case_study1

October 31, 2021

0.1 DDA Full Stack Interview

Part 1
Samir Farhoumand

0.1.1 Case Study 1

[1]:
```
#import pandas
import pandas as pd
#import requests to get around 406 denial
import requests
import io
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

#url for csv
csv_url = "https://www.openintro.org/data/csv/loans_full_schema.csv"

#import data
urlData = requests.get(csv_url, headers={"User-Agent": "XY"}).content
df = pd.read_csv(io.StringIO(urlData.decode('utf-8')))

df
```

[1]:

	emp_title	emp_length	state	homeownership	annual_income	\
0	global config engineer	3.0	NJ	MORTGAGE	90000.0	
1	warehouse office clerk	10.0	HI	RENT	40000.0	
2	assembly	3.0	WI	RENT	40000.0	
3	customer service	1.0	PA	RENT	30000.0	
4	security supervisor	10.0	CA	RENT	35000.0	
…	…	…	…	…	…	
9995	owner	10.0	TX	RENT	108000.0	
9996	director	8.0	PA	MORTGAGE	121000.0	
9997	toolmaker	10.0	CT	MORTGAGE	67000.0	
9998	manager	1.0	WI	MORTGAGE	80000.0	
9999	operations analyst	3.0	CT	RENT	66000.0	

```
      verified_income  debt_to_income  annual_income_joint  \
0            Verified           18.01                  NaN
1       Not Verified            5.04                  NaN
2     Source Verified           21.15                  NaN
3       Not Verified           10.16                  NaN
4            Verified           57.96              57000.0
…               …              …                     …
9995  Source Verified           22.28                  NaN
9996         Verified           32.38                  NaN
9997         Verified           45.26             107000.0
9998  Source Verified           11.99                  NaN
9999    Not Verified           20.82                  NaN

     verification_income_joint  debt_to_income_joint  …  sub_grade  \
0                          NaN                   NaN  …         C3
1                          NaN                   NaN  …         C1
2                          NaN                   NaN  …         D1
3                          NaN                   NaN  …         A3
4                     Verified                 37.66  …         C3
…                        …                     …  …        …
9995                       NaN                   NaN  …         A4
9996                       NaN                   NaN  …         D3
9997           Source Verified                 29.57  …         E2
9998                       NaN                   NaN  …         A1
9999                       NaN                   NaN  …         B4

      issue_month  loan_status  initial_listing_status  disbursement_method  \
0       Mar-2018      Current                   whole                 Cash
1       Feb-2018      Current                   whole                 Cash
2       Feb-2018      Current              fractional                 Cash
3       Jan-2018      Current                   whole                 Cash
4       Mar-2018      Current                   whole                 Cash
…          …          …                        …                    …
9995    Jan-2018      Current                   whole                 Cash
9996    Feb-2018      Current                   whole                 Cash
9997    Feb-2018      Current              fractional                 Cash
9998    Feb-2018      Current                   whole                 Cash
9999    Feb-2018      Current                   whole                 Cash

       balance  paid_total  paid_principal  paid_interest  paid_late_fees
0     27015.86     1999.33          984.14        1015.19             0.0
1      4651.37      499.12          348.63         150.49             0.0
2      1824.63      281.80          175.37         106.43             0.0
3     18853.26     3312.89         2746.74         566.15             0.0
4     21430.15     2324.65         1569.85         754.80             0.0
…        …         …              …              …              …
9995  21586.34     2969.80         2413.66         556.14             0.0
```

```
9996   9147.44     1456.31          852.56         603.75              0.0
9997  27617.65     4620.80         2382.35        2238.45              0.0
9998  21518.12     2873.31         2481.88         391.43              0.0
9999  11574.83     1658.56         1225.17         433.39              0.0

[10000 rows x 55 columns]
```

Dataframe contains 10,000 obversations of 55 features, some qualitative and some quantitative.

0.1.2 Describe the dataset and any issues with it:

[2]:
```
#let's investigate our dataframe
def investigate(data):
    print(data.info())
    print(data.describe())

investigate(df)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 55 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   emp_title                       9167 non-null   object
 1   emp_length                      9183 non-null   float64
 2   state                           10000 non-null  object
 3   homeownership                   10000 non-null  object
 4   annual_income                   10000 non-null  float64
 5   verified_income                 10000 non-null  object
 6   debt_to_income                  9976 non-null   float64
 7   annual_income_joint             1495 non-null   float64
 8   verification_income_joint       1455 non-null   object
 9   debt_to_income_joint            1495 non-null   float64
 10  delinq_2y                       10000 non-null  int64
 11  months_since_last_delinq        4342 non-null   float64
 12  earliest_credit_line            10000 non-null  int64
 13  inquiries_last_12m              10000 non-null  int64
 14  total_credit_lines              10000 non-null  int64
 15  open_credit_lines               10000 non-null  int64
 16  total_credit_limit              10000 non-null  int64
 17  total_credit_utilized           10000 non-null  int64
 18  num_collections_last_12m        10000 non-null  int64
 19  num_historical_failed_to_pay    10000 non-null  int64
 20  months_since_90d_late           2285 non-null   float64
 21  current_accounts_delinq         10000 non-null  int64
 22  total_collection_amount_ever    10000 non-null  int64
 23  current_installment_accounts    10000 non-null  int64
 24  accounts_opened_24m             10000 non-null  int64
```

```
 25  months_since_last_credit_inquiry  8729 non-null   float64
 26  num_satisfactory_accounts         10000 non-null  int64
 27  num_accounts_120d_past_due        9682 non-null   float64
 28  num_accounts_30d_past_due         10000 non-null  int64
 29  num_active_debit_accounts         10000 non-null  int64
 30  total_debit_limit                 10000 non-null  int64
 31  num_total_cc_accounts             10000 non-null  int64
 32  num_open_cc_accounts              10000 non-null  int64
 33  num_cc_carrying_balance           10000 non-null  int64
 34  num_mort_accounts                 10000 non-null  int64
 35  account_never_delinq_percent      10000 non-null  float64
 36  tax_liens                         10000 non-null  int64
 37  public_record_bankrupt            10000 non-null  int64
 38  loan_purpose                      10000 non-null  object
 39  application_type                  10000 non-null  object
 40  loan_amount                       10000 non-null  int64
 41  term                              10000 non-null  int64
 42  interest_rate                     10000 non-null  float64
 43  installment                       10000 non-null  float64
 44  grade                             10000 non-null  object
 45  sub_grade                         10000 non-null  object
 46  issue_month                       10000 non-null  object
 47  loan_status                       10000 non-null  object
 48  initial_listing_status            10000 non-null  object
 49  disbursement_method               10000 non-null  object
 50  balance                           10000 non-null  float64
 51  paid_total                        10000 non-null  float64
 52  paid_principal                    10000 non-null  float64
 53  paid_interest                     10000 non-null  float64
 54  paid_late_fees                    10000 non-null  float64
dtypes: float64(17), int64(25), object(13)
memory usage: 4.2+ MB
None
        emp_length  annual_income  debt_to_income  annual_income_joint  \
count  9183.000000   1.000000e+04     9976.000000         1.495000e+03
mean      5.930306   7.922215e+04       19.308192         1.279146e+05
std       3.703734   6.473429e+04       15.004851         7.016838e+04
min       0.000000   0.000000e+00        0.000000         1.920000e+04
25%       2.000000   4.500000e+04       11.057500         8.683350e+04
50%       6.000000   6.500000e+04       17.570000         1.130000e+05
75%      10.000000   9.500000e+04       25.002500         1.515455e+05
max      10.000000   2.300000e+06      469.090000         1.100000e+06

       debt_to_income_joint    delinq_2y  months_since_last_delinq  \
count           1495.000000  10000.00000               4342.000000
mean              19.979304      0.21600                 36.760709
std                8.054781      0.68366                 21.634939
min                0.320000      0.00000                  1.000000
```

```
25%               14.160000      0.00000                19.000000
50%               19.720000      0.00000                34.000000
75%               25.500000      0.00000                53.000000
max               39.980000     13.00000               118.000000

       earliest_credit_line  inquiries_last_12m  total_credit_lines  …  \
count           10000.00000         10000.00000        10000.000000  …
mean             2001.29000             1.95820           22.679600  …
std                 7.79551             2.38013           11.885439  …
min              1963.00000             0.00000            2.000000  …
25%              1997.00000             0.00000           14.000000  …
50%              2003.00000             1.00000           21.000000  …
75%              2006.00000             3.00000           29.000000  …
max              2015.00000            29.00000           87.000000  …

       public_record_bankrupt   loan_amount          term  interest_rate  \
count            10000.000000  10000.000000  10000.000000   10000.000000
mean                 0.123800  16361.922500     43.272000      12.427524
std                  0.337172  10301.956759     11.029877       5.001105
min                  0.000000   1000.000000     36.000000       5.310000
25%                  0.000000   8000.000000     36.000000       9.430000
50%                  0.000000  14500.000000     36.000000      11.980000
75%                  0.000000  24000.000000     60.000000      15.050000
max                  3.000000  40000.000000     60.000000      30.940000

        installment       balance    paid_total  paid_principal  \
count  10000.000000  10000.000000  10000.000000    10000.000000
mean     476.205323  14458.916610   2494.234773     1894.448466
std      294.851627   9964.561865   3958.230365     3884.407175
min       30.750000      0.000000      0.000000        0.000000
25%      256.040000   6679.065000    928.700000      587.100000
50%      398.420000  12379.495000   1563.300000      984.990000
75%      644.690000  20690.182500   2616.005000     1694.555000
max     1566.590000  40000.000000  41630.443684    40000.000000

       paid_interest  paid_late_fees
count   10000.000000    10000.000000
mean      599.666781        0.119516
std       517.328062        1.813468
min         0.000000        0.000000
25%       221.757500        0.000000
50%       446.140000        0.000000
75%       825.420000        0.000000
max      4216.440000       52.980000

[8 rows x 42 columns]
```

Of the 55 features, 42 are quantitative. The rest are qualitative. We do have some columns with

missing values so we should investigate the number of missing values.

[3]:
```
#check percentage of missing values
def null_values(df):
        mis_val = df.isnull().sum()
        mis_val_percent = 100 * df.isnull().sum() / len(df)
        mis_val_table = pd.concat([mis_val, mis_val_percent], axis=1)
        mis_val_table_ren_columns = mis_val_table.rename(
        columns = {0 : 'Missing Values', 1 : '% of Total Values'})
        mis_val_table_ren_columns = mis_val_table_ren_columns[
            mis_val_table_ren_columns.iloc[:,1] != 0].sort_values(
        '% of Total Values', ascending=False).round(1)
        return mis_val_table_ren_columns

null_values(df)
```

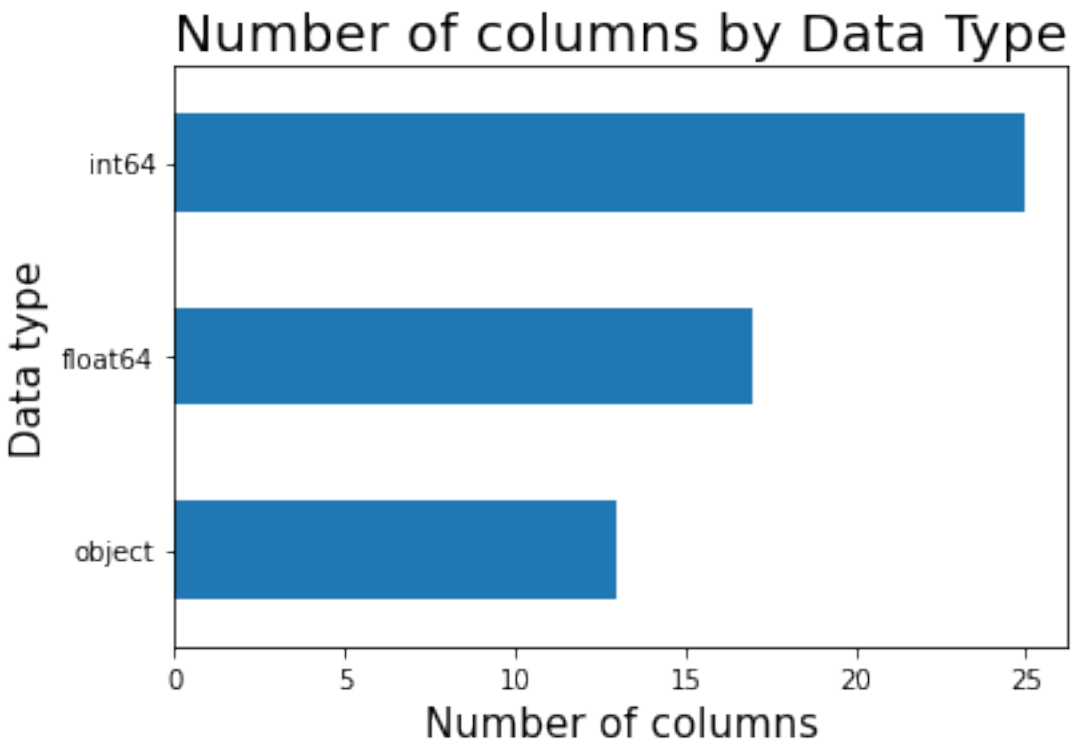
[3]:

	Missing Values	% of Total Values
verification_income_joint	8545	85.4
annual_income_joint	8505	85.0
debt_to_income_joint	8505	85.0
months_since_90d_late	7715	77.2
months_since_last_delinq	5658	56.6
months_since_last_credit_inquiry	1271	12.7
emp_title	833	8.3
emp_length	817	8.2
num_accounts_120d_past_due	318	3.2
debt_to_income	24	0.2

Of the 55 columns, only 10 have missing data. Of these, only a few seem to be do to record keeping mistakes. The majority are due to clients not filing jointly or fitting special circumstances such as being delinquent. Before running any models we will need to address these columns. In the mean time, let's check the distribution of data types.

[4]:
```
df.dtypes.value_counts().sort_values().plot(kind='barh')
plt.title('Number of columns by Data Type',fontsize=20)
plt.xlabel('Number of columns',fontsize=15)
plt.ylabel('Data type',fontsize=15)
```

[4]:
```
Text(0, 0.5, 'Data type')
```



We have quite a few columns that are of an object datatype that will also pose a problem while modeling, unless we convert them using label encoding.

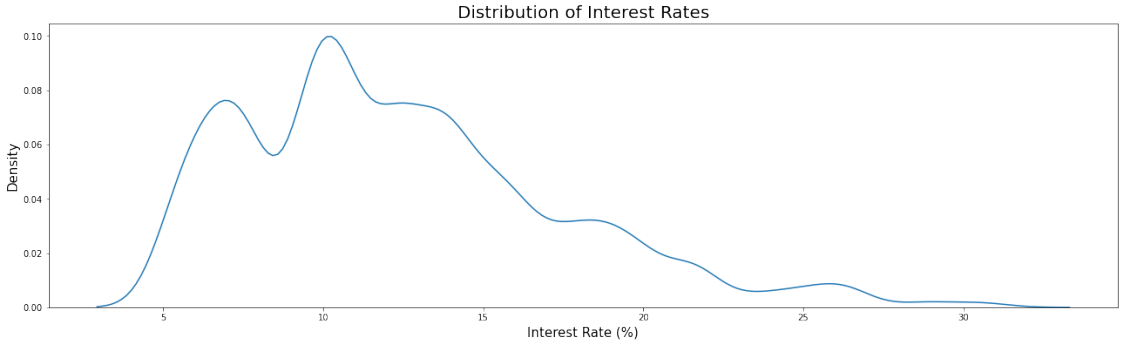
[5]:
```
#check to how many categories for each categorical column
df.select_dtypes('object').apply(pd.Series.nunique, axis = 0)
```

[5]:
```
emp_title                     4741
state                           50
homeownership                    3
verified_income                  3
verification_income_joint        3
loan_purpose                    12
application_type                 2
grade                            7
sub_grade                       32
issue_month                      3
loan_status                      6
initial_listing_status           2
disbursement_method              2
dtype: int64
```

If we were to model with any of these columns, we should label encode the columns having only 2 categorical data and one-hot encode columns with more than 2. We may drop emp_title as there may not be enough unique values to use for modeling.

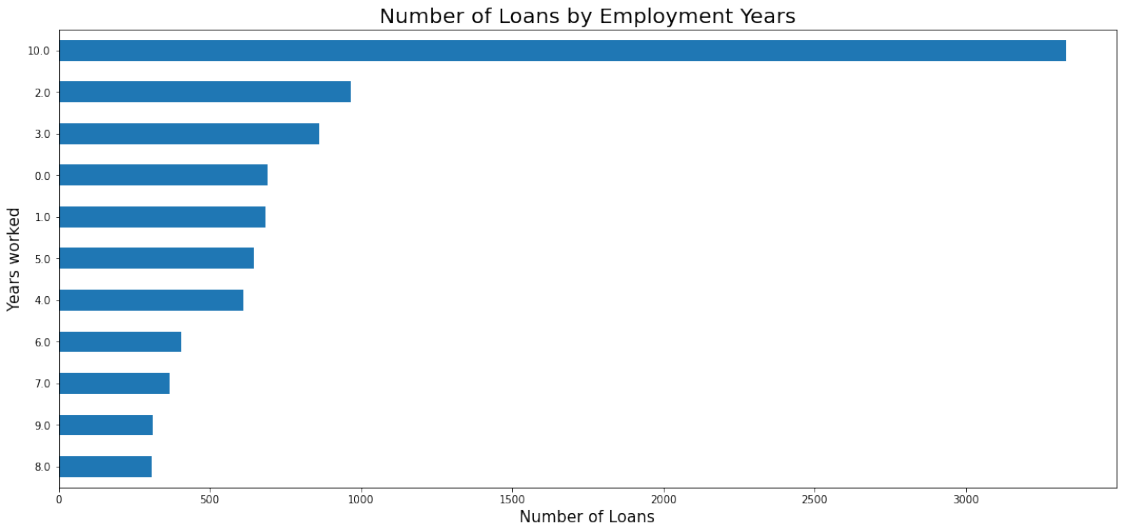
0.1.3 Generate a minimum of 5 unique visualizations using the data and write a brief description of your observations. Additionally, all attempts should be made to make the visualizations visually appealing

[6]:
```
#plot interest rate distribution
fig = plt.figure(figsize=(22,6))
sns.kdeplot(df["interest_rate"], label = 'target = 1')
plt.xlabel('Interest Rate (%)',fontsize=15)
plt.ylabel('Density',fontsize=15)
plt.title('Distribution of Interest Rates',fontsize=20);
plt.show()
```



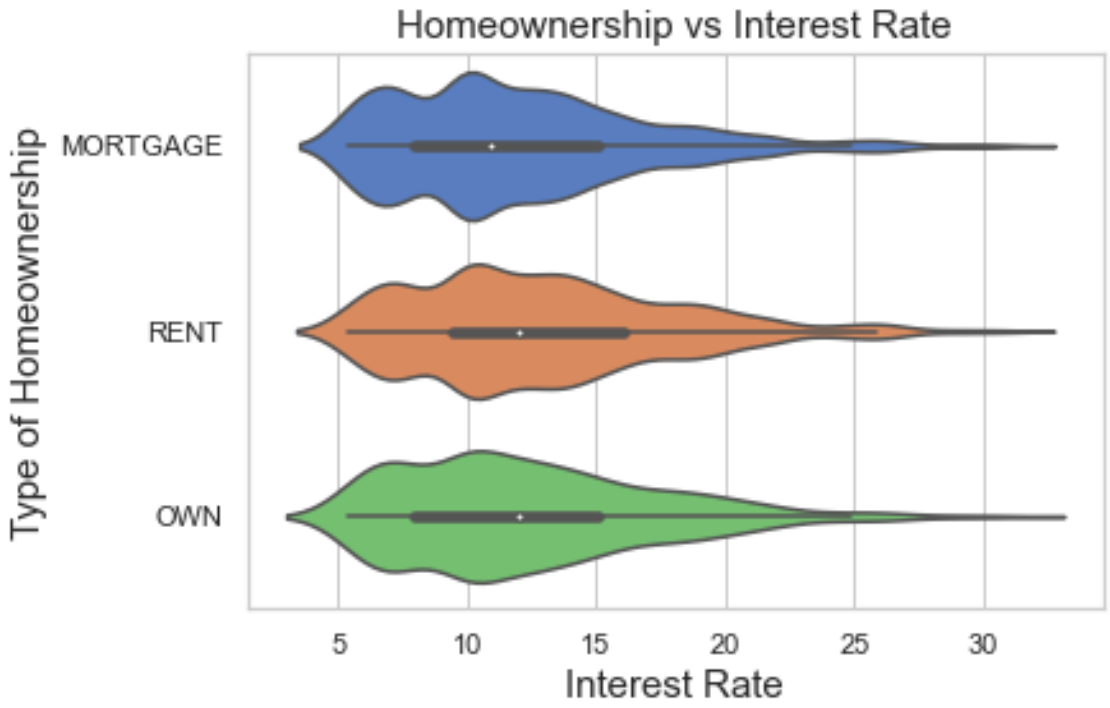
Interest rates seem to be multimodal. The PDF curve suggests that there are several different interest rates curves that are overlapping one another. This may be due to interest rates are defined discretely rather than continously.

[7]:
```
#check number of loans by years employed
df['emp_length'].value_counts().sort_values().plot(kind='barh',figsize=(18,8))
plt.title('Number of Loans by Employment Years',fontsize=20)
plt.xlabel('Number of Loans',fontsize=15)
plt.ylabel('Years worked',fontsize=15)
plt.show()
```



Ten years is the max; meaning employees who have worked longer than 10 years have been consolidated into this category. Without this is mind, one may be tempted to postulate that only those with many years of employment are more likely to get loans. I, however, feel that a correct interpretation would be those who have stable employment of at least 1 year or more are more likely to get a loan, and then this tapers as the years go on.

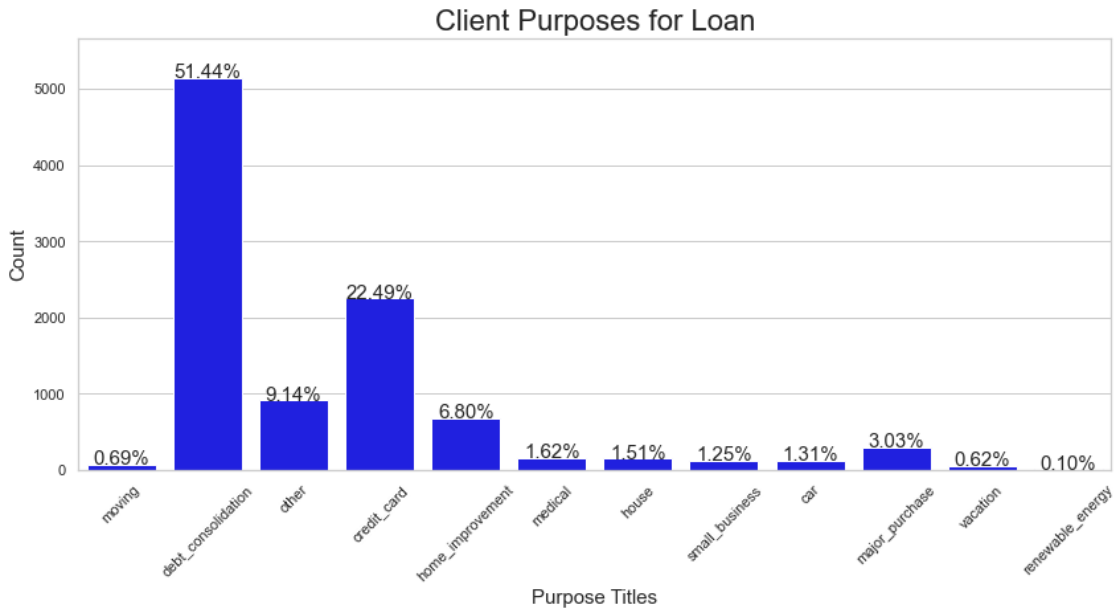
[8]:
```
#violin plots of interest rate by homeownership
sns.set_theme(style = "whitegrid")
ax = sns.violinplot(x = df["interest_rate"], y = df["homeownership"],
                    palette = "muted")
plt.xlabel('Interest Rate',fontsize=15)
plt.ylabel('Type of Homeownership',fontsize=15)
plt.title('Homeownership vs Interest Rate',fontsize=15)
plt.show()
```



The difference between the three types of home ownership is slight, however it seems renters usually have a higher interest rate than either those with mortgages or who own. I would venture to say homeownership has a very small has a slight impact on the interest rate.

[9]:
```
#find major purposes for the loans
plt.figure(figsize=(14,6))

g = sns.countplot(x='loan_purpose', data=df, color='blue')
g.set_title("Client Purposes for Loan", fontsize=22)
g.set_xlabel("Purpose Titles", fontsize=15)
g.set_ylabel('Count', fontsize=15)

sizes=[]

total = len(df)

for p in g.patches:
    height = p.get_height()
    sizes.append(height)
    g.text(p.get_x()+p.get_width()/2.,
            height + 3,
            '{:1.2f}%'.format(height/total*100),
            ha="center", fontsize=15)
```

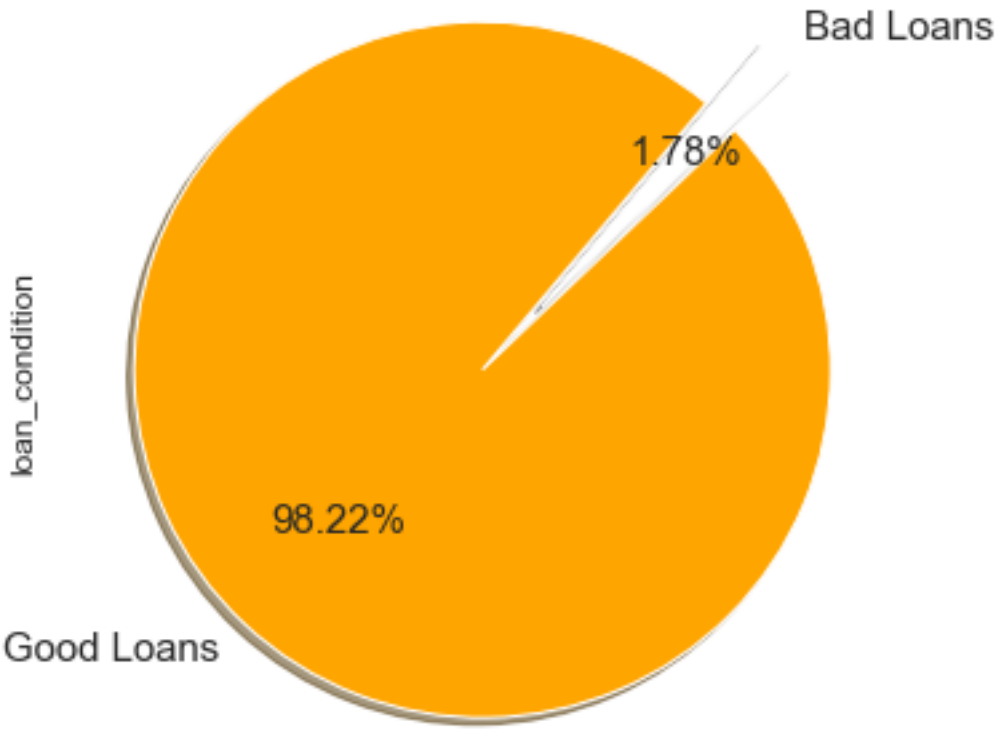
```
g.set_ylim(0, max(sizes) * 1.10)
g.set_xticklabels(g.get_xticklabels(), rotation=45)

plt.show()
```



By far the largest purpose for a loan is for debt consolidation followed distantly by credit card debt. It seems people may turn to this type of debt when other, more traditional, avenues are not available due to their existing debt.

[10]:
```
#determining percentage of bad loans
bad = ["Charged Off", "Default", "Does not meet the credit policy. Status:
 ↪Charged Off",
       "In Grace Period", "Late (16-30 days)", "Late (31-120 days)"]

df['loan_condition'] = np.nan

def loan_condition(status):
    if status in bad:
        return 'Bad Loan'
    else:
        return 'Good Loan'

df['loan_condition'] = df['loan_status'].apply(loan_condition)

colors = ["orange", "white"]
labels ="Good Loans", "Bad Loans"
```

```
plt.figure(figsize=(14,6))

df["loan_condition"].value_counts().plot.pie(explode=[0,0.25],
                                             autopct='%1.2f%%',
                                             shadow=True,
                                             colors=colors,
                                             labels=labels,
                                             fontsize=15,
                                             startangle=50)
plt.show()
```



The vast majority of the loans are in good standing, with a small minority fitting into the one of the "bad" categories.

0.1.4 Create a feature set and create a model which predicts interest rate using at least 2 algorithms. Describe any data cleansing that must be performed and analysis when examining the data.

First let's drop all the columns missing more than 50% of data

[11]:
```
def drop_nan_columns(data, ratio=1.0):
    """
    The ratio parameter (0.0<=ratio<1.0) lets you drop columns which has␣
 ↪'ratio'% of nans.
    (i.e if ratio is 0.8 then all columns with 80% or more entries being nan␣
 ↪get dropped)
    Returns a new dataframe
    """
    col_list = []
    na_df = data.isna()
    total_size = na_df.shape[0]
    for col in na_df:
        a = na_df[col].value_counts()
        if False not in a.keys():
            col_list.append(col)
        elif True not in a.keys():
            pass
        else:
            if a[True]/total_size >= ratio:
                col_list.append(col)
    print(f"{len(col_list)} columns dropped- {col_list}")
    return data.drop(col_list, axis=1)

df = drop_nan_columns(df, ratio=0.50)
```

```
5 columns dropped- ['annual_income_joint', 'verification_income_joint',
'debt_to_income_joint', 'months_since_last_delinq', 'months_since_90d_late']
```

Let's recheck missing values remaining:

[12]:
```
#check percentage of missing values remaining
null_values(df)
```

[12]:

	Missing Values	% of Total Values
months_since_last_credit_inquiry	1271	12.7
emp_title	833	8.3
emp_length	817	8.2
num_accounts_120d_past_due	318	3.2
debt_to_income	24	0.2

Perhaps emp_title is empty for those who are unemployed, so let's check to see if that category even exists.

[13]:
```
unemployed = ['unemployed', 'none', 'Unemployed', 'other', 'Other', "not␣
 ↪working", "no"]
for item in unemployed:
    if item in df['emp_title']:
        print("Found It at ", item)
```

I am going to assume the missing data in emp_title stand for unemployed. Let's replace it with "unemployed." For emp_length, debt_to_income, and months_since_last_credit_inquiry, and num_accounts_120d_past_due we will use mean filling.

[14]:
```
def handle_nans(data):
    """
    Handle the nans induvidually per column
    emp_title: make Nan -> Unemployed
    """
    data['emp_title'] = data['emp_title'].fillna("Unemployed")
    mean_cols = ['emp_length', 'debt_to_income',␣
 ↪'months_since_last_credit_inquiry',
                 "num_accounts_120d_past_due"]
    for col in mean_cols:
        data[col] = data[col].fillna(data[col].mean())
    return

handle_nans(df)
```

[15]:
```
null_values(df)
```

[15]:
```
Empty DataFrame
Columns: [Missing Values, % of Total Values]
Index: []
```

We can confirm we've handled all the missing data. Now let's begin addressing the categorical data.

[16]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 51 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   emp_title                       10000 non-null  object
 1   emp_length                      10000 non-null  float64
 2   state                           10000 non-null  object
 3   homeownership                   10000 non-null  object
 4   annual_income                   10000 non-null  float64
 5   verified_income                 10000 non-null  object
 6   debt_to_income                  10000 non-null  float64
 7   delinq_2y                       10000 non-null  int64
 8   earliest_credit_line            10000 non-null  int64
 9   inquiries_last_12m              10000 non-null  int64
 10  total_credit_lines              10000 non-null  int64
 11  open_credit_lines               10000 non-null  int64
 12  total_credit_limit              10000 non-null  int64
 13  total_credit_utilized           10000 non-null  int64
```

```
 14  num_collections_last_12m          10000 non-null  int64
 15  num_historical_failed_to_pay      10000 non-null  int64
 16  current_accounts_delinq           10000 non-null  int64
 17  total_collection_amount_ever      10000 non-null  int64
 18  current_installment_accounts      10000 non-null  int64
 19  accounts_opened_24m               10000 non-null  int64
 20  months_since_last_credit_inquiry  10000 non-null  float64
 21  num_satisfactory_accounts         10000 non-null  int64
 22  num_accounts_120d_past_due        10000 non-null  float64
 23  num_accounts_30d_past_due         10000 non-null  int64
 24  num_active_debit_accounts         10000 non-null  int64
 25  total_debit_limit                 10000 non-null  int64
 26  num_total_cc_accounts             10000 non-null  int64
 27  num_open_cc_accounts              10000 non-null  int64
 28  num_cc_carrying_balance           10000 non-null  int64
 29  num_mort_accounts                 10000 non-null  int64
 30  account_never_delinq_percent      10000 non-null  float64
 31  tax_liens                         10000 non-null  int64
 32  public_record_bankrupt            10000 non-null  int64
 33  loan_purpose                      10000 non-null  object
 34  application_type                  10000 non-null  object
 35  loan_amount                       10000 non-null  int64
 36  term                              10000 non-null  int64
 37  interest_rate                     10000 non-null  float64
 38  installment                       10000 non-null  float64
 39  grade                             10000 non-null  object
 40  sub_grade                         10000 non-null  object
 41  issue_month                       10000 non-null  object
 42  loan_status                       10000 non-null  object
 43  initial_listing_status            10000 non-null  object
 44  disbursement_method               10000 non-null  object
 45  balance                           10000 non-null  float64
 46  paid_total                        10000 non-null  float64
 47  paid_principal                    10000 non-null  float64
 48  paid_interest                     10000 non-null  float64
 49  paid_late_fees                    10000 non-null  float64
 50  loan_condition                    10000 non-null  object
dtypes: float64(13), int64(25), object(13)
memory usage: 3.9+ MB
```

We must change the following into categorical variables: emp_title, state, homeownership, verified_income, loan_purpose, application_type, grade, sub_grade, issue_month, loan_status, initial_listing_status, and disbursement_method.

[17]:
```
categoricals = ["emp_title", "state", "homeownership", "verified_income",␣
↪"loan_purpose",
                "application_type", "grade", "sub_grade", "issue_month",␣
↪"loan_status",
```

```
                "initial_listing_status", "disbursement_method",␣
↪"loan_condition"]

def handle_types(data, categoricals):
    for category in categoricals:
        try:
            data[category] = data[category].astype('category')
        except:
            pass
    return

handle_types(df, categoricals)
```

[18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 51 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   emp_title                       10000 non-null  category
 1   emp_length                      10000 non-null  float64
 2   state                           10000 non-null  category
 3   homeownership                   10000 non-null  category
 4   annual_income                   10000 non-null  float64
 5   verified_income                 10000 non-null  category
 6   debt_to_income                  10000 non-null  float64
 7   delinq_2y                       10000 non-null  int64
 8   earliest_credit_line            10000 non-null  int64
 9   inquiries_last_12m              10000 non-null  int64
 10  total_credit_lines              10000 non-null  int64
 11  open_credit_lines               10000 non-null  int64
 12  total_credit_limit              10000 non-null  int64
 13  total_credit_utilized           10000 non-null  int64
 14  num_collections_last_12m        10000 non-null  int64
 15  num_historical_failed_to_pay    10000 non-null  int64
 16  current_accounts_delinq         10000 non-null  int64
 17  total_collection_amount_ever    10000 non-null  int64
 18  current_installment_accounts    10000 non-null  int64
 19  accounts_opened_24m             10000 non-null  int64
 20  months_since_last_credit_inquiry 10000 non-null  float64
 21  num_satisfactory_accounts       10000 non-null  int64
 22  num_accounts_120d_past_due      10000 non-null  float64
 23  num_accounts_30d_past_due       10000 non-null  int64
 24  num_active_debit_accounts       10000 non-null  int64
 25  total_debit_limit               10000 non-null  int64
 26  num_total_cc_accounts           10000 non-null  int64
 27  num_open_cc_accounts            10000 non-null  int64
```

```
 28  num_cc_carrying_balance           10000 non-null  int64
 29  num_mort_accounts                 10000 non-null  int64
 30  account_never_delinq_percent      10000 non-null  float64
 31  tax_liens                         10000 non-null  int64
 32  public_record_bankrupt            10000 non-null  int64
 33  loan_purpose                      10000 non-null  category
 34  application_type                  10000 non-null  category
 35  loan_amount                       10000 non-null  int64
 36  term                              10000 non-null  int64
 37  interest_rate                     10000 non-null  float64
 38  installment                       10000 non-null  float64
 39  grade                             10000 non-null  category
 40  sub_grade                         10000 non-null  category
 41  issue_month                       10000 non-null  category
 42  loan_status                       10000 non-null  category
 43  initial_listing_status            10000 non-null  category
 44  disbursement_method               10000 non-null  category
 45  balance                           10000 non-null  float64
 46  paid_total                        10000 non-null  float64
 47  paid_principal                    10000 non-null  float64
 48  paid_interest                     10000 non-null  float64
 49  paid_late_fees                    10000 non-null  float64
 50  loan_condition                    10000 non-null  category
dtypes: category(13), float64(13), int64(25)
memory usage: 3.2 MB
```

We've cleaned up our data! We've handled all the NA's and made sure our datatypes are compatible with our modeling methods. Now let's split the data for testing.

[19]:
```
def split_data(data, column='interest_rate', test_size=0.2):
    from sklearn.model_selection import train_test_split
    target = data[column]
    data.drop(column, axis=1, inplace=True)
    return train_test_split(data, target, test_size=test_size)

X_train, X_test, y_train, y_test = split_data(df)
```

Let's normalize the quantitative data.

[20]:
```
numericals = ["emp_length", "annual_income", "debt_to_income", "debt_to_income",
              "delinq_2y","earliest_credit_line", "inquiries_last_12m",␣
 ↪"total_credit_lines",
              "open_credit_lines", "total_credit_limit",␣
 ↪"total_credit_utilized",
              "num_collections_last_12m", "num_historical_failed_to_pay",
              "current_accounts_delinq","total_collection_amount_ever",
              "current_installment_accounts", "accounts_opened_24m",
              "months_since_last_credit_inquiry","num_satisfactory_accounts",
```

```
                ␣
 ↪"num_accounts_30d_past_due","num_active_debit_accounts","total_debit_limit",
                "num_total_cc_accounts", "num_open_cc_accounts",␣
 ↪"num_cc_carrying_balance",
                "num_mort_accounts", "account_never_delinq_percent","tax_liens",
                "public_record_bankrupt", "loan_amount","term","installment",
                ␣
 ↪"balance",'paid_total','paid_principal','paid_interest','paid_late_fees']

def scale_numerical_data(X_train, X_test, numerical):
    from sklearn.preprocessing import StandardScaler
    sc = StandardScaler()
    X_train[numericals] = sc.fit_transform(X_train[numericals])
    X_test[numericals] = sc.transform(X_test[numericals])
    return

scale_numerical_data(X_train, X_test, numericals)
```

```
/opt/anaconda3/lib/python3.7/site-packages/pandas/core/frame.py:3673:
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
  self[col] = igetitem(value, i)
/opt/anaconda3/lib/python3.7/site-packages/pandas/core/frame.py:3673:
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
  self[col] = igetitem(value, i)
```

Recall for employment type there are so many different categories. This may cause our models to not work properly. So let's only select the top 25 categories and throw the rest into other.

[21]:
```
def shrink_categoricals(X_train, X_test, categoricals, top=25):
    """
    Mutatues categoricals to only keep the entries which are the top
    25 of the daframe otherwise they become other
    """
    for category in categoricals:
        if category not in X_train.columns:
            continue
        tops = X_train[category].value_counts().index[:top]
        def helper(x):
```

```
            if x in tops:
                return x
            else:
                return "Other"
        X_train[category] = X_train[category].apply(helper)
        X_test[category] = X_test[category].apply(helper)
    return

shrink_categoricals(X_train, X_test, categoricals)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:15:
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
  from ipykernel import kernelapp as app
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:16:
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
  app.launch_new_instance()
```

[22]:
```
#so for example we've shrunk employment types to the top 25 jobs
X_train['emp_title'].value_counts()
```

[22]:
```
Other                5917
Unemployed            683
manager               181
owner                 164
teacher               162
driver                 98
sales                  77
registered nurse       66
rn                     66
supervisor             62
truck driver           53
office manager         47
president              42
project manager        41
engineer               38
director               37
general manager        35
operations manager     32
```

```
sales manager           27
machine operator        26
owner                   25
account manager         25
analyst                 24
technician              24
software engineer       24
nurse                   24
Name: emp_title, dtype: int64
```

Next we encode all our categorical data.

[23]:
```
def encode_categorical_data(X_train, X_test, categoricals):
    from sklearn.preprocessing import LabelEncoder
    for category in categoricals:
        if category not in X_train.columns:
            continue
        le = LabelEncoder()
        X_train[category] = le.fit_transform(X_train[category])
        X_test[category] = le.transform(X_test[category])
    return

encode_categorical_data(X_train, X_test, categoricals)
```

```
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:7:
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
  import sys
/opt/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:8:
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
```

Now let's do a dimensionality reduction using PCA. This will make our model more accurate and faster. We will choose n = 95.

[24]:
```
def dimensionality_reduction(X_train, X_test):
    from sklearn.decomposition import PCA
    pca = PCA(n_components=0.95)
    X_train = pca.fit_transform(X_train)
    X_test = pca.transform(X_test)
```

```
    return X_train, X_test

X_train, X_test = dimensionality_reduction(X_train, X_test)
X_train.shape
```

[24]: (8000, 19)

We've reduced the number of columns from 48 to 19. We will now use Random Forest, Linear Regression, KNN and Support Vector Machines to predict our models.

[25]:
```
def random_forest(X_train, y_train, optimal=False):
    """
    Optimal = True returns an untrained model
    """
    from sklearn.ensemble import RandomForestRegressor
    if optimal:
        return RandomForestRegressor(n_estimators=120, max_depth=25,␣
↪bootstrap=True, max_features=3)
    from sklearn.model_selection import GridSearchCV

    param_grid = [{'n_estimators':[60, 70, 80, 100, 120], 'max_depth':[15, 20,␣
↪25, None],
                   'bootstrap':[True, False], 'max_features':[None, 2, 3]}]
    forest = RandomForestRegressor()
    grid_search = GridSearchCV(forest, param_grid, cv=3, scoring="r2")
    grid_search.fit(X_train, y_train)
    final = grid_search.best_params_
    print(final)
    return grid_search.best_estimator_

def regression(X_train, y_train, optimal=False):
    """
    Optimal = True returns an untrained model
    """
    from sklearn.linear_model import ElasticNetCV
    if optimal:
        return ElasticNetCV(alphas=[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
                                    0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],␣
↪l1_ratio=0.0)
    from sklearn.model_selection import GridSearchCV

    elastic_net = ElasticNetCV()
    param_grid = {'alphas':[[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7,
                             0.8, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]],
                  'l1_ratio':[0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9,␣
↪1]}
    grid_search = GridSearchCV(elastic_net, param_grid, scoring="r2", cv=3)
```

```
    grid_search.fit(X_train, y_train)
    print(grid_search.best_params_)
    return grid_search.best_estimator_

def knn(X_train, y_train, optimal=False):
    """
    Optimal = True returns an untrained model
    """
    from sklearn.neighbors import KNeighborsRegressor
    if optimal:
        return KNeighborsRegressor(n_neighbors=10, weights='distance')

    from sklearn.model_selection import GridSearchCV

    model = KNeighborsRegressor()
    param_grid = {'n_neighbors':[2,4,6,8,10,12,14], 'weights':['uniform',␣
↪'distance']}
    grid_search = GridSearchCV(model, param_grid, scoring="r2", cv=3)
    grid_search.fit(X_train, y_train)
    print(grid_search.best_params_)
    return grid_search.best_estimator_

def svm(X_train, y_train, optimal=False):
    """
    Optimal = True returns an untrained model
    """
    from sklearn.svm import SVR
    if optimal:
        return SVR()
    from sklearn.model_selection import RandomizedSearchCV

    svr = SVR()

    param_grid = {'kernel':['rbf', 'sigmoid', 'poly', 'linear'], 'C':[0.8, 1.0,␣
↪1.2]}
    n_iter = 2
    rsv = RandomizedSearchCV(svr, param_grid, n_iter=n_iter, scoring="r2")
    rsv.fit(X_train, y_train)
    final = rsv.best_params_
    print(final)
    return rsv.best_estimator_
```

[26]:
```
order = {0: "rfr", 1:"lin_reg", 2: "knn", 3: "svm"}
model_creators = [random_forest,regression, knn, svm]
model_list = []
models = []
for i, creator in enumerate(model_creators):
```

```
    model_list.append((order[i] , creator(X_train, y_train, optimal=True)))
    models.append(creator(X_train, y_train, optimal=True))
```

[27]:
```
def train_and_test(models, order):
    for i, model in enumerate(models):
        model.fit(X_train, y_train)
    scores = []
    for model in models:
        scores.append(model.score(X_test, y_test))
    final = {}
    for score_no in range(len(scores)):
        final[order[score_no]] = scores[score_no]
    return final
```

[28]:
```
train_and_test(models, order)
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 18467.28878958505, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 17096.671223077035, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 15663.244653363323, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
```

```
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 14162.552118575855, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 12589.684297501311, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 10939.204408068013, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 9205.03980611651, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 7380.29856500701, tolerance: 15.955841280493752
  tol, rng, random, positive)
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 5456.873184663011, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3424.2198515179043, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3214.3732709385713, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3003.241334093672, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
```

```
Duality gap: 2790.7924491062117, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2576.9896728368203, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2361.7887870474906, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2145.1352369924666, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1926.9588157596359, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
```

```
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1707.1631159545907, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1485.5982478698288, tolerance: 15.955841280493752
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 18453.40764537757, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 17074.576267328128, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 15633.12645157838, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
```

```
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 14124.65608534947, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 12544.320765276992, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 10886.761291176554, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 9145.999528490463, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 7315.263694515785, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
```

```
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 5386.616600412515, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3349.847126947084, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3139.680089096133, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2928.2519345666915, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2715.535036635325, tolerance: 16.134928426810937
  tol, rng, random, positive)
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2501.4979622307646, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2286.104471631804, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2069.3121710050746, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1851.0706658500949, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
```

```
Duality gap: 1631.3189808559246, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1409.9818737944565, tolerance: 16.134928426810937
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 18330.454892662165, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 16965.328529579405, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 15537.696601892843, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
```

```
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 14043.135546482208, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 12476.777933084413, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 10833.242345650115, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 9106.535159911535, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 7289.890840414269, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
```

```
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 5375.442022858228, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3353.2497405947615, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3144.5878413287583, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2934.6789196772665, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2723.4974872913317, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
```

```
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2511.0146194886356, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2297.197028542718, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2082.005804422488, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1865.3946717595074, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1647.307525916257, tolerance: 15.881489057435946
  tol, rng, random, positive)
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1427.6748645094585, tolerance: 15.881489057435946
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 18443.169251805302, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 17071.25602753688, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 15636.698670617512, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
```

```
Duality gap: 14135.075339631958, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 12561.521175321648, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 10910.657864750616, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 9176.494200922169, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 7352.26227592338, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
```

```
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 5430.072629782402, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3399.875844108557, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3190.3854596410265, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2979.6395376509768, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2767.611013190295, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
```

```
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2554.2689608699434, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2339.5775227366694, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2123.4944406102754, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1905.969007752447, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1686.9391464672, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
```

```
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1466.3271315127233, tolerance: 15.999813875685936
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 18414.333444929387, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 17043.313399691015, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 15609.723569465059, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 14109.147971256447, tolerance: 15.99497388777344
  tol, rng, random, positive)
```

```
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 12536.727506971454, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 10887.088964081193, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 9154.24446104491, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 7331.425512502228, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
```

```
Duality gap: 5410.734920660107, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3382.1139027230497, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 3172.786821752142, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2962.206131103424, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2750.34533706865, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
```

```
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2537.17432703147, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2322.6584036321656, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2106.756976607539, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1889.4217574812246, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1670.594211158947, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: UserWarning:
Coordinate descent with alpha=0 may lead to unexpected results and is
discouraged.
```

```
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:528: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 1450.201845296787, tolerance: 15.99497388777344
  tol, rng, random, positive)
/opt/anaconda3/lib/python3.7/site-
packages/sklearn/linear_model/_coordinate_descent.py:532: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 2086.7116545010135, tolerance: 19.99230590318001
  positive)
```

[28]:
```
{'rfr': 0.9476712778186067,
 'lin_reg': 0.9806461919795219,
 'knn': 0.9793348594702171,
 'svm': 0.9862192825260014}
```

Our best model are support vector machines, followed by linear regression, knn and random forests. Now let's look at best models accuracy to get a sense of how close we are to the true interest rate.

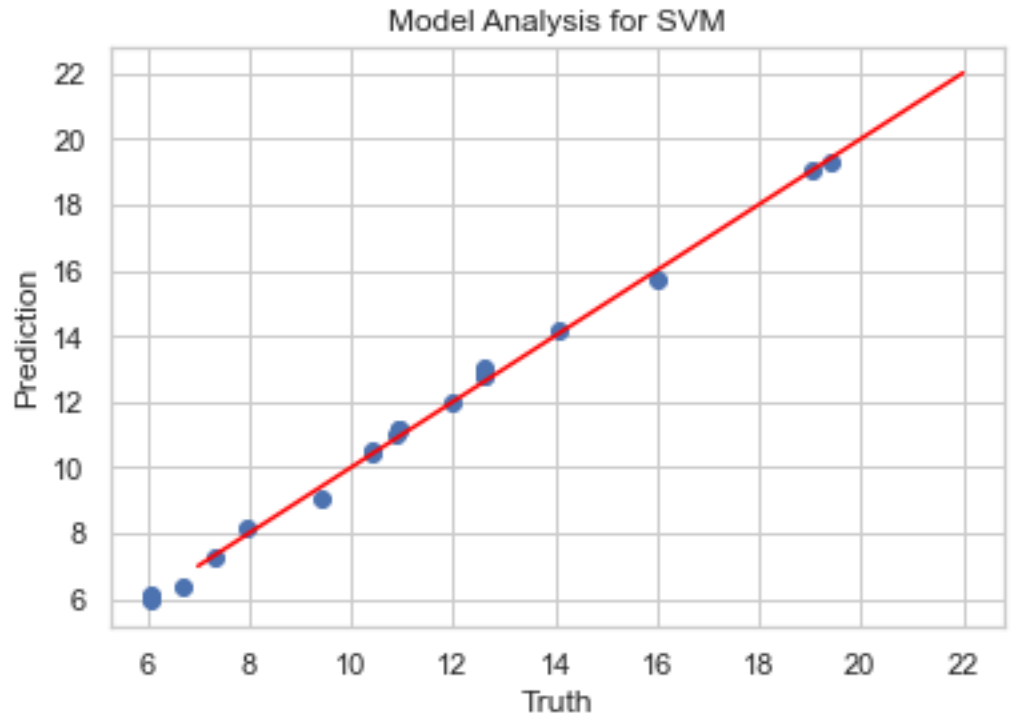
[29]:
```
best_model = models[3]
# We take the first 20 inputs and compare the predictions with the outputs
truths = y_test[0:20]
preds = best_model.predict(X_test[0:20])
residual_error = truths - preds
print(residual_error)
```

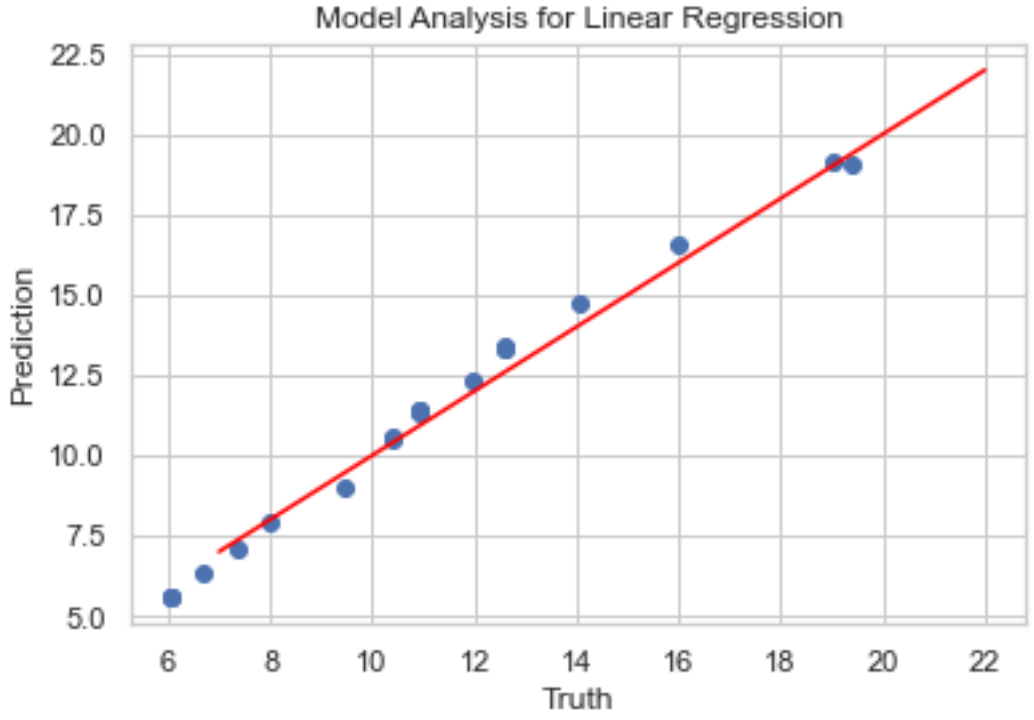
```
1489   -0.013951
2230   -0.265830
927    -0.013168
5425   -0.109158
1910   -0.165216
755    -0.420203
8170   -0.193551
6133   -0.296098
4454    0.077300
2347    0.036593
6411    0.142490
5895    0.263502
3685   -0.227392
3965   -0.065930
9239   -0.071229
4862    0.309456
7880    0.404903
73      0.138523
9795    0.043424
4845   -0.135115
Name: interest_rate, dtype: float64
```

0.1.5 Visualize the test results and propose enhancements to the model, what would you do if you had more time. Also describe assumptions you made and your approach.

Let's visualize our test results for our two best predictors: SVM and linear regression.

[30]:
```
best_model = models[3]
truths = y_test[0:20]
preds = best_model.predict(X_test[0:20])
residual_error = truths - preds

plt.scatter(truths, preds)
plt.plot([7.0, 22], [7.0, 22], c = "red")
plt.title("Model Analysis for SVM")
plt.xlabel("Truth")
plt.ylabel("Prediction")
plt.show()

best_model = models[1]
truths = y_test[0:20]
preds = best_model.predict(X_test[0:20])
residual_error = truths - preds

plt.scatter(truths, preds)
plt.plot([7.0, 22], [7.0, 22], c = "red")
plt.title("Model Analysis for Linear Regression")
plt.xlabel("Truth")
plt.ylabel("Prediction")
plt.show()
```


Model Analysis for SVM
Prediction
Truth
22
20
18
16
14
12
10
8
6
6
8
10
12
14
16
18
20
22


Model Analysis for Linear Regression
Prediction
Truth
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
6
8
10
12
14
16
18
20
22

Had I more time, would try to use a Boosting technique -Gradient Boosting and grid search. We made assumptions when replaced the null values of employment_title with unemployed, and when we filled the other null values with the mean. When selecting the linear regression model we assumed that data is linearly separatable and the features are not correlated.