НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ «МИФИ»

Отчет по лабораторной работе №2

по курсу: «Специальные технологии баз данных»

Выполнил: *студент группы С20-702* *Нуритдинходжаева А.А.*

(подпись) (Фамилия И.О.)

Проверил: *Манаенкова Т.А.*

(оценка) (подпись) (Фамилия И.О.)

Москва 2023 г

Условие задания

Вариант 4

Зарегистрируйтесь на сайте https://www.kaggle.com/datasets и загрузите с него набор статистических данных, посвящённый продажам домов
https://www.kaggle.com/harlfoxem/housesalesprediction

Блок 1

1. На основе загруженного CSV-файла создайте Pandas DataFrame, гарантировав правильные типы данных, переменных (например, правильную загрузку дат). Назовите переменные также, как названы колонки в исходном наборе данных.

2. Измените полученный в задании 1 Pandas DataFrame, оставив из исходного Pandas DataFrame переменные date price yr_built yr_renovated sqft_living condition. Кроме того, добавьте в Pandas DataFrame новую переменную real_year, собрав её как максимальное значение из значений переменных yr_built yr_renovated.

3. Измените полученный в задании 2 Pandas DataFrame, отсортировав его в следующем порядке real_year, sqft_living condition.

4. Выполните простейший количественный анализ по переменной real_year Pandas DataFrame, полученного в задании 3, отсортировав при этом результаты в порядке возрастания количества продаж среди наблюдений. Сохраните результаты в новый Pandas DataFrame.

5. Сформируйте новый Pandas DataFrame, в котором останутся только 4 значения real_year: два наиболее часто встречающихся и два наименее часто встречающихся среди наблюдений исходного массива.

6. Сформируйте новый Pandas DataFrame, исключив из Pandas DataFrame, полученного в задании 3 все наблюдения, относящиеся к 4-м годам, полученным в задании 5.

7. Выполните простейший количественный анализ по переменной condition Pandas DataFrame, полученного в задании 6, отсортировав при этом результаты в порядке убывания количества появлений данного состояния среди наблюдений. Сохраните результаты в новый Pandas DataFrame.

8. Выполните нормализацию Pandas DataFrame, полученного в пункте 3, по переменным real_year и condition.

9. Постройте гистограммы и кривые распределения для переменных real_year и condition. Сравните их кривые распределения графически с кривыми нормального распределения. Сделайте выводы.

10. Постройте график линейной регрессии для переменных real_year и condition. Сделайте выводы о их взаимосвязи.

Решение

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

#1
df = pd.read_csv("kc_house_data.csv", sep = ",", parse_dates = ["date"], encoding = "utf8")
print(df)

#2
f1 = df[["date", "price", "yr_built", "yr_renovated", "sqft_living", "condition"]]
f1['real_year'] = f1[['yr_built', 'yr_renovated']].max(axis = 1)
print(f1)

#3
f1_sorted = f1.sort_values(by = ["real_year", "sqft_living", "condition"])
print(f1_sorted)

#4
real_year_counts = f1_sorted['real_year'].value_counts().sort_index() #считаем кол-во продаж для
каждого года
print(real_year_counts)

analys_res = pd.DataFrame({'real_year' : real_year_counts.index, 'sales_count' : real_year_counts.values})
#создаем новый DataFrame

analys_res = analys_res.sort_values(by = 'sales_count') #сортируем
print(analys_res)

#5
value_counts = f1['real_year'].value_counts()
most_frequent_values = value_counts.nlargest(2).index
#print(most_frequent_values)

least_frequent_values = value_counts.nsmallest(2).index
#print(least_frequent_values)

max_min_frequent_values = pd.DataFrame({'real_year' :
most_frequent_values.union(least_frequent_values)})
print(max_min_frequent_values)

#6
values_to_exclude = [1934, 1935, 2005, 2014]

f1_exclude = f1[~f1['real_year'].isin(values_to_exclude)]
print(f1_exclude)

#7
condition_counts = f1_exclude['condition'].value_counts().sort_index()
#print(condition_counts)

analys_res_condition = pd.DataFrame({'condition' : condition_counts.index, 'state_count' :
condition_counts.values}) #создаем новый DataFrame

analys_res_condition = analys_res_condition.sort_values(by = 'state_count', ascending = False) #сортируем
print(analys_res_condition)
```

```
#8
scaler = MinMaxScaler()

f1_sorted_norm = f1_sorted.copy()
f1_sorted_norm[['real_year', 'condition']] = scaler.fit_transform(f1_sorted[['real_year', 'condition']])
print(f1_sorted_norm)

#9
sub_df_9 = f1_sorted[['real_year','condition']]
sub_df_9.hist()

import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import norm

f1_sorted_norm['real_year'].plot.kde()
f1_sorted_norm['condition'].plot.kde()
x =np.linspace(-3,3,1000)
data = norm.pdf(x)
plt.plot(x,data)
plt.show

#10
from sklearn.linear_model import LinearRegression
model = LinearRegression()
#model.fit(df_sorted_norm['real_year'].values, df_sorted_norm['condition'].values)
x = f1_sorted_norm['real_year'].to_numpy()
y = f1_sorted_norm['condition'].to_numpy()
model.fit(x.reshape(-1,1),y.reshape(-1,1))
model.coef_, model.intercept_
plt.plot(x,y,'o')
l = np.linspace(-3,3,10)
res = model.coef_[0] * l + model.intercept_
plt.plot(l, res)
```