Рабцевич К.Р. ИУ5-21М

Вариант 13

Каждая задача предполагает использование набора данных.

Набор данных выбирается Вами произвольно с учетом следующих условий:

- Вы можете использовать один набор данных для решения всех задач, или решать каждую задачу на своем наборе данных.
- Набор данных должен отличаться от набора данных, который использовался в лекции для решения рассматриваемой задачи.
- Вы можете выбрать произвольный набор данных (например тот, который Вы использовали в лабораторных работах) или создать собственный набор данных (что актуально для некоторых задач, например, для задач удаления псевдоконстантных или повторяющихся признаков).
- Выбранный или созданный Вами набор данных должен удовлетворять условиям поставленной задачи. Например, если решается задача устранения пропусков, то набор данных должен содержать пропуски.

Номер задачи №1 - 13

Номер задачи №2 - 33

Задача №13.

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "обратная зависимость - 1 / X".

Задача №33.

Для набора данных проведите процедуру отбора признаков (feature selection).
Используйте метод обертывания (wrapper method), алгоритм полного перебора (exhaustive feature selection).

Дополнительные требования:

Для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Загрузка и первичный анализ данных

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [2]:
```
data = pd.read_csv('bike-hour.csv', sep=",")
```

In [3]:
```
data.head(5)
```

Out[3]:

	instant	dteday	season	mnth	hr	holiday	weekday	workingday	weathersit	temp	atemp	hu
0	1	01-01-2011	1	1	0	0	6	0	1	0.24	0.2879	0.
1	2	01-01-2011	1	1	1	0	6	0	1	0.22	0.2727	0.
2	3	01-01-2011	1	1	2	0	6	0	1	0.22	0.2727	0.
3	4	01-01-2011	1	1	3	0	6	0	1	0.24	0.2879	0.
4	5	01-01-2011	1	1	4	0	6	0	1	0.24	0.2879	0.

Датасет

Информация об атрибутах:

- instant: индекс записи
- dteday: дата
- season: Сезон (1: зима, 2: весна, 3: лето, 4: осень)
- mnth: месяц (от 1 до 12)
- hour: час (от 0 до 23)
- holiday: выходной или нет
- weekday: день недели
- workingday: если день не является ни выходным, ни праздничным - 1, в противном случае - 0.
- weathersit:

  1: Ясно, Небольшая облачность, Небольшая облачность,

  2: Туман + Облачно, Туман + Разбитые облака, Туман + Несколько облаков, Туман

  3: слабый снег, легкий дождь + гроза + рассеянные облака, легкий дождь + рассеянные облака

  4: сильный дождь + ледяные поддоны + гроза + туман, снег + туман
- temp: нормализованная температура в градусах Цельсия

- atemp: нормализованная температура ощущения в градусах Цельсия
- hum: нормализованная влажность
- windspeed: нормализованная скорость ветра
- casul: количество случайных прохожих
- cnt: общее количество взятых напрокат велосипедов

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8645 entries, 0 to 8644
Data columns (total 15 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   instant     8645 non-null   int64  
 1   dteday      8645 non-null   object 
 2   season      8645 non-null   int64  
 3   mnth        8645 non-null   int64  
 4   hr          8645 non-null   int64  
 5   holiday     8645 non-null   int64  
 6   weekday     8645 non-null   int64  
 7   workingday  8645 non-null   int64  
 8   weathersit  8645 non-null   int64  
 9   temp        8645 non-null   float64
 10  atemp       8645 non-null   float64
 11  hum         8645 non-null   float64
 12  windspeed   8645 non-null   float64
 13  casual      8645 non-null   int64  
 14  cnt         8645 non-null   int64  
dtypes: float64(4), int64(10), object(1)
memory usage: 1013.2+ KB
```

In [5]:
```
data.describe()
```

Out[5]:

	instant	season	mnth	hr	holiday	weekday	workingday	
count	8645.000000	8645.000000	8645.000000	8645.000000	8645.000000	8645.000000	8645.000000	8
mean	4323.000000	2.513592	6.573973	11.573626	0.027646	3.012724	0.683748	
std	2495.740872	1.105477	3.428147	6.907822	0.163966	2.006370	0.465040	
min	1.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	
25%	2162.000000	2.000000	4.000000	6.000000	0.000000	1.000000	0.000000	
50%	4323.000000	3.000000	7.000000	12.000000	0.000000	3.000000	1.000000	
75%	6484.000000	3.000000	10.000000	18.000000	0.000000	5.000000	1.000000	
max	8645.000000	4.000000	12.000000	23.000000	1.000000	6.000000	1.000000	

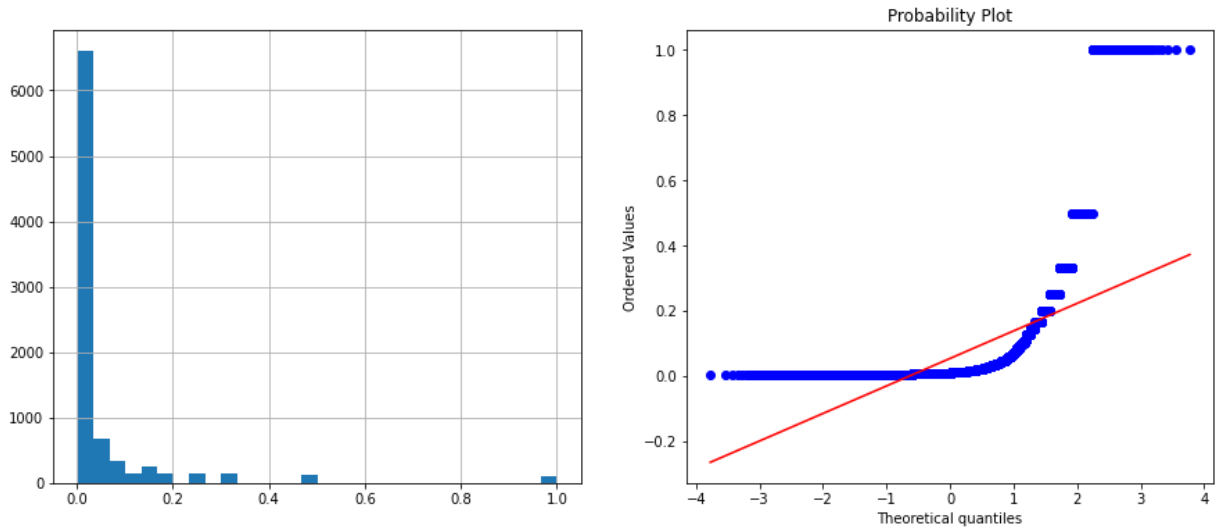
Задача №13.

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием функции "обратная зависимость - 1 / X".

Нормализацию будем проводить для поля cnt

In [6]:
```
import matplotlib.pyplot as plt
import scipy.stats as stats
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()
```

In [7]:
```
data['cnt_reciprocal'] = 1 / (data['cnt'])
diagnostic_plots(data, 'cnt_reciprocal')
```



In [8]:
```
c = pd.DataFrame({'cnt':data['cnt'], 'cnt_reciprocal':data['cnt_reciprocal']})
c
```

Out[8]:

	cnt	cnt_reciprocal
0	16	0.062500
1	40	0.025000
2	32	0.031250
3	13	0.076923
4	1	1.000000
...	...	...
8640	92	0.010870
8641	71	0.014085
8642	52	0.019231
8643	38	0.026316

	cnt	cnt_reciprocal
8644	31	0.032258

8645 rows × 2 columns

Задача №33.

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте метод обертывания (wrapper method), алгоритм полного перебора (exhaustive feature selection).

In [9]:
```
data = pd.read_csv('WineQT.csv', sep=",")
```

In [10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1143 entries, 0 to 1142
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1143 non-null   float64
 1   volatile acidity      1143 non-null   float64
 2   citric acid           1143 non-null   float64
 3   residual sugar        1143 non-null   float64
 4   chlorides             1143 non-null   float64
 5   free sulfur dioxide   1143 non-null   float64
 6   total sulfur dioxide  1143 non-null   float64
 7   density               1143 non-null   float64
 8   pH                    1143 non-null   float64
 9   sulphates             1143 non-null   float64
 10  alcohol               1143 non-null   float64
 11  quality               1143 non-null   int64  
 12  Id                    1143 non-null   int64  
dtypes: float64(11), int64(2)
memory usage: 116.2 KB
```

In [11]:
```
from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
from sklearn.neighbors import KNeighborsClassifier
from sklearn.datasets import load_iris
knn = KNeighborsClassifier(n_neighbors=3)
```

In [12]:
```
data_x = data.drop('quality', 1).values
data_y = data["quality"]
```

```
/tmp/ipykernel_3101147/3300780049.py:1: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
  data_x = data.drop('quality', 1).values
```

In [14]:
```
efs = EFS(knn,
          min_features=1,
          max_features=3,
          scoring='accuracy',
          print_progress=True,
```

```
            cv=5)

efs = efs.fit(data_x, data_y)

print('Best accuracy score: %.2f' % efs.best_score_)
print('Best subset (indices):', efs.best_idx_)
print('Best subset (corresponding names):', efs.best_feature_names_)
```

```
Features: 298/298
Best accuracy score: 0.52
Best subset (indices): (2, 9, 10)
Best subset (corresponding names): ('2', '9', '10')
```

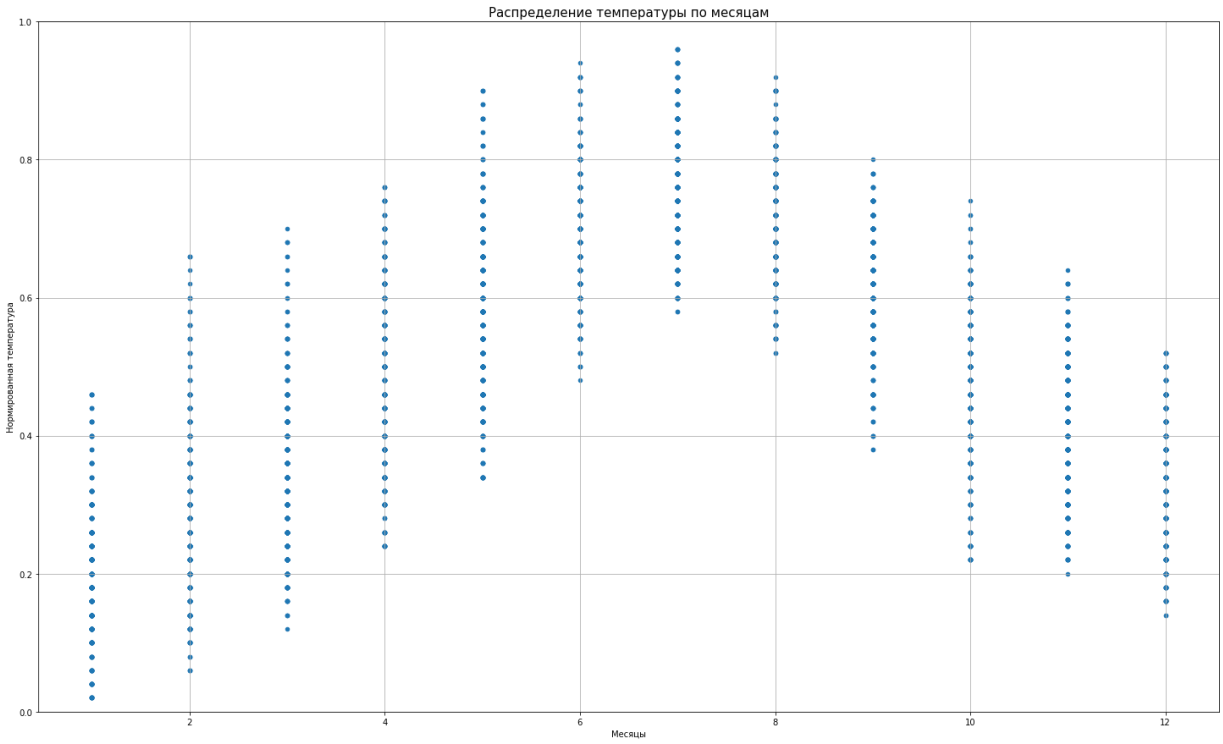
Дополнительное задание

Для пары произвольных колонок данных построить график "Диаграмма рассеяния".

Построим диаграмму рассеяния, демонстрирующую зависимость температуры от месяца года

In [15]:
```
data = pd.read_csv('bike-hour.csv', sep=",")
```

In [16]:
```
data.plot(x='mnth', y='temp', kind='scatter',  figsize=(25, 15)) ;
plt.title(f'Распределение температуры по месяцам', fontsize=15);
plt.ylim(0,1);
plt.ylabel('Нормированная температура');
plt.xlabel('Месяцы');
plt.grid(True);
```



In [ ]:
```

```