In [1]:

```
import pandas as pd
```

In [2]:

```
df = pd.read_csv('survey.csv')
```

In [3]:

```
df.head()
```

Out[3]:

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_interf
0	27-08-2014 11:29	37	Female	United States	IL	NaN	No	Yes	O
1	27-08-2014 11:29	44	M	United States	IN	NaN	No	No	Ra
2	27-08-2014 11:29	32	Male	Canada	NaN	NaN	No	No	Ra
3	27-08-2014 11:29	31	Male	United Kingdom	NaN	NaN	Yes	Yes	O
4	27-08-2014 11:30	31	Male	United States	TX	NaN	No	No	Ne

5 rows × 27 columns

In [4]:

```
df.tail()
```

Out[4]:

	Timestamp	Age	Gender	Country	state	self_employed	family_history	treatment	work_in
1254	12-09-2015 11:17	26	male	United Kingdom	NaN	No	No	Yes	
1255	26-09-2015 01:07	32	Male	United States	IL	No	Yes	Yes	
1256	07-11-2015 12:36	34	male	United States	CA	No	Yes	Yes	Son
1257	30-11-2015 21:25	46	f	United States	NC	No	No	No	
1258	01-02-2016 23:04	25	Male	United States	IL	No	Yes	Yes	Son

5 rows × 27 columns

In [5]:

```
df.shape
```

Out[5]:

```
(1259, 27)
```

In [6]:

```
df.columns
```

Out[6]:

```
Index(['Timestamp', 'Age', 'Gender', 'Country', 'state', 'self_employed',
       'family_history', 'treatment', 'work_interfere', 'no_employees',
       'remote_work', 'tech_company', 'benefits', 'care_options',
       'wellness_program', 'seek_help', 'anonymity', 'leave',
       'mental_health_consequence', 'phys_health_consequence', 'coworkers',
       'supervisor', 'mental_health_interview', 'phys_health_interview',
       'mental_vs_physical', 'obs_consequence', 'comments'],
      dtype='object')
```

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
1
```

In [8]:

```
df = df.drop_duplicates()
```

In [9]:

```
df.isnull().sum()
```

Out[9]:

```
Timestamp                      0
Age                            0
Gender                         0
Country                        0
state                        514
self_employed                 18
family_history                 0
treatment                      0
work_interfere               264
no_employees                   0
remote_work                    0
tech_company                   0
benefits                       0
care_options                   0
wellness_program               0
seek_help                      0
anonymity                      0
leave                          0
mental_health_consequence      0
phys_health_consequence        0
coworkers                      0
supervisor                     0
mental_health_interview        0
phys_health_interview          0
mental_vs_physical             0
obs_consequence                0
comments                    1094
dtype: int64
```

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1258 entries, 0 to 1258
Data columns (total 27 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   Timestamp                  1258 non-null   object
 1   Age                        1258 non-null   int64 
 2   Gender                     1258 non-null   object
 3   Country                    1258 non-null   object
 4   state                      744 non-null    object
 5   self_employed              1240 non-null   object
 6   family_history             1258 non-null   object
 7   treatment                  1258 non-null   object
 8   work_interfere             994 non-null    object
 9   no_employees               1258 non-null   object
 10  remote_work                1258 non-null   object
 11  tech_company               1258 non-null   object
 12  benefits                   1258 non-null   object
 13  care_options               1258 non-null   object
 14  wellness_program           1258 non-null   object
 15  seek_help                  1258 non-null   object
 16  anonymity                  1258 non-null   object
 17  leave                      1258 non-null   object
 18  mental_health_consequence  1258 non-null   object
 19  phys_health_consequence    1258 non-null   object
 20  coworkers                  1258 non-null   object
 21  supervisor                 1258 non-null   object
 22  mental_health_interview    1258 non-null   object
 23  phys_health_interview      1258 non-null   object
 24  mental_vs_physical         1258 non-null   object
 25  obs_consequence            1258 non-null   object
 26  comments                   164 non-null    object
dtypes: int64(1), object(26)
memory usage: 275.2+ KB
```

In [11]:

```
df.describe()
```

Out[11]:

	Age
count	1.258000e+03
mean	7.949129e+07
std	2.819419e+09
min	-1.726000e+03
25%	2.700000e+01
50%	3.100000e+01
75%	3.600000e+01
max	1.000000e+11

In [12]:

```
df.nunique()
```

Out[12]:

```
Timestamp                    884
Age                           53
Gender                        49
Country                       48
state                         45
self_employed                  2
family_history                 2
treatment                      2
work_interfere                 4
no_employees                   6
remote_work                    2
tech_company                   2
benefits                       3
care_options                   3
wellness_program               3
seek_help                      3
anonymity                      3
leave                          5
mental_health_consequence      3
phys_health_consequence        3
coworkers                      3
supervisor                     3
mental_health_interview        3
phys_health_interview          3
mental_vs_physical             3
obs_consequence                2
comments                     160
dtype: int64
```

In [13]:

```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [14]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [15]:

```
df_cat = df[['self_employed','family_history', 'treatment', 'work_interfere',
             'no_employees','remote_work', 'tech_company', 'benefits', 'care_options',
       'wellness_program', 'seek_help', 'anonymity', 'leave',
       'mental_health_consequence', 'phys_health_consequence', 'coworkers',
       'supervisor', 'mental_health_interview', 'phys_health_interview',
       'mental_vs_physical', 'obs_consequence']]
```

In [16]:

```
for i in df_cat.columns:
    print(df_cat[i].unique())
```

```
[nan 'Yes' 'No']
['No' 'Yes']
['Yes' 'No']
['Often' 'Rarely' 'Never' 'Sometimes' nan]
['Jun-25' 'More than 1000' '26-100' '100-500' '01-May' '500-1000']
['No' 'Yes']
['Yes' 'No']
['Yes' "Don't know" 'No']
['Not sure' 'No' 'Yes']
['No' "Don't know" 'Yes']
['Yes' "Don't know" 'No']
['Yes' "Don't know" 'No']
['Somewhat easy' "Don't know" 'Somewhat difficult' 'Very difficult'
 'Very easy']
['No' 'Maybe' 'Yes']
['No' 'Yes' 'Maybe']
['Some of them' 'No' 'Yes']
['Yes' 'No' 'Some of them']
['No' 'Yes' 'Maybe']
['Maybe' 'No' 'Yes']
['Yes' "Don't know" 'No']
['No' 'Yes']
```

In [17]:

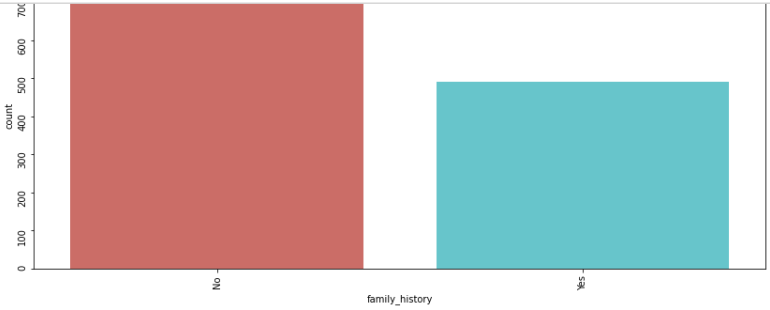
```
for i in df_cat.columns:
    print(df_cat[i].value_counts())
```

```
No     1094
Yes     146
Name: self_employed, dtype: int64
No     767
Yes    491
Name: family_history, dtype: int64
Yes    636
No     622
Name: treatment, dtype: int64
Sometimes    465
Never        213
Rarely       172
Often        144
Name: work_interfere, dtype: int64
Jun-25            289
26-100            289
More than 1000    282
100-500           176
01-May            162
500-1000           60
Name: no_employees, dtype: int64
No     882
Yes    376
Name: remote_work, dtype: int64
Yes    1031
No      227
Name: tech_company, dtype: int64
Yes           477
Don't know    408
No            373
Name: benefits, dtype: int64
No          500
Yes         444
Not sure    314
Name: care_options, dtype: int64
No            841
Yes           229
Don't know    188
Name: wellness_program, dtype: int64
No            645
Don't know    363
Yes           250
Name: seek_help, dtype: int64
Don't know    819
Yes           375
No             64
Name: anonymity, dtype: int64
Don't know            562
Somewhat easy         266
Very easy             206
Somewhat difficult    126
Very difficult         98
Name: leave, dtype: int64
No       490
Maybe    477
Yes      291
Name: mental_health_consequence, dtype: int64
No       925
Maybe    272
Yes       61
Name: phys_health_consequence, dtype: int64
Some of them    773
```

```
No              260
Yes             225
Name: coworkers, dtype: int64
Yes             516
No              392
Some of them    350
Name: supervisor, dtype: int64
No       1007
Maybe     207
Yes        44
Name: mental_health_interview, dtype: int64
Maybe    557
No       499
Yes      202
Name: phys_health_interview, dtype: int64
Don't know    576
Yes           343
No            339
Name: mental_vs_physical, dtype: int64
No     1074
Yes     184
Name: obs_consequence, dtype: int64
```

In [18]:

```
for i in df_cat.columns:
    plt.figure(figsize = (14,6))
    sns.countplot(df_cat[i],data=df_cat, palette = 'hls')
    plt.xticks(rotation = 90)
    plt.yticks(rotation = 90)
    plt.show()
```



In [19]:

```
print(df['Country'].value_counts())
```

```
United States             751
United Kingdom            184
Canada                     72
Germany                    45
Ireland                    27
Netherlands                27
Australia                  21
France                     13
India                      10
New Zealand                 8
Poland                      7
Switzerland                 7
Sweden                      7
Italy                       7
South Africa                6
Belgium                     6
Brazil                      6
Israel                      5
Singapore                   4
Bulgaria                    4
Austria                     3
Finland                     3
Mexico                      3
Russia                      3
Denmark                     2
Greece                      2
Colombia                    2
Croatia                     2
Portugal                    2
Moldova                     1
Georgia                     1
Bahamas, The                1
China                       1
Thailand                    1
Czech Republic              1
Norway                      1
Romania                     1
Nigeria                     1
Japan                       1
Hungary                     1
Bosnia and Herzegovina      1
Uruguay                     1
Spain                       1
Zimbabwe                    1
Latvia                      1
Costa Rica                  1
Slovenia                    1
Philippines                 1
Name: Country, dtype: int64
```

In [20]:

```
print(df['state'].unique())
```

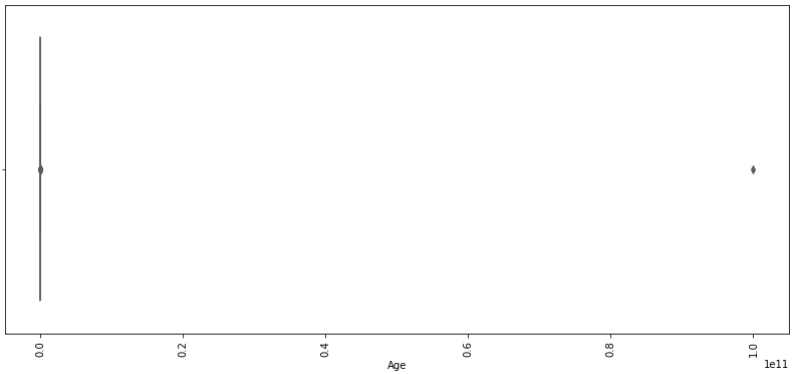
```
['IL' 'IN' nan 'TX' 'TN' 'MI' 'OH' 'CA' 'CT' 'MD' 'NY' 'NC' 'MA' 'IA' 'PA'
 'WA' 'WI' 'UT' 'NM' 'OR' 'FL' 'MN' 'MO' 'AZ' 'CO' 'GA' 'DC' 'NE' 'WV'
 'OK' 'KS' 'VA' 'NH' 'KY' 'AL' 'NV' 'NJ' 'SC' 'VT' 'SD' 'ID' 'MS' 'RI'
 'WY' 'LA' 'ME']
```

In [21]:

```
df.drop(columns=['Timestamp', 'Country', 'state', 'comments'], inplace = True)
```

In [22]:

```
plt.figure(figsize = (14,6))
sns.boxplot(df['Age'],data=df, palette = 'hls')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



In [23]:

```
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)
```

```
Age    9.0
dtype: float64
```

In [24]:

```
df[~((df < (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

Out[24]:

	Age	Gender	self_employed	family_history	treatment	work_interfere	no_employees	remo
0	37	Female	NaN	No	Yes	Often	Jun-25	
1	44	M	NaN	No	No	Rarely	More than 1000	
2	32	Male	NaN	No	No	Rarely	Jun-25	
3	31	Male	NaN	Yes	Yes	Often	26-100	
4	31	Male	NaN	No	No	Never	100-500	
...	...	...	...	...	...	...	...	
1254	26	male	No	No	Yes	NaN	26-100	
1255	32	Male	No	Yes	Yes	Often	26-100	
1256	34	male	No	Yes	Yes	Sometimes	More than 1000	
1257	46	f	No	No	No	NaN	100-500	
1258	25	Male	No	Yes	Yes	Sometimes	26-100	

1218 rows × 23 columns

In [25]:

```
df.shape
```

Out[25]:

```
(1258, 23)
```

In [26]:

```
df.drop(df[df['Age'] < 0].index, inplace = True)
df.drop(df[df['Age'] > 100].index, inplace = True)
```

In [27]:

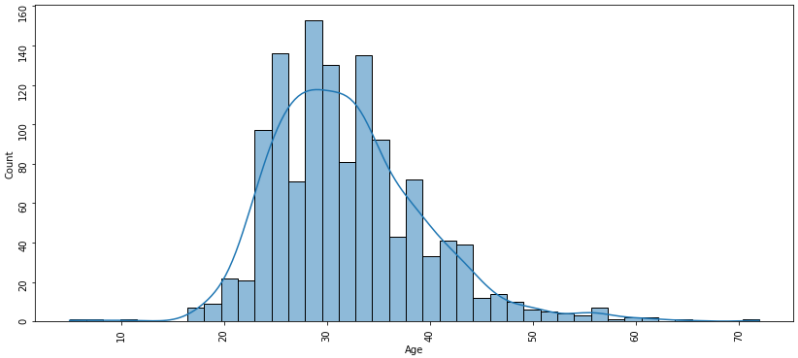
```
df['Age'].unique()
```

Out[27]:

```
array([37, 44, 32, 31, 33, 35, 39, 42, 23, 29, 36, 27, 46, 41, 34, 30, 40,
       38, 50, 24, 18, 28, 26, 22, 19, 25, 45, 21, 43, 56, 60, 54, 55, 48,
       20, 57, 58, 47, 62, 51, 65, 49,  5, 53, 61,  8, 11, 72],
      dtype=int64)
```

In [28]:

```
plt.figure(figsize = (14,6))
sns.histplot(df['Age'], kde = True, palette = 'hls')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



In [29]:

```
df['Gender'].replace(['Male ', 'male', 'M', 'm', 'Male', 'Cis Male',
                      'Man', 'cis male', 'Mail', 'Male-ish', 'Male (CIS)',
                       'Cis Man', 'msle', 'Malr', 'Mal', 'maile', 'Make',], 'Male', inplace =

df['Gender'].replace(['Female ', 'female', 'F', 'f', 'Woman', 'Female',
                      'femail', 'Cis Female', 'cis-female/femme', 'Femake', 'Female (cis)',
                      'woman',], 'Female', inplace = True)

df["Gender"].replace(['Female (trans)', 'queer/she/they', 'non-binary',
                      'fluid', 'queer', 'Androgyne', 'Trans-female', 'male leaning androgynou
                       'Agender', 'A little about you', 'Nah', 'All',
                       'ostensibly male, unsure what that really means',
                       'Genderqueer', 'Enby', 'p', 'Neuter', 'something kinda male?',
                       'Guy (-ish) ^_^', 'Trans woman',], 'Other', inplace = True)
```

In [30]:

```
df['Gender'].unique()
```

Out[30]:

```
array(['Female', 'Male', 'Other'], dtype=object)
```

In [31]:

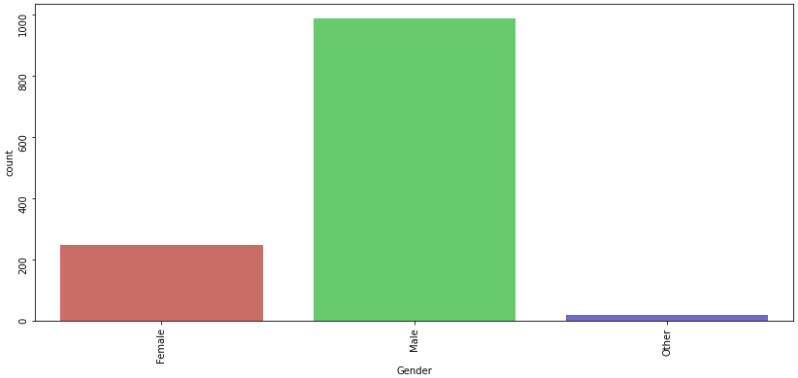
```
df['Gender'].value_counts()
```

Out[31]:

```
Male      987
Female    247
Other      19
Name: Gender, dtype: int64
```

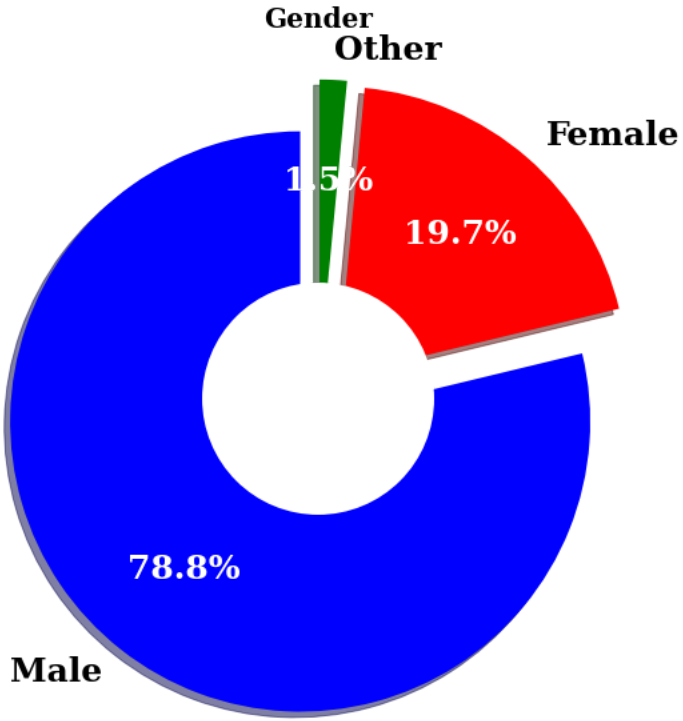
In [32]:

```
plt.figure(figsize = (14,6))
sns.countplot(df['Gender'],data=df, palette = 'hls')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



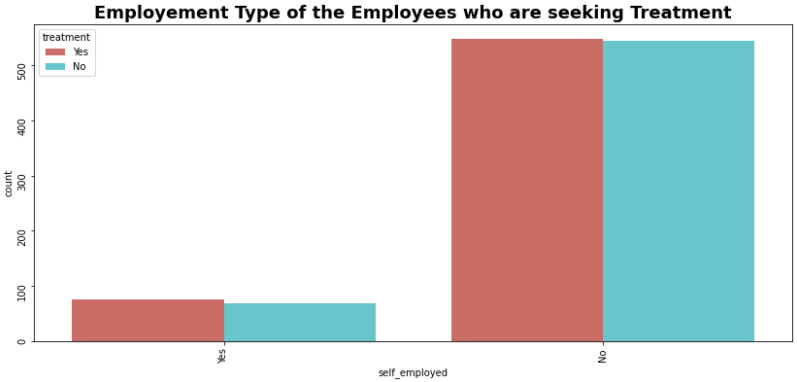
In [33]:

```
gender_data = df['Gender'].value_counts()

explode = (0.1, 0.1, 0.1)
plt.figure(figsize=(14, 10))
patches, texts, pcts = plt.pie(gender_data,
                               labels = gender_data.index,
                               colors = ['blue', 'red', 'green'],
                               pctdistance = 0.65,
                               shadow = True,
                               startangle = 90,
                               explode = explode,
                               autopct = '%1.1f%%',
                               textprops={ 'fontsize': 25,
                                           'color': 'black',
                                           'weight': 'bold',
                                           'family': 'serif' })
plt.setp(pcts, color='white')

hfont = {'fontname':'serif', 'weight': 'bold'}
plt.title('Gender', size=20, **hfont)

centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.show()
```


Gender
Other
Female
1.5%
19.7%
78.8%
Male

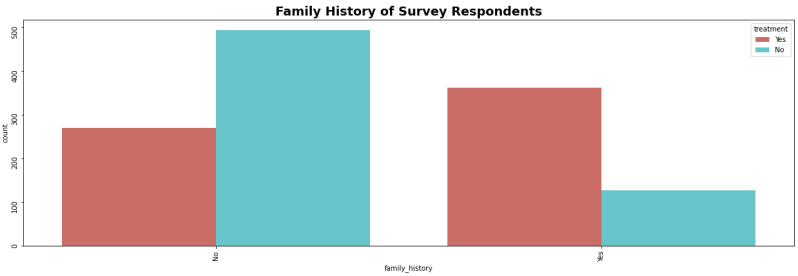
In [34]:

```
plt.figure(figsize = (14,6))
plt.title('Employement Type of the Employees who are seeking Treatment',
          fontsize=18, fontweight='bold')
sns.countplot(df['self_employed'], hue = df['treatment'],data=df,
              palette = 'hls')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



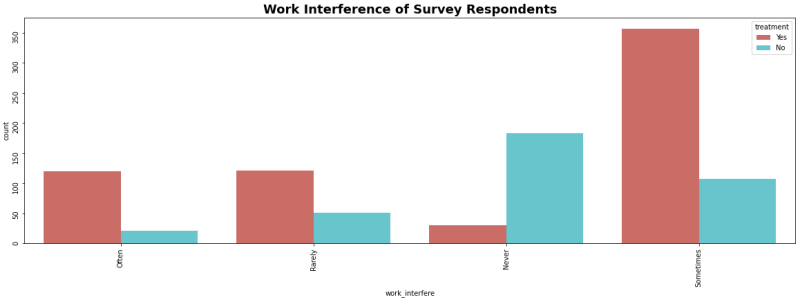
In [35]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['family_history'], hue = df['treatment'], palette='hls')
plt.title('Family History of Survey Respondents', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



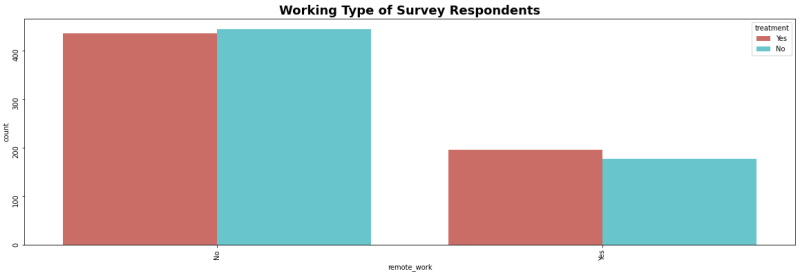
In [36]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['work_interfere'], hue = df['treatment'],
              palette = 'hls')
plt.title('Work Interference of Survey Respondents', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



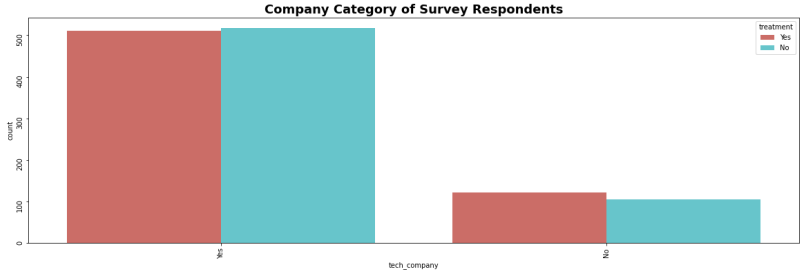
In [37]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['remote_work'], hue = df['treatment'], palette='hls')
plt.title('Working Type of Survey Respondents', fontsize=18, fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



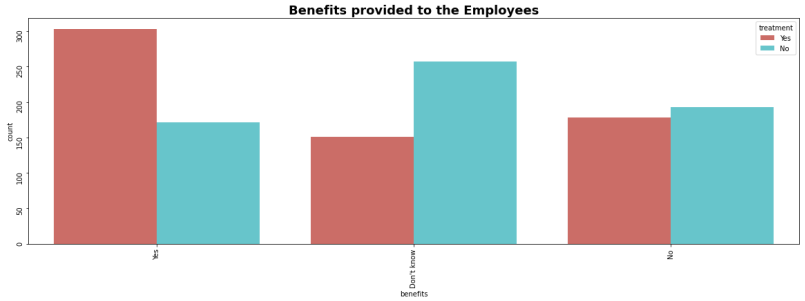
In [38]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['tech_company'], hue = df['treatment'],
              palette='hls')
plt.title('Company Category of Survey Respondents', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



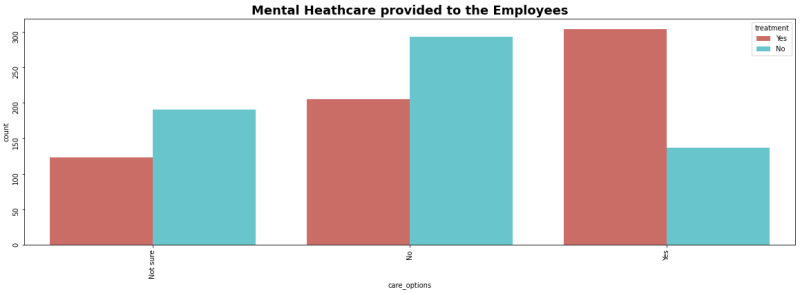
In [39]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['benefits'], hue = df['treatment'], palette='hls')
plt.title('Benefits provided to the Employees', fontsize=18, fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



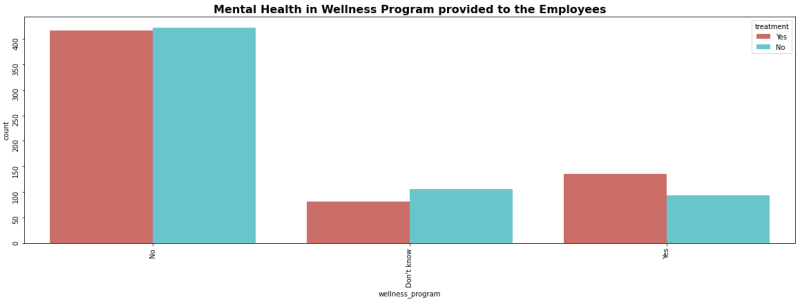
In [40]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['care_options'], hue = df['treatment'], palette='hls')
plt.title('Mental Heathcare provided to the Employees', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



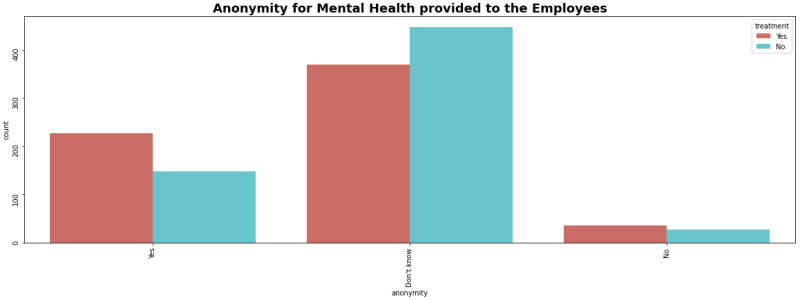
In [41]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['wellness_program'], hue = df['treatment'],
              palette='hls')
plt.title('Mental Health in Wellness Program provided to the Employees',
          fontsize=16, fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



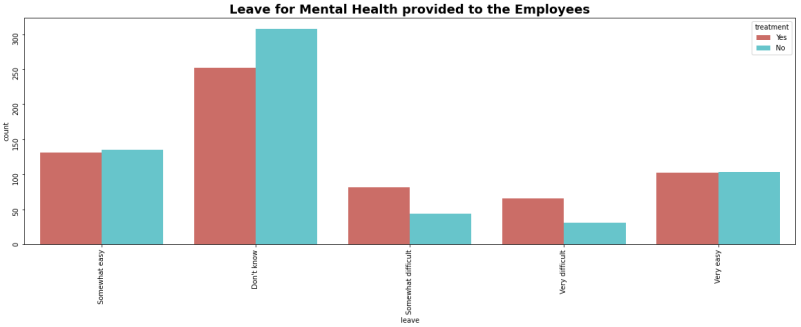
In [42]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['anonymity'], hue = df['treatment'],
              palette='hls')
plt.title('Anonymity for Mental Health provided to the Employees', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



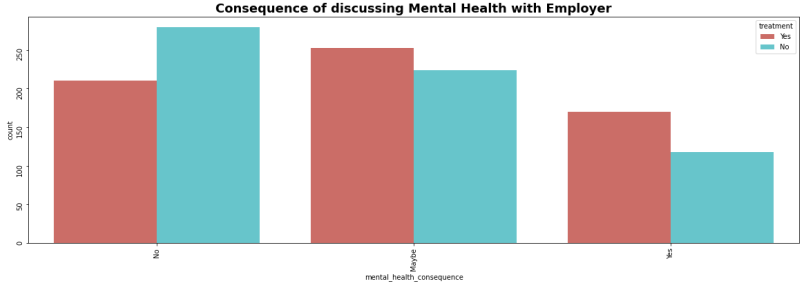
In [43]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['leave'], hue = df['treatment'],
              palette='hls')
plt.title('Leave for Mental Health provided to the Employees', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



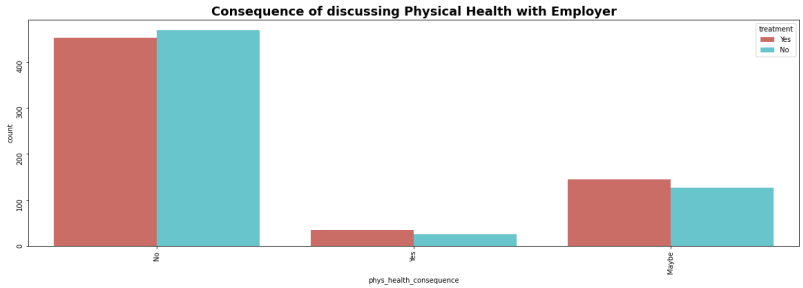
In [44]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['mental_health_consequence'], hue = df['treatment'],
              palette='hls')
plt.title('Consequence of discussing Mental Health with Employer', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



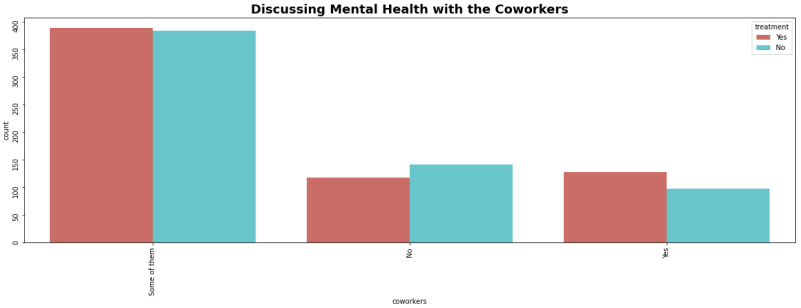
In [45]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['phys_health_consequence'], hue = df['treatment'],
              palette='hls')
plt.title('Consequence of discussing Physical Health with Employer',
          fontsize=18, fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



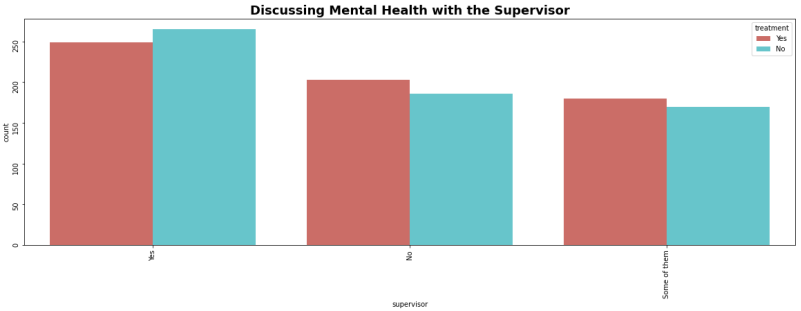
In [46]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['coworkers'], hue = df['treatment'],
              palette='hls')
plt.title('Discussing Mental Health with the Coworkers', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



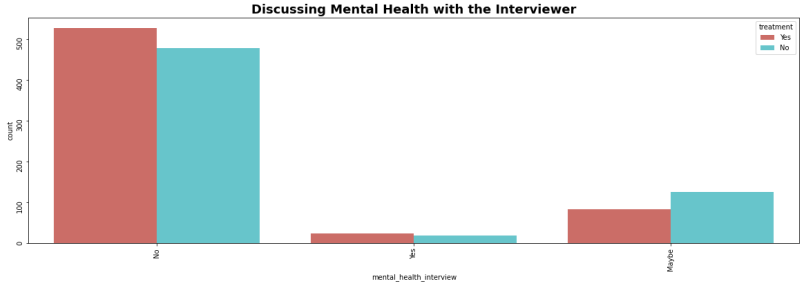
In [47]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['supervisor'], hue = df['treatment'],
              palette='hls')
plt.title('Discussing Mental Health with the Supervisor', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



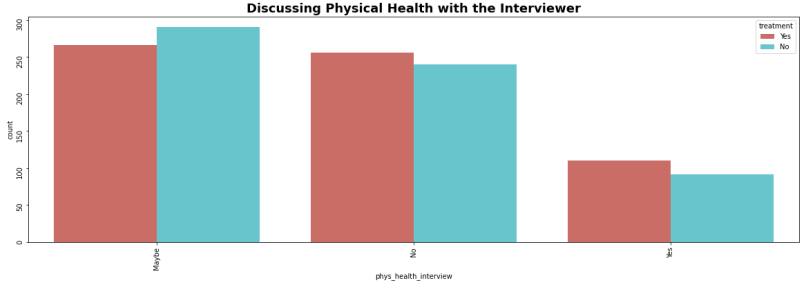
In [48]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['mental_health_interview'], hue = df['treatment'],
              palette='hls')
plt.title('Discussing Mental Health with the Interviewer', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



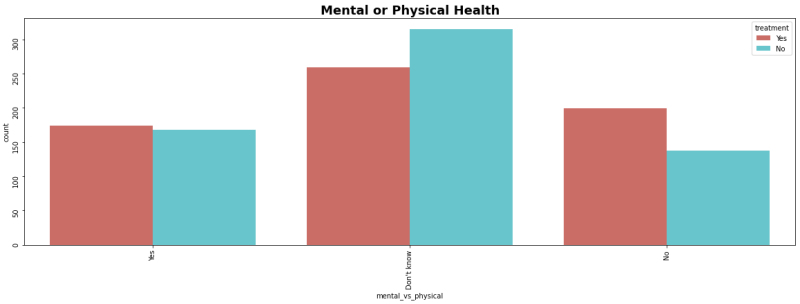
In [49]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['phys_health_interview'], hue = df['treatment'],
              palette='hls')
plt.title('Discussing Physical Health with the Interviewer', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



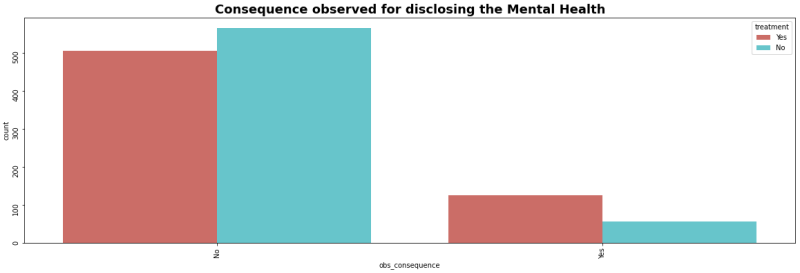
In [50]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['mental_vs_physical'], hue = df['treatment'],
              palette='hls')
plt.title('Mental or Physical Health', fontsize=18, fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



In [51]:

```
plt.figure(figsize = (20,6))
sns.countplot(df['obs_consequence'], hue = df['treatment'],
              palette='hls')
plt.title('Consequence observed for disclosing the Mental Health', fontsize=18,
          fontweight='bold')
plt.xticks(rotation = 90)
plt.yticks(rotation = 90)
plt.show()
```



In [52]:

```
df['work_interfere'] = df['work_interfere'].fillna('Not Avialable' )
```

In [53]:

```
df['self_employed'] = df['self_employed'].fillna('No')
```

In [54]:

```
df.isnull().sum()
```

Out[54]:

```
Age                          0
Gender                       0
self_employed                0
family_history               0
treatment                    0
work_interfere               0
no_employees                 0
remote_work                  0
tech_company                 0
benefits                     0
care_options                 0
wellness_program             0
seek_help                    0
anonymity                    0
leave                        0
mental_health_consequence    0
phys_health_consequence      0
coworkers                    0
supervisor                   0
mental_health_interview      0
phys_health_interview        0
mental_vs_physical           0
obs_consequence              0
dtype: int64
```

In [55]:

```
from sklearn.preprocessing import LabelEncoder
```

In [56]:

```
object_cols = ['Gender', 'self_employed', 'family_history', 'treatment',
       'work_interfere', 'no_employees', 'remote_work', 'tech_company',
       'benefits', 'care_options', 'wellness_program', 'seek_help',
       'anonymity', 'leave', 'mental_health_consequence',
       'phys_health_consequence', 'coworkers', 'supervisor',
       'mental_health_interview', 'phys_health_interview',
       'mental_vs_physical', 'obs_consequence']
```

In [57]:

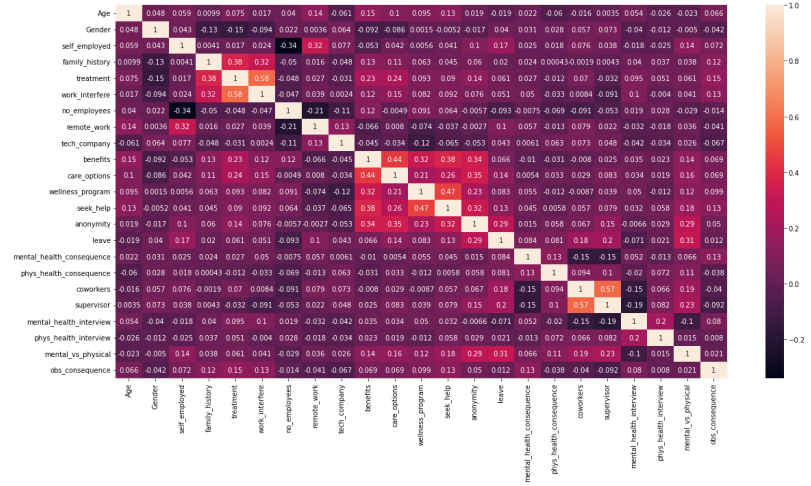
```
label_encoder = LabelEncoder()
for col in object_cols:
    label_encoder.fit(df[col])
    df[col] = label_encoder.transform(df[col])
```

In [58]:

```
corr = df.corr()
```

In [59]:

```
plt.figure(figsize = (20,10))
sns.heatmap(corr, annot = True)
plt.show()
```



In [60]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, recall_score, plot_roc_curve, confusion_matrix,
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingCla
from xgboost.sklearn import XGBClassifier
import xgboost as xgb
from sklearn import metrics
```

In [61]:

```
from sklearn.model_selection import train_test_split
X = df.drop('treatment', axis = 1)
y = df['treatment']

X_train, X_test, y_train, y_test = train_test_split(X,y, stratify = y,
                                                    test_size = 0.25,
                                                    random_state = 40)
```

In [62]:

```
key = ['LogisticRegression','KNeighborsClassifier','DecisionTreeClassifier','RandomForestCla
value = [LogisticRegression(), KNeighborsClassifier(n_neighbors = 2, weights ='uniform'), De
models = dict(zip(key,value))
models
```

Out[62]:

```
{'LogisticRegression': LogisticRegression(),
 'KNeighborsClassifier': KNeighborsClassifier(n_neighbors=2),
 'DecisionTreeClassifier': DecisionTreeClassifier(random_state=10),
 'RandomForestClassifier': RandomForestClassifier(n_estimators=60, random_stat
e=0),
 'GradientBoostingClassifier': GradientBoostingClassifier(random_state=20),
 'AdaBoostClassifier': AdaBoostClassifier(),
 'XGBClassifier': XGBClassifier(base_score=None, booster='gbtree', callbacks=N
one,
               colsample_bylevel=None, colsample_bynode=None,
               colsample_bytree=None, early_stopping_rounds=None,
               enable_categorical=False, eval_metric=None, gamma=None,
               gpu_id=None, grow_policy=None, importance_type=None,
               interaction_constraints=None, learning_rate=None, max_bin=None,
               max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
               max_leaves=None, min_child_weight=None, missing=nan,
               monotone_constraints=None, n_estimators=100, n_jobs=None,
               num_parallel_tree=None, predictor=None, random_state=0,
               reg_alpha=None, reg_lambda=None, ...)}
```

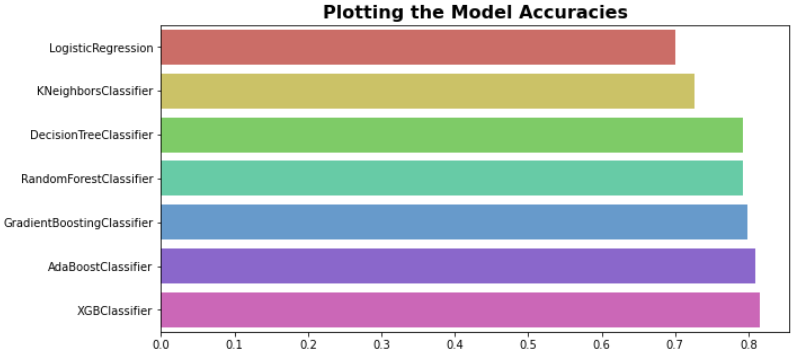
In [63]:

```
predicted =[]
for name,algo in models.items():
    model=algo
    model.fit(X_train,y_train)
    predict = model.predict(X_test)
    acc = accuracy_score(y_test, predict)
    predicted.append(acc)
    print(name,acc)
```

```
LogisticRegression 0.7929936305732485
KNeighborsClassifier 0.7006369426751592
DecisionTreeClassifier 0.7261146496815286
RandomForestClassifier 0.8152866242038217
GradientBoostingClassifier 0.8089171974522293
AdaBoostClassifier 0.7929936305732485
XGBClassifier 0.7993630573248408
```

In [64]:

```
plt.figure(figsize = (10,5))
ax = sns.barplot(x = predicted, y = key, palette='hls', order=predicted.sort())
plt.title("Plotting the Model Accuracies", fontsize=16, fontweight="bold")
plt.show()
```

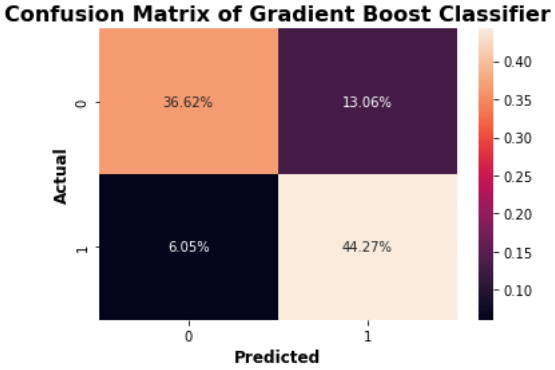


In [65]:

```
import numpy as np
```

In [66]:

```
gbc = GradientBoostingClassifier()
gbc.fit(X_train,y_train)
pred = gbc.predict(X_test)
cf_matrix = confusion_matrix(y_test, pred)
sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True,
            fmt='.2%')
plt.title('Confusion Matrix of Gradient Boost Classifier', fontweight='bold', fontsize=16)
plt.xlabel('Predicted', fontweight='bold', fontsize=12)
plt.ylabel('Actual', fontweight='bold', fontsize=12)
plt.show()
```



In [67]:

```
fpr, tpr, thresholds = metrics.roc_curve(y_test, pred)
plt.figure(figsize=(8,8))
roc_auc = metrics.auc(fpr, tpr)
plt.plot(fpr, tpr, color='darkorange', label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.rcParams['font.size'] = 12
plt.title('ROC Curve', fontweight = 'bold', fontsize=16)
plt.xlabel('False Positive Rate (1 - Specificity)', fontweight = 'bold', fontsize=14)
plt.ylabel('True Positive Rate (Sensitivity)', fontweight = 'bold', fontsize=14)
plt.legend(loc="lower right")
plt.show()
metrics.roc_curve(y_test, pred)
plt.show()
```