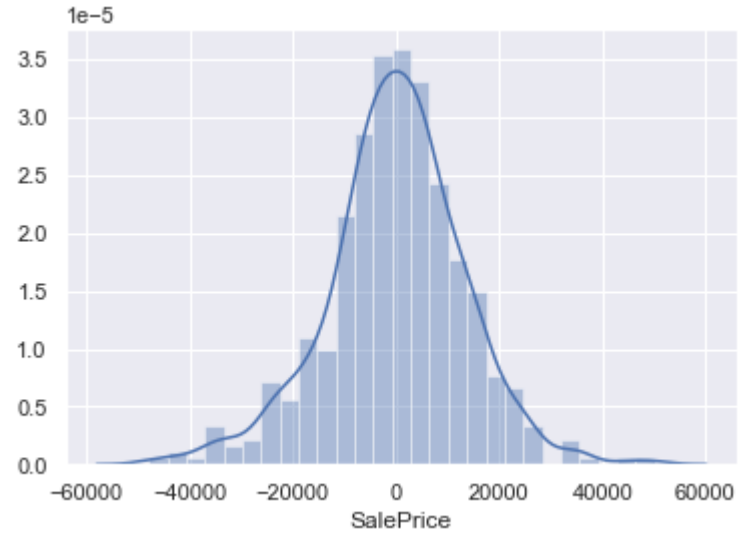


In [190]:

```
sns.distplot(residuals[500:1000])
```

Out[190]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c28cec5088>
```



In [191]:

```
print("Skewness: %f" % residuals.skew())
print("Kurtosis: %f" % residuals.kurt())
```

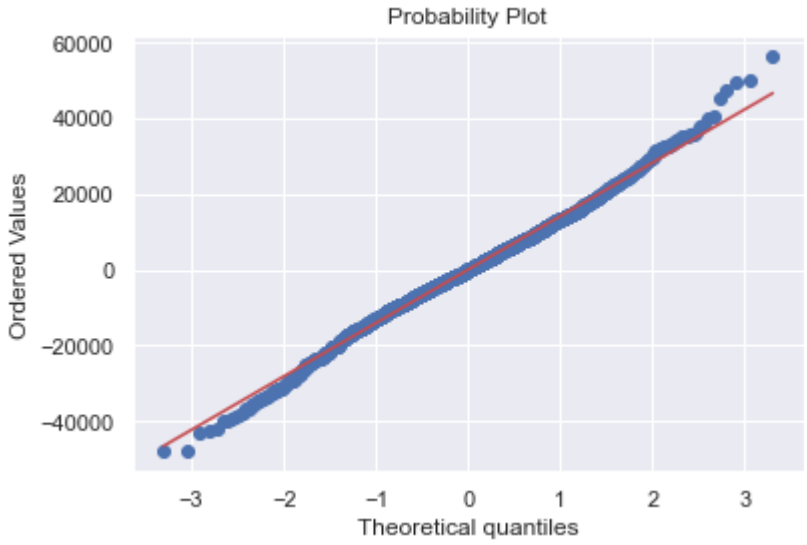
```
Skewness: 0.003565
Kurtosis: 0.655485
```

In [192]:

```
stats.probplot(residuals, plot=plt)
```

Out[192]:

```
((array([-3.30513952, -3.04793228, -2.90489705, ...,  2.90489705,
          3.04793228,  3.30513952]),
  array([-48036.82662753, -47906.24354124, -43327.90511743, ...,
         49502.11397364,  50017.52854807,  56546.22391532])),
 (14101.746374388877, -2.2691041910926567e-12, 0.9973863812743627))
```



In [193]:

```
from scipy.stats import normaltest
statistics, pvalue = normaltest(residuals)
print("statistics=%.3f, pvalue=%.3f\n" % (statistics, pvalue))
if p > 0.05:
    print("Probably Gaussian distribution")
else:
    print("Probably not Gaussian distribution")
```

```
statistics=16.010, pvalue=0.000

Probably not Gaussian distribution
```

In [194]:

```
from scipy.stats import kstest
statistics, pvalue = kstest(residuals, "norm")
print("statistic=%.3f, pvalue=%.3f\n" % (statistics, pvalue))
if pvalue > 0.05:
    print("Probably Gaussian distribution")
else:
    print("Probably not Gaussian distribution")
```

```
statistic=0.501, pvalue=0.000

Probably not Gaussian distribution
```

Basing on data providedI reject H0

7) Suggestions for next steps in analyzing this data

Numerous variables computed during Feature Engineering process have been exhibiting strong correlation between each other.
Following Feature Selection solutions have been applied to deal with an issue of multicolinarity:

- Variance Threshold
- Univariate feature selection
- Variance Inflation Factor

8) A paragraph that summarizes the quality of this data set and a request for additional data if needed

Missing values of Ames Housing data are providing additional challenge into process of data sanitazation.
Moreover, since Training data set is rather little in size than the process of fitting the model on such set most likely will result in a "data leakage".
In compare to characteristics of Train set the cardinality of characteristcs from Test data set can simply prove to be diffrent.
Hence deriving predictions will not be possible without recalibration of a model.
Since characteristics from business perspective are often provifing strong interpretability it is strongly adviced to increase sample of provided data set.
Better models are providing better and smarter decisions. Smarter decisions are either providing larger pofits and/or larger savings.

In [ ]:

```
```

In [ ]:

```
```