NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme 'HSE University and University of London Double Degree Programme in Data Science and Business Analytics'

UDC 004

Research Project Report (Final)

on the topic "Determining impactful factors in the pricing of the Yandex Direct bids"

Fulfilled by the Student:

group #БПАД204  Borisov Artem Nikolayevich

Signature — Surname, First name, Patronymic, if any

28.09.2022

Date

Checked by the Project Supervisor:

Rudakov K. A.

Surname, First name, Patronymic (if any), Academic title (if any)

Job

Visiting Scholar, Faculty of Computer Science, Big Data and Retrieval School

Place of Work (Company or HSE Department)

Date 28 september 2022 — 6 — 

Grade according to 10-point scale — Signature

Moscow 2022

Content:

Abstract:

In this paper I intend to research the main factors that determine the fluctuations of prices of keywords in Yandex Direct. After retrieving the data, I will apply the Mutual Information Score method to data from Yandex Direct Forecaster in order to see which parameters influence the price formation the most. Aside from that, I will extract historical data of advertising campaigns that I am currently running and see whether it can provide meaningful insights on those parameters. After executing the plan stated above, I have concluded that factor ‘Shows’ causes the greatest impact on bid price formation prior to the auction. On the other hand, the factor ‘Clicks’ influences the bids the most during the auction. In my work I have mostly referenced publicly available Yandex API documentation and referred to some sources on Machine Learning in order to ensure that I am applying the proper methodology for my tests.

Link to github: https://github.com/artem456borisov/Yandex-Direct-Factors

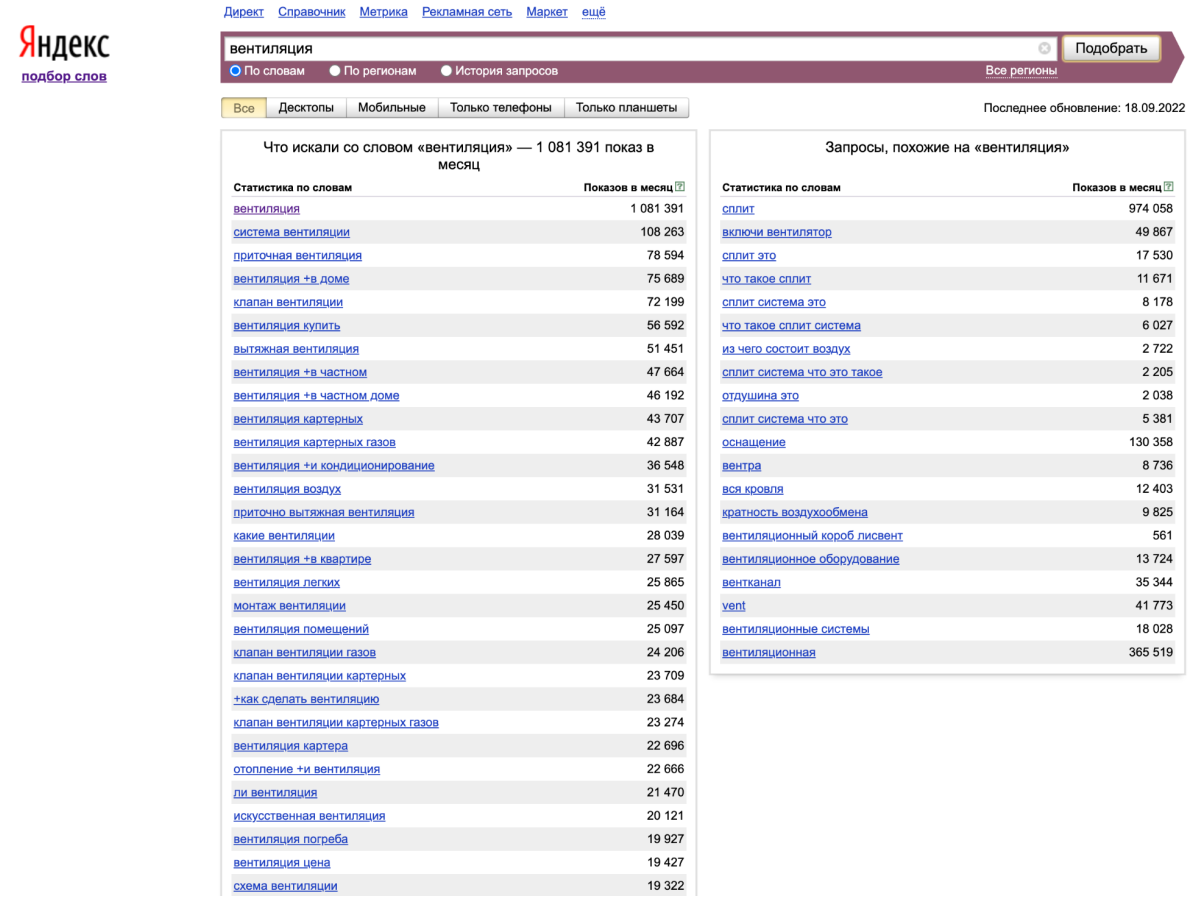
List of Keywords: Yandex, Yandex Direct, Yandex API, online marketing, advertisement, ML.

Basic terms and definitions:

Yandex Direct - A platform for locating and managing advertisement campaigns in Yandex browser.

MIS (Mutual Information Score) - a method to determine the extent of dependency between several variables.

Yandex Wordstat: A public service provided by Yandex Direct. It allows advertisers to see the frequency at which a certain phrase is searched in the Yandex browser. Aside from that, it also displays phrases that relate to the key phrase. The following screenshot displays the results for entering the phrase ‘вентиляция’ into WordStat’s search bar (Try it yourself: https://wordstat.yandex.ru/):



Яндекс
подбор слов

Директ Справочник Метрика Рекламная сеть Маркет ещё

вентиляция — Подобрать

По словам | По регионам | История запросов | Все регионы

Все | Десктопы | Мобильные | Только телефоны | Только планшеты

Последнее обновление: 18.09.2022

Что искали со словом «вентиляция» — 1 081 391 показ в месяц

Статистика по словам	Показов в месяц
вентиляция	1 081 391
система вентиляции	108 263
приточная вентиляция	78 594
вентиляция +в доме	75 689
клапан вентиляции	72 199
вентиляция купить	56 592
вытяжная вентиляция	51 451
вентиляция +в частном	47 664
вентиляция +в частном доме	46 192
вентиляция картерных	43 707
вентиляция картерных газов	42 887
вентиляция +и кондиционирование	36 548
вентиляция воздух	31 531
приточно вытяжная вентиляция	31 164
какие вентиляции	28 039
вентиляция +в квартире	27 597
вентиляция легких	25 865
монтаж вентиляции	25 450
вентиляция помещений	25 097
клапан вентиляции газов	24 206
клапан вентиляции картерных	23 709
+как сделать вентиляцию	23 684
клапан вентиляции картерных газов	23 274
вентиляция картера	22 696
отопление +и вентиляция	22 666
ли вентиляция	21 470
искусственная вентиляция	20 121
вентиляция погреба	19 927
вентиляция цена	19 427
схема вентиляции	19 322

Запросы, похожие на «вентиляция»

Статистика по словам	Показов в месяц
сплит	974 058
включи вентилятор	49 867
сплит это	17 530
что такое сплит	11 671
сплит система это	8 178
что такое сплит система	6 027
из чего состоит воздух	2 722
сплит система что это такое	2 205
отдушина это	2 038
сплит система что это	5 381
оснащение	130 358
вентра	8 736
вся кровля	12 403
кратность воздухообмена	9 825
вентиляционный короб лисвент	561
вентиляционное оборудование	13 724
вентканал	35 344
vent	41 773
вентиляционные системы	18 028
вентиляционная	365 519

Yandex Direct Budget Forecaster: A tool for approximating the monthly advertising budget of a certain group of phrases based on the amount of traffic you want to attract (the percentage of users who will see your advertisement). The following screenshot provides a forecast for this group of keywords, given that you want 62% of users to see your advertisement:

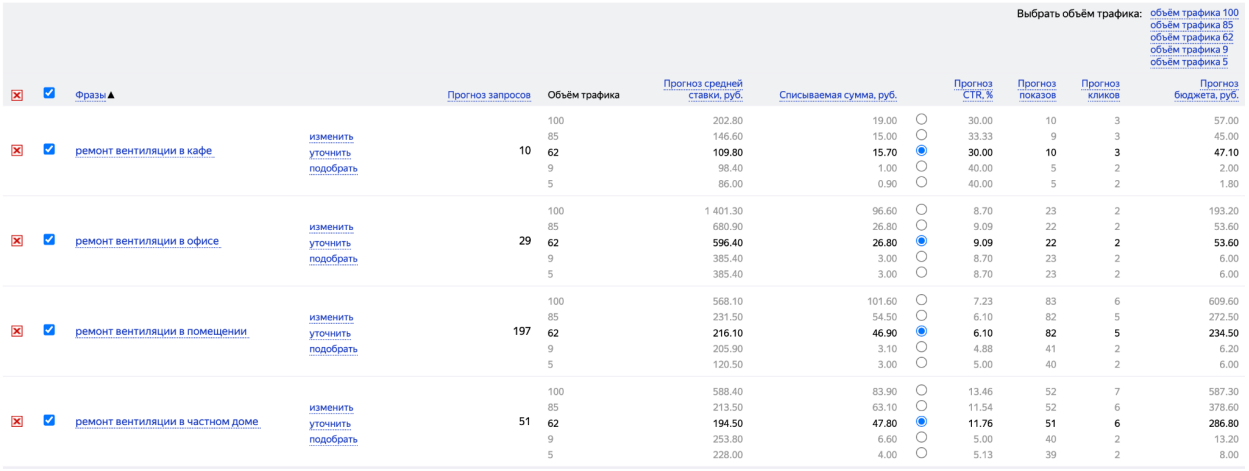
ремонт вентиляции бассейна

ремонт вентиляции в кафе

ремонт вентиляции в офисе

ремонт вентиляции в помещении

ремонт вентиляции в частном доме



Выбрать объём трафика: объём трафика 100, объём трафика 85, объём трафика 62, объём трафика 9, объём трафика 5

Фразы▲		Прогноз запросов	Объём трафика	Прогноз средней ставки, руб.	Списываемая сумма, руб.		Прогноз CTR, %	Прогноз показов	Прогноз кликов	Прогноз бюджета, руб.
ремонт вентиляции в кафе	изменить уточнить подобрать	10	100	202.80	19.00	○	30.00	10	3	57.00
			85	146.60	15.00	○	33.33	9	3	45.00
			62	109.80	15.70	●	30.00	10	3	47.10
			9	98.40	1.00	○	40.00	5	2	2.00
			5	86.00	0.90	○	40.00	5	2	1.80
ремонт вентиляции в офисе	изменить уточнить подобрать	29	100	1 401.30	96.60	○	8.70	23	2	193.20
			85	680.90	26.80	○	9.09	22	2	53.60
			62	596.40	26.80	●	9.09	22	2	53.60
			9	385.40	3.00	○	8.70	23	2	6.00
			5	385.40	3.00	○	8.70	23	2	6.00
ремонт вентиляции в помещении	изменить уточнить подобрать	197	100	568.10	101.60	○	7.23	83	6	609.60
			85	231.50	54.50	○	6.10	82	5	272.50
			62	216.10	46.90	●	6.10	82	5	234.50
			9	205.90	3.10	○	4.88	41	2	6.20
			5	120.50	3.00	○	5.00	40	2	6.00
ремонт вентиляции в частном доме	изменить уточнить подобрать	51	100	588.40	83.90	○	13.46	52	7	587.30
			85	213.50	63.10	○	11.54	52	6	378.60
			62	194.50	47.80	●	11.76	51	6	286.80
			9	253.80	6.60	○	5.00	40	2	13.20
			5	228.00	4.00	○	5.13	39	2	8.00

CTR (Click Through Rate) - the number of users that have clicked on the advertisement, placed under a certain keyword, divided by the total number of users who have seen the advertisement.

Shows - the number of times a certain advertisement was shown in the Yandex browser.
Clicks - the amount of users that have visited the advertised page, after seeing the advertisement.

Introduction:

Relevance of the work:

Advertisement campaign managers often face uncertainty when trying to forecast advertising budgets. Some use Yandex Direct Forecaster, which often significantly overestimates the budget, while others use their experience or intuition. Knowing which parameters affect the budget the most may not erase the ambiguity of forecasting, but it may provide advertisement managers a better budget planning tool.

Object and subject of research:

1) Yandex direct budget forecast data, retrieved through Yandex API in the csv format.
2) Yandex bids historical data, retrieved from a personal account in csv format.

Research methods:

1) Mutual Information Scores
2) Correlation coefficients.
3) Linear regression.

The purpose and objectives of the work:

The purpose of this research is to determine the parameters that have the most influence over the price formation in Yandex Direct campaigns. My objective is to apply API and ML methods to yield a comprehensive conclusion.

The novelty and reliability of the results obtained:

The research was performed on datasets that were provided by Yandex, which is a large corporation that values its reputation. The datasets I have used for my research are published in my github and can be used to replicate my results.

Practical value:

Reduction of uncertainty when planning budget for new advertisement campaigns.

Theoretical part:

Description of MIS (Mutual Information Scores):

MIS is a measure of the extent to which knowledge of one quantity reduces uncertainty about the other. (https://www.kaggle.com/code/ryanholbrook/mutual-information)

$$MI(i,j) = \sum_{a,b} P(a_i, b_j) \cdot \log\left(\frac{P(a_i, b_j)}{P(a_i) \cdot P(b_j)}\right)$$

Description of Correlation Coefficients:

A number between +1 and −1 calculated so as to represent the linear interdependence of two variables or sets of data.

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

Description of API:

A set of functions and procedures allowing the creation of applications that access the features or data of an operating system, application, or other service.

Description of Linear Regression:

In statistics, linear regression is a linear approach for modeling the relationship between a scalar response and one or more explanatory variables

$$\beta_1 = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{m}(x_i - \bar{x})^2}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

Review and comparative analysis of sources on the topic of the project:

After reviewing possible research techniques, I came to the conclusion that the most popular technique for determining the scope of how one variable affects the other is MIS (the appropriate reference can be found in the bibliography). I have also been introduced to appropriate ML techniques through the textbook An Introduction to Statistical Learning and decided that a linear regression analysis would be appropriate for the topic of my research.

Description of the experiment:

I. Experimenting with data from Yandex Direct Forecaster:

I have used the following Yandex Direct API methods to retrieve the required data:

- CreateNewWordstatReport, GetWordstatReport: API versions of Yandex Direct Wordstat. The first method is used for creating a report based on a given phrase, while the second method is used to retrieve the report.

Code implementation:

CreateNewWordstatReport:

```
def Get_Words(main_key): #Crete a new Wordstat report and store the result in Wordstat_report.json
    report_id = 0

    filepath_minus = Path('Files/Minus_Keywords')
    lines = filepath_minus.read_text()
    main_key = main_key + lines

    body_create_report = { #ask the server to create a report
        "method": "",
        "param": {
            'Phrases': [main_key],
            'GeoID': [213]
        },
        "token": token
    }

    r1 = requests.post(link4, json.dumps(body_create_report, ensure_ascii=False).encode('utf8'))
    r1 = r1.json()
    report_id = r1['data']

    body_get_report = { #ask the server to retrieve the report, based onn id
        "method": "GetWordstatReport",
        "param": report_id,
        "token": token
    }

    time.sleep(10) #server needs time to complete the request

    r1 = requests.post(link4, json.dumps(body_get_report, ensure_ascii=False).encode('utf8'))
    r1 = r1.json()

    with open('Files/Wordstat_report.json', 'w', encoding='utf-8') as f:
        json.dump(r1, f, ensure_ascii=False, indent=4)
```

- CreateNewForecast, GetForecast:

```
def Make_forecast (file_name): #parses the csv file for api call and returns a price forecast
    file = open(file_name)
    headers = ['keywords', 'shows']
    dtypes = {'keywords': 'str', 'shows': 'int'}
    parse_dates = ['keywords', 'shows']
    data = pd.read_csv(file, sep=',', header=None, names=headers, dtype=dtypes, parse_dates=parse_dates)
    data['shows'] = pd.to_numeric(data['shows'])
    data = data.sort_values(by='shows') #sort values by shows to get the most meaningful data
    forcast_keys = np.array(data['keywords'], dtype=str)
    forcast_keys = forcast_keys[:100]
    forcast_keys = forcast_keys.tolist()

    body_for_forecast = {
        "method": "CreateNewForecast",
        "param": {
            'Phrases': forcast_keys,  # массив со словами
            'GeoID': [213],
            'Currency': "RUB",
        },
        "token": token
    }

    r1 = requests.post(link4_live, json.dumps(body_for_forecast, ensure_ascii=False).encode('utf8'))
    r1 = r1.json()
    forecast_id = r1['data']

    time.sleep(20)  # server needs time to complete the request

    body_get_forecasts = {
        "method": "GetForecast",
        "param": forecast_id,
        "token": token
    }

    r1 = requests.post(link4_live, json.dumps(body_get_forecasts, ensure_ascii=False).encode('utf8'))
    r1 = r1.json()
    with open('Files/Forecast_report.json', 'w', encoding='utf-8') as f:
        json.dump(r1, f, ensure_ascii=False, indent=4)
```

I have decided retrieve data for the following phrases (those phrases are in Russian, because Yandex is primarily used by Russians):
-колонка
-видеокарта
-монитор
-стол
-машина
-телефон
-принтер
-клавиатура
-телевизор
-ноутбук

The reports had the following labels:

-CTR
-Clicks
-Currency
-FirstPlaceCTR
-FirstPlaceClicks
-IsRubric
-Max
-Min
-Phrase
-PremiumCTR
-PremiumClicks
-PremiumMax
-PremiumMin
-Shows

```
CTR,Clicks,Currency,FirstPlaceCTR,FirstPlaceClicks,IsRubric,Max,Min,Phrase,PremiumCTR,PremiumClicks,PremiumMax,PremiumMin,Shows
100.0,2,RUB,100.0,2,No,32.34,32.34,компьютер asus laptop,100.0,2,158.89,31.5,51
0.0,0,RUB,0.0,0,No,0.3,0.3,asus x550l notebook pc,0.0,0,0.3,0.3,58
1.85,2,RUB,2.7,3,No,31.78,26.35,notebook 14s,8.48,14,222.05,85.5,169
3.64,2,RUB,3.51,2,No,28.98,12.54,asus notebook series,4.94,4,98.06,86.19,223
2.33,2,RUB,2.33,2,No,40.83,40.83,asus notebook pc,8.11,9,611.07,73.31,319
1.2,3,RUB,1.18,3,No,35.46,24.73,netbook,6.44,21,210.17,107.21,568
```

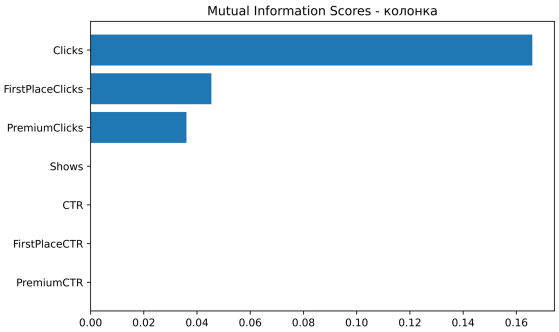
And only the following could be potentially relevant to the price formation:
-CTR
-Clicks
-FirstPlaceCTR
-FirstPlaceClicks
-PremiumCTR
-PremiumClicks
-PremiumMax
-PremiumMin
-Shows

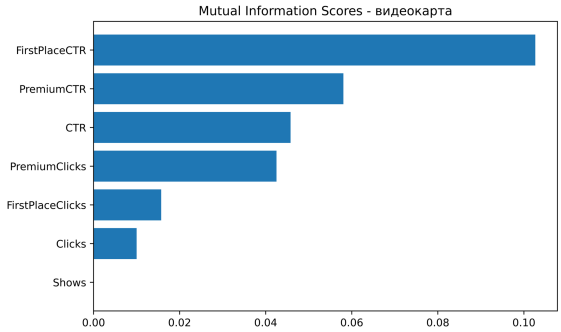
Now it was time to apply the Mutual Information Score method on the retrieved dataset:

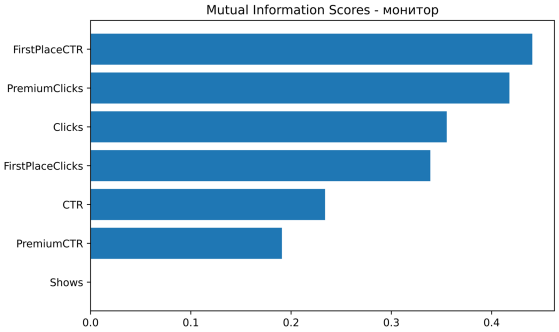
Code implementation:

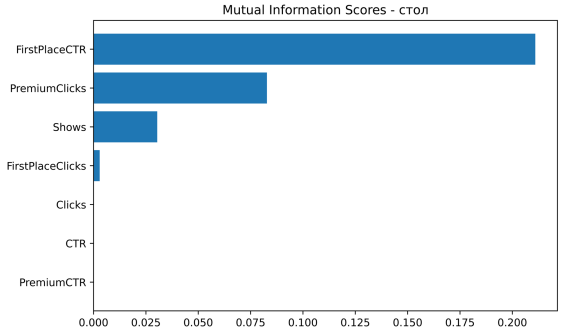
```python
import pandas as pd
from sklearn.feature_selection import mutual_info_regression
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
# data = pd.read_csv (r'Files/Forecast_report.csv')
data = pd.read_csv (r'Files/Forecast_report.csv')

X = data.copy()
y = X.pop("Max")
X = X[['Shows','Clicks','CTR','FirstPlaceCTR', 'FirstPlaceClicks', 'PremiumCTR', 'PremiumClicks']]

#CTR,Clicks,Currency,FirstPlaceCTR,FirstPlaceClicks,IsRubric,Max,Min,Phrase,PremiumCTR,PremiumClicks,PremiumMax,PremiumMin,Shows

for colname in X.select_dtypes("object"):
    X[colname], _ = X[colname].factorize()

def make_mi_scores(X, y, discrete_features):
    mi_scores = mutual_info_regression(X, y, discrete_features=discrete_features)
    mi_scores = pd.Series(mi_scores, name="MI Scores", index=X.columns)
    mi_scores = mi_scores.sort_values(ascending=False)
    return mi_scores

discrete_features = X.dtypes == float

mi_scores = make_mi_scores(X, y, discrete_features)
print(mi_scores[::3])

def plot_mi_scores(scores):
    scores = scores.sort_values(ascending=True)
    width = np.arange(len(scores))
    ticks = list(scores.index)
    plt.barh(width, scores)
    plt.yticks(width, ticks)
    plt.title("Mutual Information Scores - кондей")
    plt.savefig('кондей.png', dpi=1500, bbox_inches="tight")


plt.figure(dpi=100, figsize=(8, 5))
plot_mi_scores(mi_scores)
```

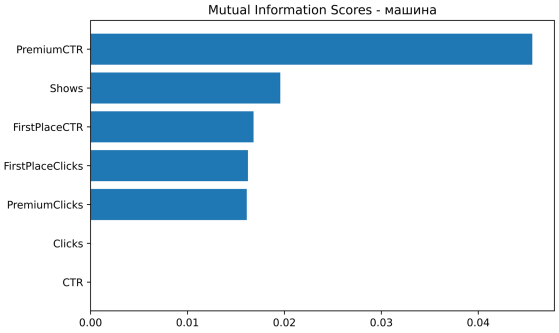
The following results were yielded, where table labeled *'all'* represent the MIS for the combination of all datasets':

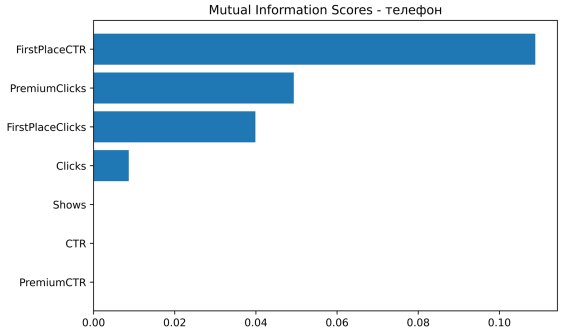

Mutual Information Scores - колонка
Clicks
FirstPlaceClicks
PremiumClicks
Shows
CTR
FirstPlaceCTR
PremiumCTR
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16

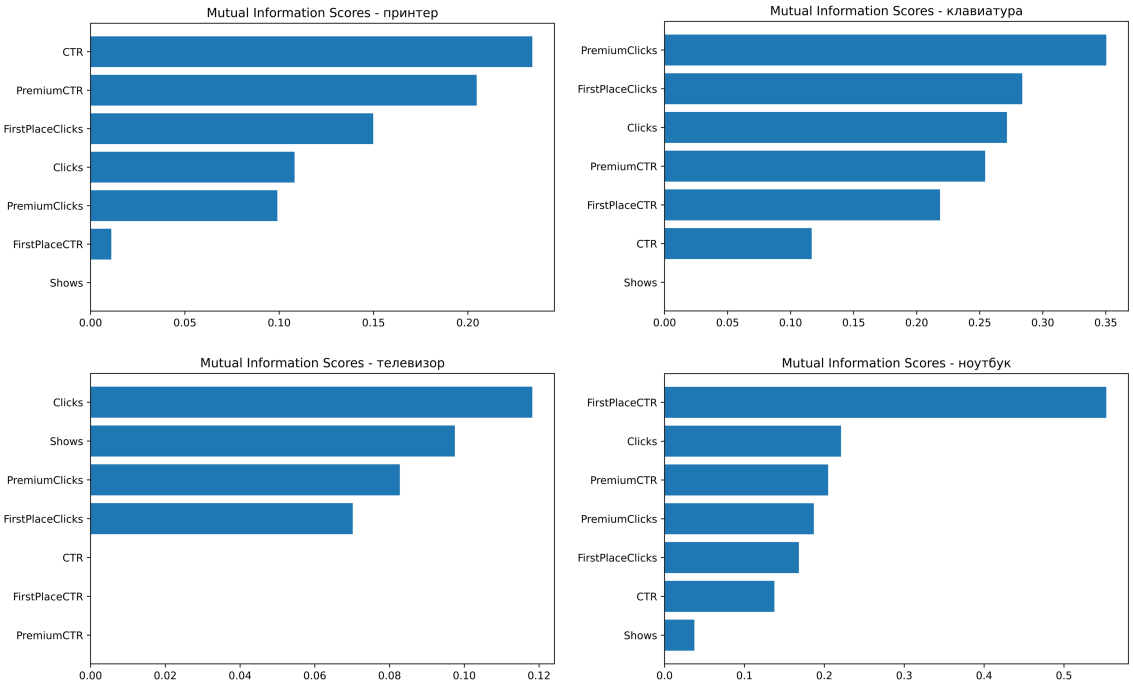

Mutual Information Scores - видеокарта
FirstPlaceCTR
PremiumCTR
CTR
PremiumClicks
FirstPlaceClicks
Clicks
Shows
0.00
0.02
0.04
0.06
0.08
0.10

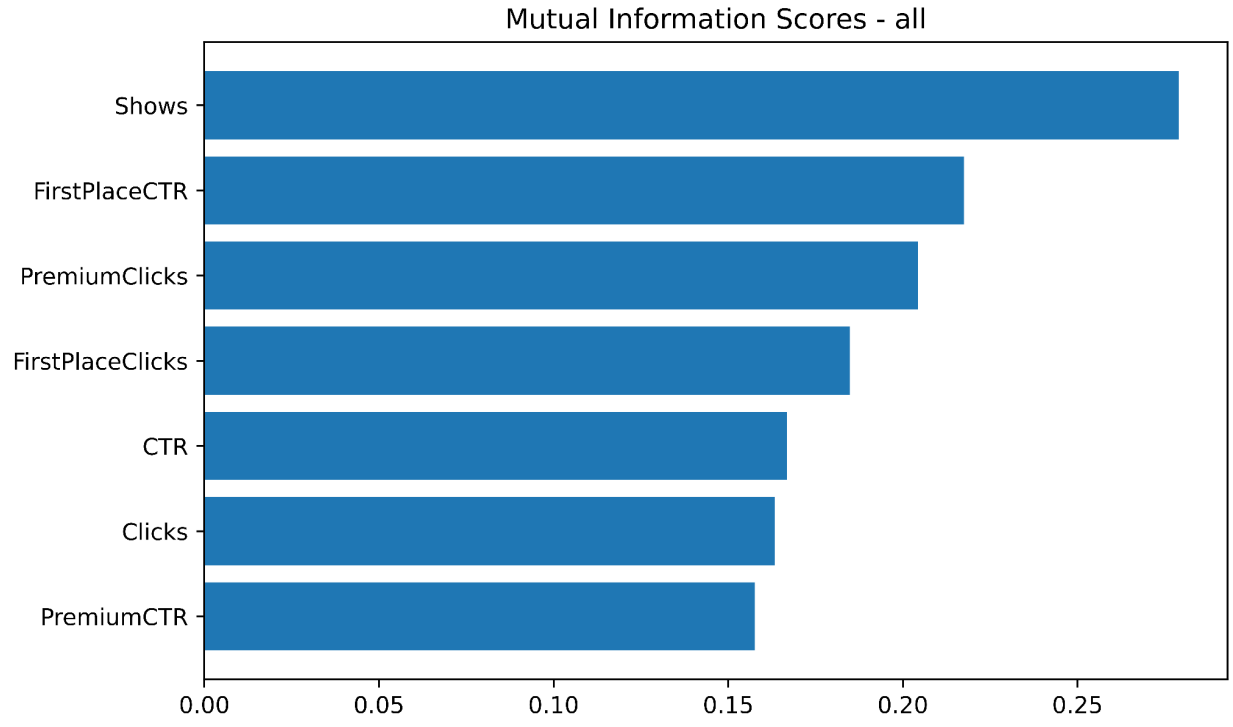

Mutual Information Scores - монитор
FirstPlaceCTR
PremiumClicks
Clicks
FirstPlaceClicks
CTR
PremiumCTR
Shows
0.0
0.1
0.2
0.3
0.4


Mutual Information Scores - стол
FirstPlaceCTR
PremiumClicks
Shows
FirstPlaceClicks
Clicks
CTR
PremiumCTR
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175
0.200


Mutual Information Scores - машина
PremiumCTR
Shows
FirstPlaceCTR
FirstPlaceClicks
PremiumClicks
Clicks
CTR
0.00
0.01
0.02
0.03
0.04


Mutual Information Scores - телефон
FirstPlaceCTR
PremiumClicks
FirstPlaceClicks
Clicks
Shows
CTR
PremiumCTR
0.00
0.02
0.04
0.06
0.08
0.10





Conclusion of Experiment #1: the parameter ‘Shows’ turned out to be the most impactful on the combined dataset.

II. Experimenting with historical data of current campaigns:

I have extracted historical data of the campaigns that I am currently running:

№	Тип	Название	Статус ↑	Стратегия	Места показ	Бюджет, ₽	Расход, ₽	Расход с НДС, ₽	Кон
		Итого				445 950,00 в неделю	0,00	0,00	
		Поиск | Сервис | service.ventmax.ru № 53065902 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление ставками	На поиске	1 500,00 в день	0,00	0,00	
		Поиск || home.ventmax.ru || СПБ Новая № 71107915 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление ставками	На поиске	2 000,00 в день	0,00	0,00	
		Поиск_Новый_Отопление № 77551486 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление ставками	На поиске	800,00 в день	0,00	0,00	
		РСЯ||Новый_Отопление № 77689831 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление с	В сетях	300,00 в день	0,00	0,00	
		Станица РСЯ||Новый_Отопление № 78004854 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление с	В сетях	300,00 в день	0,00	0,00	
		Страница_Поиск_Новый_Отопление № 78004864 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление ставками	На поиске	2 000,00 в день	0,00	0,00	
		Фильтры||Поиск № 78025970 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление ставками	На поиске	3 000,00 в день	0,00	0,00	
		РСЯ | Сервис | service.ventmax.ru № 78604681 Перейти к кампании Редактировать Статистика	Показы начнутся п... Приостановлено	Ручное управление с	В сетях	800,00 в день	0,00	0,00	

Here is a snippet of the retrieved data:

```
[31] import matplotlib.pyplot as plt, seaborn as sns
     from sklearn.model_selection import train_test_split as tts
     from sklearn.metrics import r2_score
     import statsmodels.api as sm
```

```
[2] df = pd.read_excel('2022-08-20-2022-09-19_impressions_criteria_date78025970.xls', skiprows=4)
    pd.get_option('display.max_columns')
    df.head(4)
```

	Группа	№ Группы	№ Объявления	Заголовок	Текст	Ссылка	Условие показа	Дата	Показы	Клики	CTR (%)	* Расход (руб.)	* Ср. цена клика (руб.)	Доля рекламных расходов
0	Очистка воздуха	5.013760e+09	M-12669758242	Финские фильтры для тонкой очистки воздуха	Гарантия на чистый воздух с финскими системами...	esv.company	биологическая очистка воздуха	06.09.2022	1.0	0.0	0.0	0.0	-	-
1	Очистка воздуха	5.013760e+09	M-12669758242	Финские фильтры для тонкой очистки воздуха	Гарантия на чистый воздух с финскими системами...	esv.company	блок очистки воздуха	02.09.2022	1.0	0.0	0.0	0.0	-	-
2	Очистка воздуха	5.013760e+09	M-12669758242	Финские фильтры для тонкой очистки воздуха	Гарантия на чистый воздух с финскими системами...	esv.company	блок очистки воздуха	08.09.2022	3.0	0.0	0.0	0.0	-	-
3	Очистка воздуха	5.013760e+09	M-12669758242	Финские фильтры для тонкой очистки воздуха	Гарантия на чистый воздух с финскими системами...	esv.company	камера для очистки воздуха	02.09.2022	2.0	0.0	0.0	0.0	-	-

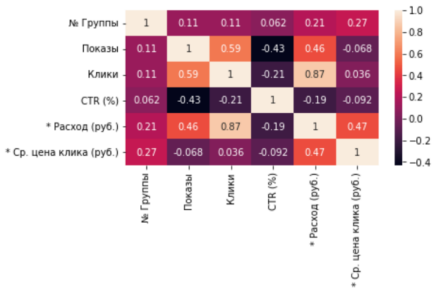
And also did some cleaning by only selecting the groups that had average price per click:

```
[69] groups = df[df['* Ср. цена клика (руб.)'] != '-']
     groups['* Ср. цена клика (руб.)'] = pd.to_numeric(groups['* Ср. цена клика (руб.)'])
     groups = groups.drop(index=[589, 588])
     groups[groups['* Ср. цена клика (руб.)'].isna()]
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
```

Then I calculated the correlation coefficients to approximate what parameters would impact the average price the most:

```
plt.rcParams['figure.figsize'] = [6, 3];  # control the size of your visualiztaions
sns.heatmap(groups.corr(), annot = True);     # note the semicolon, which suppresses extraneous output
```



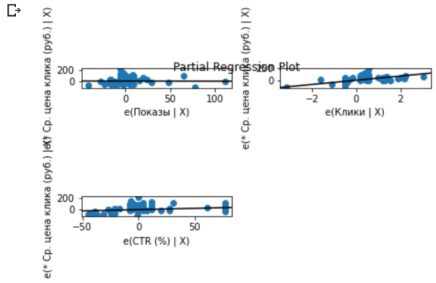
And build a linear regression:

```
mdl = sm.OLS(tY, tX)     # training on train observations
fmdl = mdl.fit()
print(fmdl.summary(title='Baseline model for Advertising dataset', alpha=.01))
pY = fmdl.predict(vX)   # predicted values on the testing set
```

```
                          Baseline model for Advertising dataset
==============================================================================================
Dep. Variable:     * Ср. цена клика (руб.)   R-squared (uncentered):                   0.654
Model:                                 OLS   Adj. R-squared (uncentered):              0.641
Method:                      Least Squares   F-statistic:                              48.54
Date:                     Sun, 18 Sep 2022   Prob (F-statistic):                    1.02e-17
Time:                             22:15:24   Log-Likelihood:                         -442.26
No. Observations:                       80   AIC:                                      890.5
Df Residuals:                           77   BIC:                                      897.7
Df Model:                                3
Covariance Type:                 nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.005      0.995]
--------------------------------------------------------------------------------
Показы         -0.0157      0.339     -0.046      0.963      -0.910       0.878
Клики          44.9496      7.205      6.239      0.000      25.920      63.980
CTR (%)         0.4609      0.180      2.565      0.012      -0.014       0.935
==============================================================================
Omnibus:                        7.919   Durbin-Watson:                   2.218
Prob(Omnibus):                  0.019   Jarque-Bera (JB):                7.402
Skew:                           0.704   Prob(JB):                       0.0247
Kurtosis:                       3.488   Cond. No.                         50.3
==============================================================================

Notes:
[1] R² is computed without centering (uncentered) since the model does not contain a constant.
[2] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Conclusion: As we see it's quite difficult to predict price formation even on real data, since train R^2 turned out to be relatively low (0.64). However, we can conclude that the parameter ‘Clicks’ is the most impactful based on the graphs below.

```
fig =  sm.graphics.plot_partregress_grid(fmdl)
fig.tight_layout(pad=4)
```



Conclusion:

From the experiments done above we see that it's best to use the parameter 'Shows' to approximate the budget required to get a certain amount of traffic (MIS was >0.25). On the other hand, we also saw that parameter 'Clicks' does have a prominent effect on price charged on the auction (the pure effect of the 'Clicks' parameter, keeping other factors constant, on average was 44.5 rubles increase in average bid pricing, per click). With this knowledge and a solid understanding of ML principles, one could use this research as a foundation for building an automated Yandex Direct bid manager.

Bibliography:

– Yandex Direct API documentation: https://yandex.ru/dev/direct/doc/dg/concepts/overview.html
– Mutual Information Scores: https://www.kaggle.com/code/ryanholbrook/mutual-information
–An Introduction to Statistical Learning (Gareth James • Daniela Witten • Trevor Hastie • Robert Tibshirani, Second Edition)
–Source for MIS formula: http://mistic.leloir.org.ar/docs/help.html