ММО РК1

Павловская А.А. ИУ5-22М

Вариант №12
Задача №12:
Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "логарифм - np.log(X)".
Задача №32:
Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод обертывания (wrapper method), обратный алгоритм (sequential backward selection).
Доп.задание: для произвольной колонки данных построить гистограмму.

Ввод [6]:
```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

Датасет London Weather Data https://www.kaggle.com/datasets/emmanuelfwerr/london-weather-data (https://www.kaggle.com/datasets/emmanuelfwerr/london-weather-data)

1. date - записанная дата измерения - (int)
2. cloud_cover - измерение облачного покрова в октантах - (float)
3. sunshine - измерение солнечного света в часах (hrs) - (float)
4. global_radiation - измерение интенсивности излучения в ваттах на квадратный метр (W/m2) - (float)
5. max_temp - максимальная зарегистрированная температура в градусах Цельсия (°C) - (float)
6. mean_temp - средняя температура в градусах Цельсия (°C) - (float)
7. min_temp – минимальная зарегистрированная температура в градусах Цельсия (°C) - (float)
8. precipitation - измерение осадков в миллиметрах (mm) - (float)
9. pressure - измерение давления в паскалях (Pa) - (float)
10. snow_depth - измерение глубины снега в сантиметрах (cm) - (float)

Ввод [7]:
```
#Загрузка данных
data = pd.read_csv(r"D:\Py\MAD\weather_nulls.csv")
```

Ввод [8]:
```
cols_filter = ['cloud_cover', 'sunshine','global_radiation','min_temp','max_temp','mean_te
data = data[cols_filter]
data.head()
```

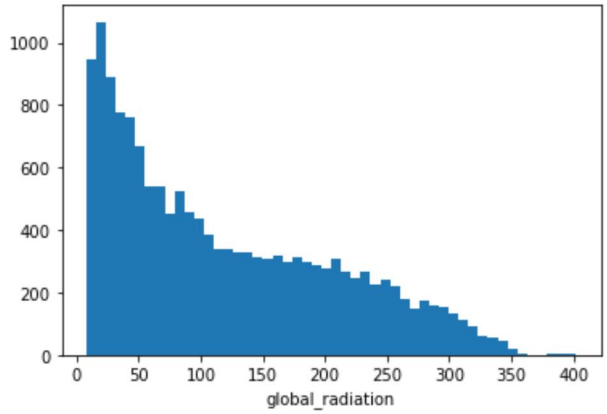
Out[8]:

	cloud_cover	sunshine	global_radiation	min_temp	max_temp	mean_temp	pressure	snow_depth
0	2.0	7.0	52.0	-7.5	2.3	-4.1	101900.0	9.0
1	6.0	1.7	27.0	-7.5	1.6	-2.6	102530.0	8.0
2	5.0	0.0	13.0	-7.2	1.3	-2.8	102050.0	4.0
3	8.0	0.0	13.0	-6.5	-0.3	-2.6	100840.0	2.0
4	6.0	2.0	29.0	-1.4	5.6	-0.8	102250.0	1.0

Ввод [9]:
```
data.shape
```

Out[9]: (15341, 8)

Ввод [10]:
```
#гистограмма для колонки "global_radiation"
plt.hist(data['global_radiation'], 50)
plt.xlabel('global_radiation')
plt.show()
```

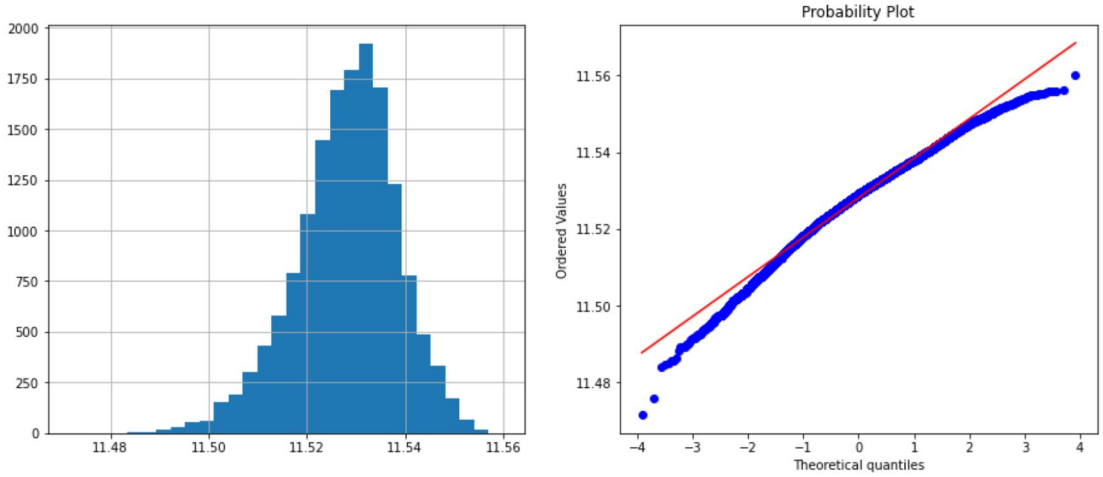


Задача №12

Ввод [11]:
```
import scipy.stats as stats
```

Ввод [12]:
```
def diagnostic_plots(df, variable):
    plt.figure(figsize=(15,6))
    # гистограмма
    plt.subplot(1, 2, 1)
    df[variable].hist(bins=30)
    ## Q-Q plot
    plt.subplot(1, 2, 2)
    stats.probplot(df[variable], dist="norm", plot=plt)
    plt.show()
```

Ввод [13]:
```
#нормализация признака pressure с использованием функции логарифм - np.log(X)
data1 = pd.DataFrame()
data1['pressure_log'] = np.log(data['pressure'])
diagnostic_plots(data1, 'pressure_log')
```



Задача №32

Ввод [50]:
```
# DataFrame не содержащий целевой признак
X_ALL = data.drop('snow_depth', axis=1)
```

Ввод [51]:
```
# целевой признак - snow_depth
y = data['snow_depth']
```

Отбор признаков методом обертывания (wrapper method), обратным алгоритмом (sequential backward selection)

Ввод [18]:
```
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
```

Ввод [23]:
```
from sklearn.neighbors import KNeighborsClassifier
```

Ввод [60]:
```
knn = KNeighborsClassifier(n_neighbors=4)

sfs1 = SFS(knn,
           k_features=3,
           forward=False,
           floating=False,
           verbose=2,
           scoring='accuracy',
           cv=0)
```

Ввод [61]:
```
sfs1 = sfs1.fit(X_ALL, y)
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.1s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   7 out of   7 | elapsed:    7.7s finished

[2023-04-02 18:58:59] Features: 6/3 -- score: 0.9912652369467441[Parallel(n_jobs=1)]: Usi
ng backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.1s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   6 out of   6 | elapsed:    7.2s finished

[2023-04-02 18:59:06] Features: 5/3 -- score: 0.9914607913434587[Parallel(n_jobs=1)]: Usi
ng backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.0s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   5 out of   5 | elapsed:    5.4s finished

[2023-04-02 18:59:12] Features: 4/3 -- score: 0.9915259761423636[Parallel(n_jobs=1)]: Usi
ng backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:    1.1s remaining:    0.0s
[Parallel(n_jobs=1)]: Done   4 out of   4 | elapsed:    4.6s finished

[2023-04-02 18:59:17] Features: 3/3 -- score: 0.9915911609412685
```

Ввод [63]:
```
sfs1.subsets_
```

Out[63]:
```
{7: {'feature_idx': (0, 1, 2, 3, 4, 5, 6),
  'cv_scores': array([0.99087413]),
  'avg_score': 0.9908741281533147,
  'feature_names': ('cloud_cover',
   'sunshine',
   'global_radiation',
   'min_temp',
   'max_temp',
   'mean_temp',
   'pressure')},
 6: {'feature_idx': (0, 1, 2, 3, 4, 5),
  'cv_scores': array([0.99126524]),
  'avg_score': 0.9912652369467441,
  'feature_names': ('cloud_cover',
   'sunshine',
   'global_radiation',
   'min_temp',
   'max_temp',
   'mean_temp')},
 5: {'feature_idx': (0, 1, 2, 3, 5),
  'cv_scores': array([0.99146079]),
  'avg_score': 0.9914607913434587,
  'feature_names': ('cloud_cover',
   'sunshine',
   'global_radiation',
   'min_temp',
   'mean_temp')},
 4: {'feature_idx': (0, 1, 2, 3),
  'cv_scores': array([0.99152598]),
  'avg_score': 0.9915259761423636,
  'feature_names': ('cloud_cover',
   'sunshine',
   'global_radiation',
   'min_temp')},
 3: {'feature_idx': (0, 1, 3),
  'cv_scores': array([0.99159116]),
  'avg_score': 0.9915911609412685,
  'feature_names': ('cloud_cover', 'sunshine', 'min_temp')}}
```

Таким образом, были отобраны признаки 'cloud_cover', 'sunshine' и 'min_temp'