Business Understanding

In an effort to make educational investments less speculative, the US Department of Education has matched information from the student financial aid system with federal tax returns to create the College Scorecard dataset.

This dataset contains a wide range of college features including academics, admissions, student demographics, cost, financial aid, completion %'s, repayment, and earnings.

This project looks at the 'Percent completed within 4 years at original institution' [COMP_ORIG_YR4_RT] and see if there is a correlation with a combination of selected college features from the dataset.

Data Understanding

The dataset is large and contains data ranging back from 1998. The dataset is limited to the last five years to reduce the running time for the code below. For every column, 'PrivacySuppressed' values were converted to NaN values and the median value grouped by the instutition's ID replaced NaN values.

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import sqlite3
import matplotlib.pyplot as plt
%matplotlib inline
pd.options.display.max_columns = None
pd.options.display.max_rows = None
from sklearn.dummy import DummyClassifier, DummyRegressor
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import KNNImputer
np.random.seed(9)
```

In [2]:
```
# Connect to SQLite file
conn = sqlite3.connect('data/database.sqlite')
cur = conn.cursor()
```

In [3]:
```
# Create connection and check table names
cur.execute("""SELECT name FROM sqlite_master WHERE type = 'table';""")

table_names = cur.fetchall()
table_names
```

Out[3]:
```
[('Scorecard',)]
```

In [4]:
```
# Examine columns

cur.execute("""SELECT sql FROM sqlite_master WHERE type = 'table' AND name =
scorecard_columns = cur.fetchone()
```

In [5]:
```
# Create a dataframe of currently operating schools that offer
# Predominantly bachelor's and associate's degrees granting
# institutions between the year 2007 and 2012

df = pd.read_sql("""
SELECT *
FROM Scorecard
WHERE year BETWEEN 2007 AND 2012
AND CURROPER = "Currently certified as operating"
AND PREDDEG in ("Predominantly bachelor's-degree granting",
                "Predominantly associate's-degree granting")
;
""", conn, index_col='Id')
```

In [6]:
```
# Confirm SQL query year range

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

Out[6]:
```
array([2007, 2008, 2009, 2010, 2011, 2012], dtype=int64)
```

In [7]:
```
df.shape
```

Out[7]:
```
(19509, 1730)
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19509 entries, 73000 to 116507
Columns: 1730 entries, UNITID to Year
dtypes: float64(191), int64(193), object(1346)
memory usage: 257.6+ MB
```

In [9]:
```
# df.head(5)
```

In [10]:
```
# Convert all 'PrivacySuppressed' values to NaN values
df = df.replace('PrivacySuppressed', np.nan, regex=True) # All data frame
```

[COMP_ORIG_YR4_RT] - Target Variable

The target variable contains NaN's which must be handled with. Upon inspection, many of the colleges with NaN's were community or religious training colleges. The NaN's make up less than 5% so those rows were dropped.

In [11]:
```
explore1 = df[df['COMP_ORIG_YR4_RT'].isnull()]
explore1['INSTNM'].value_counts()
```

```
Porterville College                                3
Denver School of Nursing                           3
Palo Alto University                               3
Midwest College of Oriental Medicine-Chicago       3
Haskell Indian Nations University                  3
University of Puerto Rico-Aguadilla                3
Mid-South Community College                        3
Contra Costa College                               3
Saint Luke's College of Health Sciences            3
College of the Mainland                            3
Los Medanos College                                3
Guam Community College                             3
Touro College Los Angeles                          2
Centralia College                                  2
University of Puerto Rico-Mayaguez                 2
Lone Star College System                           2
Bunker Hill Community College                      2
Bethesda University of California                  2
Colegio Pentecostal Mizpa                          2
Rich Mountain Community College                    2
```

In [ ]:
```

```

In [12]:
```
# Checking the NaN Count and percent
X_NaN = df['COMP_ORIG_YR4_RT'].isna().sum()
X_len = len(df['COMP_ORIG_YR4_RT'])
X_NaN, X_len, X_NaN / X_len
```

Out[12]: (1276, 19509, 0.0654057101850428)

In [13]:
```
# Create new dataframe that drops NaN rows bc the percent is low enough
processed_df = df.dropna(subset=['COMP_ORIG_YR4_RT']).copy()
processed_df.shape
```

Out[13]: (18233, 1730)

Feature Variables

Many of the feature variables also contain NaN's values in their columns. In order to narrow the search for a strong correlation, columns will be dropped if they contain only NaN's. For other columns that have data for other years, the median for that college will replace the NaN values. Other columns that have outdated or duplicate data will be dropped as well. Finally, any remaining NaN values will be filled in with a KNN imputer. This process will be done for each feature category: school, admissions, academics, student, cost, aid, completion, repayment, and earning features.

In [14]:
```
# look at columns w NaN's
processed_df.isna().sum()
```

Out[14]:
```
UNITID                  0
OPEID                   0
opeid6                  0
INSTNM                  0
CITY                    0
STABBR                  0
ZIP                     0
AccredAgency        18233
INSTURL             18233
NPCURL              18233
sch_deg                 0
HCM2                    0
main                    0
NUMBRANCH               0
PREDDEG                 0
HIGHDEG                 0
CONTROL                 0
st_fips                 0
region                  0
```

In [15]:
```
# remove columns with all NaN's
perc = 100.0 # Like N %
min_count =  int(((100-perc)/100)*df.shape[0] + 1)
processed_df = processed_df.dropna( axis=1, thresh=min_count)
```

In [16]:
```
# Recheck dataset size after dropping 100% NaN columns
processed_df.shape
```

Out[16]:
```
(18233, 1626)
```

In [17]:
```
# Inspect other columns with NaN's and
processed_df.isna().sum()
```

```
PCIP26                  0
PCIP27                  0
PCIP29                  0
PCIP30                  0
PCIP31                  0
PCIP38                  0
PCIP39                  0
PCIP40                  0
PCIP41                  0
PCIP42                  0
PCIP43                  0
PCIP44                  0
PCIP45                  0
PCIP46                  0
PCIP47                  0
PCIP48                  0
PCIP49                  0
PCIP50                  0
```

In [20]:
```
irrelevant_columns = ['OPEID','opeid6', 'sch_deg', 'HCM2', 'UGDS_AIANOld', '

# processed_df = processed_df.drop(irrelevant_columns, axis=1)
processed_df.shape
```

Out[20]: (18233, 1616)

School Features

In [199]:
```
# create list of general college info
school_features = ['UNITID', 'INSTNM', 'CITY', 'STABBR', 'ZIP', 'main', 'NUM
school_df = processed_df[school_features].copy()
```

In [201]:
```
school_df.isna().sum()
```

Out[201]:
```
UNITID              0
INSTNM              0
CITY                0
STABBR              0
ZIP                 0
main                0
NUMBRANCH           0
PREDDEG             0
HIGHDEG             0
CONTROL             0
st_fips             0
region              0
Year                0
COMP_ORIG_YR4_RT    0
dtype: int64
```

In [310]:
```
1 school_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 18233 entries, 73000 to 116507
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype 
---  ------            --------------  ----- 
 0   UNITID            18233 non-null  int64 
 1   INSTNM            18233 non-null  object
 2   CITY              18233 non-null  object
 3   STABBR            18233 non-null  object
 4   ZIP               18233 non-null  object
 5   main              18233 non-null  object
 6   NUMBRANCH         18233 non-null  int64 
 7   PREDDEG           18233 non-null  object
 8   HIGHDEG           18233 non-null  object
 9   CONTROL           18233 non-null  object
 10  st_fips           18233 non-null  object
 11  region            18233 non-null  object
 12  Year              18233 non-null  int64 
 13  COMP_ORIG_YR4_RT  18233 non-null  int32 
dtypes: int32(1), int64(3), object(10)
memory usage: 2.0+ MB
```

In [315]:
```
1 school_cont_cols = ['UNITID', 'COMP_ORIG_YR4_RT']
```

In [318]:
```
1 school_cont_df = school_df[school_cont_cols]
```

In [423]:
```
1 school_cat_cols = ['CITY', 'STABBR', 'ZIP', 'main', 'NUMBRANCH', 'PREDDEG',
```

In [424]:
```
1 school_dummies = pd.get_dummies(school_df[school_cat_cols], drop_first=True)
2 school_dummies.shape
```

Out[424]: (18233, 5637)

In [ ]:
```
1
```

In [425]:
```
1 updated_school_df = school_cont_df.join(school_dummies)
2 updated_school_df.shape
```

Out[425]: (18233, 5639)

In [ ]:
```
1
```

School Features - Correlation (dummied)

In [426]:
```
1 updated_school_df['COMP_ORIG_YR4_RT'] = updated_school_df['COMP_ORIG_YR4_RT'
```

In [433]:
```
corr_matrix = updated_school_df.corr()
school_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
school_top5
```

Out[433]:
```
COMP_ORIG_YR4_RT    1.000000
ZIP_60616-3878      0.408080
ZIP_97230-3099      0.333187
ZIP_10027-4649      0.235592
CITY_Needham        0.235592
ZIP_02492-1200      0.235592
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [434]:
```
school_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).index[
school_top5_list
```

Out[434]:
```
['ZIP_60616-3878',
 'ZIP_97230-3099',
 'ZIP_10027-4649',
 'CITY_Needham',
 'ZIP_02492-1200']
```

Original School Features - Correlation

In [429]:
```
school_df['COMP_ORIG_YR4_RT'] = school_df['COMP_ORIG_YR4_RT'].astype(int)
```

In [430]:
```
corr_matrix = school_df.corr()
org_school_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
org_school_top5
```

Out[430]:
```
COMP_ORIG_YR4_RT    1.000000
UNITID               0.003471
Year                -0.005795
NUMBRANCH           -0.007335
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [ ]:
```

```

Admissions Features

In [159]:
```
target = ['UNITID','COMP_ORIG_YR4_RT']
```

In [30]:
```
admission_features = ['ADM_RATE', 'ADM_RATE_ALL', 'SATVR25', 'SATVR75', 'SAT
```

In [31]:
```
target_admission = target + admission_features
admission_df = processed_df[target_admission].copy()
admission_df.isna().sum()
```

Out[31]:
```
UNITID                 0
COMP_ORIG_YR4_RT       0
ADM_RATE            7277
ADM_RATE_ALL        6720
SATVR25            10844
SATVR75            10844
SATMT25            10755
SATMT75            10755
SATWR25            14112
SATWR75            14112
SATVRMID           10844
SATMTMID           10755
SATWRMID           14112
ACTCM25            10584
ACTCM75            10584
ACTEN25            11738
ACTEN75            11738
ACTMT25            11744
ACTMT75            11744
ACTWR25            17137
ACTWR75            17138
ACTCMMID           10584
ACTENMID           11738
ACTMTMID           11744
ACTWRMID           17138
SAT_AVG             9970
SAT_AVG_ALL         9645
dtype: int64
```

In [32]:
```
for i in admission_features:
    admission_df[i] = admission_df.groupby(['UNITID'], sort=False)[i].apply(
```

```
C:\Users\Jesus Baquiax\anaconda3\envs\learn-env\lib\site-packages\numpy\lib\nan
functions.py:1117: RuntimeWarning: Mean of empty slice
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
```

In [33]:
```
admission_df.isna().sum()
```

Out[33]:
```
UNITID                 0
COMP_ORIG_YR4_RT       0
ADM_RATE            6450
ADM_RATE_ALL        5937
SATVR25             9941
SATVR75             9941
SATMT25             9899
SATMT75             9899
SATWR25            12135
SATWR75            12135
SATVRMID            9941
SATMTMID            9899
SATWRMID           12135
ACTCM25             9772
ACTCM75             9772
ACTEN25            10737
ACTEN75            10737
ACTMT25            10749
ACTMT75            10749
ACTWR25            15206
ACTWR75            15206
ACTCMMID            9772
ACTENMID           10737
ACTMTMID           10749
ACTWRMID           15206
SAT_AVG             9377
SAT_AVG_ALL         8990
dtype: int64
```

In [74]:
```
imputer = KNNImputer(n_neighbors=2, weights="uniform")
admissions_KNN = pd.DataFrame(imputer.fit_transform(admission_df), columns =
```

Admission Features - Correlation

In [88]:
```
admissions_KNN['COMP_ORIG_YR4_RT'] = admissions_KNN['COMP_ORIG_YR4_RT'].asty
```

In [326]:
```
corr_matrix = admissions_KNN.corr()
admission_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
admission_top5
```

Out[326]:
```
COMP_ORIG_YR4_RT    1.000000
SATMTMID            0.025243
SATMT25             0.024867
SAT_AVG_ALL         0.024282
SATMT75             0.023885
ACTEN25             0.023854
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [209]:
```
admission_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).ind
admission_top5_list
```

Out[209]:
```
['SATMTMID', 'SATMT25', 'SAT_AVG_ALL', 'SATMT75', 'ACTEN25']
```

Academics Features

The academic features contain two types of categories:

1) the percentage of degrees awarded in a field of study

2) Whether the program is offered at the institution

Academics (Program %)

In [101]:
```
program_percentage = ['PCIP01', 'PCIP03', 'PCIP04', 'PCIP05', 'PCIP09', 'PCI
```

In [215]:
```
target_percentage =  target + program_percentage
percentage_df = processed_df[target_percentage].copy()
percentage_df.isna().sum()
```

Out[215]:
```
UNITID              0
COMP_ORIG_YR4_RT    0
PCIP01              0
PCIP03              0
PCIP04              0
PCIP05              0
PCIP09              0
PCIP10              0
PCIP11              0
PCIP12              0
PCIP13              0
PCIP14              0
PCIP15              0
PCIP16              0
PCIP19              0
PCIP22              0
PCIP23              0
PCIP24              0
PCIP25              0
PCIP26              0
PCIP27              0
PCIP29              0
PCIP30              0
PCIP31              0
PCIP38              0
PCIP39              0
PCIP40              0
PCIP41              0
PCIP42              0
PCIP43              0
PCIP44              0
PCIP45              0
PCIP46              0
PCIP47              0
PCIP48              0
PCIP49              0
PCIP50              0
PCIP51              0
PCIP52              0
PCIP54              0
dtype: int64
```

Academics (Program %) - Correlation

In [328]:
```
corr_matrix = percentage_df.corr()
percentage_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
percentage_top5
```

Out[328]:
```
COMP_ORIG_YR4_RT    1.000000
PCIP45              0.385909
PCIP23              0.384487
PCIP54              0.381086
PCIP16              0.319399
PCIP26              0.316918
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [211]:
```
percentage_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).in
percentage_top5_list
```

Out[211]:
```
['PCIP45', 'PCIP23', 'PCIP54', 'PCIP16', 'PCIP26']
```

Academics (Program Offered)

In [102]:
```
program_offered = ['CIP01CERT1', 'CIP01CERT2', 'CIP01ASSOC', 'CIP01CERT4', '
```

In [214]:
```
target_offered = target + program_offered
offered_df = processed_df[target_offered].copy()
offered_df.isna().sum()
```

Out[214]:
```
UNITID              0
COMP_ORIG_YR4_RT    0
CIP01CERT1          0
CIP01CERT2          0
CIP01ASSOC          0
CIP01CERT4          0
CIP01BACHL          0
CIP03CERT1          0
CIP03CERT2          0
CIP03ASSOC          0
CIP03CERT4          0
CIP03BACHL          0
CIP04CERT1          0
CIP04CERT2          0
CIP04ASSOC          0
CIP04CERT4          0
CIP04BACHL          0
CIP05CERT1          0
CIP05CERT2          0
```

In [279]:
```
offered_df = offered_df.astype(str)
```

In [281]:
```
1 offered_df['CIP01CERT4'].value_counts()
```

Out[281]:
```
0    17860
1      371
2        2
Name: CIP01CERT4, dtype: int64
```

In [282]:
```
1 dummies = pd.get_dummies(offered_df[program_offered], drop_first=True)
```

Academics (Program Offered) - Correlation

In [302]:
```
1 dummies.shape
```

Out[302]:
```
(18233, 380)
```

In [303]:
```
1 dummies_df = dummies.join(offered_df[target])
```

In [304]:
```
1 dummies_df.shape
```

Out[304]:
```
(18233, 382)
```

In [306]:
```
1 dummies_df['COMP_ORIG_YR4_RT'] = dummies_df['COMP_ORIG_YR4_RT'].astype(float
```

In [307]:
```
1 dummies_df['COMP_ORIG_YR4_RT'] = dummies_df['COMP_ORIG_YR4_RT'].astype(int)
```

In [308]:
```
1 corr_matrix = dummies_df.corr()
2 offered_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
3 offered_top5
```

Out[308]:
```
COMP_ORIG_YR4_RT    1.000000
CIP51BACHL_1        0.023110
CIP26BACHL_1        0.008103
CIP39CERT4_2       -0.000233
CIP25CERT4_2       -0.000233
CIP29CERT4_2       -0.000233
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [309]:
```
1 offered_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).index
2 offered_top5_list
```

Out[309]:
```
['CIP51BACHL_1',
 'CIP26BACHL_1',
 'CIP39CERT4_2',
 'CIP25CERT4_2',
 'CIP29CERT4_2']
```

Student Features

In [216]:
```
student = ['UGDS', 'UGDS_WHITE', 'UGDS_BLACK', 'UGDS_HISP', 'UGDS_ASIAN', 'U
```

In [220]:
```
target_student = target + student
student_df = processed_df[target_student].copy()
student_df.isna().sum()
```

Out[220]:
```
UNITID                0
COMP_ORIG_YR4_RT      0
UGDS                 10
UGDS_WHITE         2859
UGDS_BLACK         2859
UGDS_HISP          2859
UGDS_ASIAN         2859
UGDS_AIAN          2859
UGDS_NHPI          2859
UGDS_2MOR          2859
UGDS_NRA             10
UGDS_UNKN            10
UGDS_WHITENH       9574
UGDS_BLACKNH       9574
UGDS_API           9574
dtype: int64
```

In [221]:
```
for i in student:
    student_df[i] = student_df.groupby(['UNITID'], sort=False)[i].apply(lamb
```

In [222]:
```
student_df.isna().sum()
```

Out[222]:
```
UNITID                0
COMP_ORIG_YR4_RT      0
UGDS                  9
UGDS_WHITE           50
UGDS_BLACK           50
UGDS_HISP            50
UGDS_ASIAN           50
UGDS_AIAN            50
UGDS_NHPI            50
UGDS_2MOR            50
UGDS_NRA              9
UGDS_UNKN             9
UGDS_WHITENH        977
UGDS_BLACKNH        977
UGDS_API            977
dtype: int64
```

In [223]:
```
imputer2 = KNNImputer(n_neighbors=2, weights="uniform")
student_KNN = pd.DataFrame(imputer2.fit_transform(student_df), columns = lis
```

In [225]:
```
1 student_KNN.isna().sum()
```

Out[225]:
```
UNITID              0
COMP_ORIG_YR4_RT    0
UGDS                0
UGDS_WHITE          0
UGDS_BLACK          0
UGDS_HISP           0
UGDS_ASIAN          0
UGDS_AIAN           0
UGDS_NHPI           0
UGDS_2MOR           0
UGDS_NRA            0
UGDS_UNKN           0
UGDS_WHITENH        0
UGDS_BLACKNH        0
UGDS_API            0
dtype: int64
```

Student Features - Correlation

In [226]:
```
1 student_KNN['COMP_ORIG_YR4_RT'] = student_KNN['COMP_ORIG_YR4_RT'].astype(int
```

In [227]:
```
1 corr_matrix = student_KNN.corr()
2 student_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
3 student_top5
```

Out[227]:
```
COMP_ORIG_YR4_RT    1.000000
UGDS_NRA            0.063676
UGDS_ASIAN          0.027090
UGDS_UNKN           0.016609
UGDS_API            0.010668
UGDS                0.004897
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [228]:
```
1 student_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).index
2 student_top5_list
```

Out[228]:
```
['UGDS_NRA', 'UGDS_ASIAN', 'UGDS_UNKN', 'UGDS_API', 'UGDS']
```

Cost Features

In [229]:
```
cost_list = ['NPT4_PUB', 'NPT4_PRIV', 'NPT4_PROG', 'NPT4_OTHER', 'NPT41_PUB'
target_cost = target + cost_list
cost_df = processed_df[target_cost].copy()
cost_df.isna().sum()
```

Out[229]:
```
UNITID                 0
COMP_ORIG_YR4_RT       0
NPT4_PUB           13317
NPT4_PRIV          11284
NPT4_PROG          18149
NPT4_OTHER         18040
NPT41_PUB          13318
NPT42_PUB          13584
NPT43_PUB          13673
NPT44_PUB          13932
NPT45_PUB          14300
NPT41_PRIV         11356
NPT42_PRIV         11636
NPT43_PRIV         11862
NPT44_PRIV         12424
NPT45_PRIV         13025
NPT41_PROG         18149
NPT42_PROG         18149
NPT43_PROG         18149
NPT44_PROG         18150
NPT45_PROG         18150
NPT41_OTHER        18041
NPT42_OTHER        18059
NPT43_OTHER        18075
NPT44_OTHER        18101
NPT45_OTHER        18119
NPT4_048_PUB       13317
NPT4_048_PRIV      11314
NPT4_048_PROG      18149
NPT4_048_OTHER     18040
NPT4_3075_PUB      13572
NPT4_3075_PRIV     11531
NPT4_75UP_PUB      13911
NPT4_75UP_PRIV     12306
NPT4_3075_PROG     18151
NPT4_3075_OTHER    18054
NPT4_75UP_PROG     18178
NPT4_75UP_OTHER    18096
NUM4_PUB           13307
NUM4_PRIV          11258
NUM4_PROG          18147
NUM4_OTHER         18038
NUM41_PUB          13313
NUM42_PUB          13313
NUM43_PUB          13313
NUM44_PUB          13313
NUM45_PUB          13313
NUM41_PRIV         11285
NUM42_PRIV         11285
NUM43_PRIV         11285
NUM44_PRIV         11285
```

```
NUM45_PRIV          11285
NUM41_PROG          18149
NUM42_PROG          18149
NUM43_PROG          18149
NUM44_PROG          18149
NUM45_PROG          18149
NUM41_OTHER         18040
NUM42_OTHER         18040
NUM43_OTHER         18040
NUM44_OTHER         18040
NUM45_OTHER         18040
COSTT4_A             6259
COSTT4_P            18075
TUITIONFEE_IN         776
TUITIONFEE_OUT        776
TUITIONFEE_PROG     17969
dtype: int64
```

In [230]:
```
for i in cost_list:
    cost_df[i] = cost_df.groupby(['UNITID'], sort=False)[i].apply(lambda x:
```

```
C:\Users\Jesus Baquiax\anaconda3\envs\learn-env\lib\site-packages\numpy\lib\nan
functions.py:1117: RuntimeWarning: Mean of empty slice
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
```

In [231]:
```
cost_df.isna().sum()
```

Out[231]:
```
UNITID                  0
COMP_ORIG_YR4_RT        0
NPT4_PUB            10915
NPT4_PRIV            7865
NPT4_PROG           17977
NPT4_OTHER          17722
NPT41_PUB           10921
NPT42_PUB           11174
NPT43_PUB           11297
NPT44_PUB           11450
NPT45_PUB           11792
NPT41_PRIV           7903
NPT42_PRIV           8058
NPT43_PRIV           8272
NPT44_PRIV           8770
NPT45_PRIV           9461
NPT41_PROG          17977
NPT42_PROG          17977
NPT43_PROG          17977
```

In [337]:
```
imputer3 = KNNImputer(n_neighbors=2, weights="uniform")
cost_KNN = pd.DataFrame(imputer3.fit_transform(cost_df), columns = list(cost
```

Cost Features - Correlation

In [339]:
```
corr_matrix = cost_KNN.corr()
cost_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
cost_top5
```

Out[339]:
```
COMP_ORIG_YR4_RT    1.000000
TUITIONFEE_OUT      0.670658
COSTT4_A            0.626523
TUITIONFEE_IN       0.621967
NPT45_PUB           0.579966
NPT4_75UP_PUB       0.573320
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [340]:
```
cost_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).index[1:
cost_top5_list
```

Out[340]:
```
['TUITIONFEE_OUT', 'COSTT4_A', 'TUITIONFEE_IN', 'NPT45_PUB', 'NPT4_75UP_PUB']
```

Aid Features

In [160]:
```
aid_features = ['PCTPELL', 'PCTFLOAN', 'DEBT_MDN', 'GRAD_DEBT_MDN', 'WDRAW_D
target_aid = target + aid_features
aid_df = processed_df[target_aid].copy()
aid_df.isna().sum()
```

Out[160]:
```
UNITID                      0
COMP_ORIG_YR4_RT            0
PCTPELL                  2866
PCTFLOAN                 5732
DEBT_MDN                  605
GRAD_DEBT_MDN            1165
WDRAW_DEBT_MDN            663
LO_INC_DEBT_MDN          2748
MD_INC_DEBT_MDN          1425
HI_INC_DEBT_MDN          4882
DEP_DEBT_MDN             3388
IND_DEBT_MDN             5268
PELL_DEBT_MDN            2723
NOPELL_DEBT_MDN          4412
FEMALE_DEBT_MDN          1867
MALE_DEBT_MDN            3739
FIRSTGEN_DEBT_MDN        1125
NOTFIRSTGEN_DEBT_MDN      750
DEBT_N                    542
GRAD_DEBT_N               783
WDRAW_DEBT_N              590
LO_INC_DEBT_N            2576
MD_INC_DEBT_N             995
HI_INC_DEBT_N            4380
DEP_DEBT_N               2839
IND_DEBT_N               4812
PELL_DEBT_N              2519
NOPELL_DEBT_N            4189
FEMALE_DEBT_N            1660
MALE_DEBT_N              3463
FIRSTGEN_DEBT_N           979
NOTFIRSTGEN_DEBT_N        569
GRAD_DEBT_MDN10YR        1165
CUML_DEBT_N               542
CUML_DEBT_P90             772
CUML_DEBT_P75             632
CUML_DEBT_P25             632
CUML_DEBT_P10             772
DEBT_MDN_SUPP             629
GRAD_DEBT_MDN_SUPP       1381
GRAD_DEBT_MDN10YR_SUPP   1381
dtype: int64
```

In [161]:

```
for i in aid_features:
    aid_df[i] = aid_df.groupby(['UNITID'], sort=False)[i].apply(lambda x: x.
```

```
C:\Users\Jesus Baquiax\anaconda3\envs\learn-env\lib\site-packages\numpy\lib\nan
functions.py:1117: RuntimeWarning: Mean of empty slice
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
```

In [155]:
```
1 aid_df.isna().sum()
```

Out[155]:
```
UNITID                    0
INSTNM                    0
CITY                      0
STABBR                    0
ZIP                       0
main                      0
NUMBRANCH                 0
PREDDEG                   0
HIGHDEG                   0
CONTROL                   0
st_fips                   0
region                    0
Year                      0
COMP_ORIG_YR4_RT          0
PCTPELL                  57
PCTFLOAN                123
DEBT_MDN                498
GRAD_DEBT_MDN           700
WDRAW_DEBT_MDN          596
LO_INC_DEBT_MDN        1342
MD_INC_DEBT_MDN         886
HI_INC_DEBT_MDN        2661
DEP_DEBT_MDN           1934
IND_DEBT_MDN           2923
PELL_DEBT_MDN          1181
NOPELL_DEBT_MDN        2102
FEMALE_DEBT_MDN        1059
MALE_DEBT_MDN          1987
FIRSTGEN_DEBT_MDN       839
NOTFIRSTGEN_DEBT_MDN    646
DEBT_N                  498
GRAD_DEBT_N             425
WDRAW_DEBT_N            539
LO_INC_DEBT_N          1151
MD_INC_DEBT_N           511
HI_INC_DEBT_N          2007
DEP_DEBT_N             1410
IND_DEBT_N             2278
PELL_DEBT_N             986
NOPELL_DEBT_N          1816
FEMALE_DEBT_N           787
MALE_DEBT_N            1602
FIRSTGEN_DEBT_N         622
NOTFIRSTGEN_DEBT_N      416
GRAD_DEBT_MDN10YR       700
CUML_DEBT_N             498
CUML_DEBT_P90           548
CUML_DEBT_P75           455
CUML_DEBT_P25           455
CUML_DEBT_P10           548
DEBT_MDN_SUPP           585
GRAD_DEBT_MDN_SUPP      802
GRAD_DEBT_MDN10YR_SUPP  802
dtype: int64
```

In [162]:
```
imputer4 = KNNImputer(n_neighbors=2, weights="uniform")
AID_KNN = pd.DataFrame(imputer4.fit_transform(aid_df), columns = list(aid_df
```

Aid Features - Correlation

In [163]:
```
AID_KNN['COMP_ORIG_YR4_RT'] = AID_KNN['COMP_ORIG_YR4_RT'].astype(int)
```

In [502]:
```
corr_matrix7 = AID_KNN.corr()
aid_top4 = corr_matrix7['COMP_ORIG_YR4_RT'].nlargest(n=5)
aid_top4
```

Out[502]:
```
COMP_ORIG_YR4_RT    1.000000
CUML_DEBT_P10       0.209088
CUML_DEBT_P25       0.095919
DEBT_MDN            0.026732
IND_DEBT_MDN        0.006619
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [503]:
```
aid_top4_list = list(corr_matrix7['COMP_ORIG_YR4_RT'].nlargest(n=5).index[1:
aid_top4_list
```

Out[503]:
```
['CUML_DEBT_P10', 'CUML_DEBT_P25', 'DEBT_MDN', 'IND_DEBT_MDN']
```

Completion Features

In [273]:
```
completion_features = ['C150_4', 'C150_L4', 'D150_4', 'D150_L4', 'C150_4_WHI
target_completion = target + completion_features
completion_df = processed_df[target_completion].copy()

completion_df.isna().sum()
```

```
MD_INC_COMP_4YR_TRANS_YR2_RT        11939
MD_INC_COMP_2YR_TRANS_YR2_RT        13388
MD_INC_WDRAW_ORIG_YR2_RT             8621
MD_INC_WDRAW_4YR_TRANS_YR2_RT       13908
MD_INC_WDRAW_2YR_TRANS_YR2_RT       13164
MD_INC_ENRL_ORIG_YR2_RT              8089
MD_INC_ENRL_4YR_TRANS_YR2_RT        11623
MD_INC_ENRL_2YR_TRANS_YR2_RT        13631
MD_INC_UNKN_ORIG_YR2_RT             13637
MD_INC_UNKN_4YR_TRANS_YR2_RT        11242
MD_INC_UNKN_2YR_TRANS_YR2_RT        11049
HI_INC_DEATH_YR2_RT                  5814
HI_INC_COMP_ORIG_YR2_RT             12329
HI_INC_COMP_4YR_TRANS_YR2_RT        10242
HI_INC_COMP_2YR_TRANS_YR2_RT        11551
HI_INC_WDRAW_ORIG_YR2_RT            11822
HI_INC_WDRAW_4YR_TRANS_YR2_RT       13411
HI_INC_WDRAW_2YR_TRANS_YR2_RT       12789
HI_INC_ENRL_ORIG_YR2_RT              9638
HI_INC_ENRL_4YR_TRANS_YR2_RT        10979
```

In [257]:
```
imputer8 = KNNImputer(n_neighbors=2, weights="uniform")
completion_KNN = pd.DataFrame(imputer8.fit_transform(completion_df), columns
```

In [259]:
```
completion_KNN.isna().sum()
```

```
NOLOAN_COMP_ORIG_YR2_RT             0
NOLOAN_COMP_4YR_TRANS_YR2_RT        0
NOLOAN_COMP_2YR_TRANS_YR2_RT        0
NOLOAN_WDRAW_ORIG_YR2_RT            0
NOLOAN_WDRAW_4YR_TRANS_YR2_RT       0
NOLOAN_WDRAW_2YR_TRANS_YR2_RT       0
NOLOAN_ENRL_ORIG_YR2_RT             0
NOLOAN_ENRL_4YR_TRANS_YR2_RT        0
NOLOAN_ENRL_2YR_TRANS_YR2_RT        0
NOLOAN_UNKN_ORIG_YR2_RT             0
NOLOAN_UNKN_4YR_TRANS_YR2_RT        0
NOLOAN_UNKN_2YR_TRANS_YR2_RT        0
FIRSTGEN_DEATH_YR2_RT               0
FIRSTGEN_COMP_ORIG_YR2_RT           0
FIRSTGEN_COMP_4YR_TRANS_YR2_RT      0
FIRSTGEN_COMP_2YR_TRANS_YR2_RT      0
FIRSTGEN_WDRAW_ORIG_YR2_RT          0
FIRSTGEN_WDRAW_4YR_TRANS_YR2_RT     0
FIRSTGEN_WDRAW_2YR_TRANS_YR2_RT     0
FIRSTGEN_ENRL_ORIG_YR2_RT           0
```

Completion Features - Correlation

In [260]:
```
completion_KNN['COMP_ORIG_YR4_RT'] = completion_KNN['COMP_ORIG_YR4_RT'].asty
```

In [349]:
```
completion_KNN = completion_KNN.loc[:,~completion_KNN.columns.duplicated()]
```

In [486]:
```
corr_matrix1 = completion_KNN.corr()
completion_top5 = corr_matrix1['COMP_ORIG_YR4_RT'].nlargest(n=6)
completion_top5
```

Out[486]:
```
COMP_ORIG_YR4_RT           1.000000
LOAN_COMP_ORIG_YR2_RT      0.103054
COMP_ORIG_YR2_RT           0.102187
MALE_COMP_ORIG_YR2_RT      0.079000
FEMALE_COMP_ORIG_YR2_RT    0.076622
COMP_ORIG_YR3_RT           0.067808
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [488]:
```
completion_top5_list = list(corr_matrix1['COMP_ORIG_YR4_RT'].nlargest(n=6).i
completion_top5_list
```

Out[488]:
```
['LOAN_COMP_ORIG_YR2_RT',
 'COMP_ORIG_YR2_RT',
 'MALE_COMP_ORIG_YR2_RT',
 'FEMALE_COMP_ORIG_YR2_RT',
 'COMP_ORIG_YR3_RT']
```

Repayment Features

In [175]:
```
repayment_features = ['CDR2', 'CDR3', 'DEATH_YR2_RT', 'COMP_ORIG_YR2_RT', 'C
```

In [176]:
```
target_repayment = target + repayment_features
repayment_df = processed_df[target_repayment].copy()
repayment_df.isna().sum()
```

```
IND_ENRL_2YR_TRANS_YR2_RT          12895
IND_UNKN_ORIG_YR2_RT               11307
IND_UNKN_4YR_TRANS_YR2_RT          13439
IND_UNKN_2YR_TRANS_YR2_RT          13490
FEMALE_DEATH_YR2_RT                 9054
FEMALE_COMP_ORIG_YR2_RT             8612
FEMALE_COMP_4YR_TRANS_YR2_RT       13626
FEMALE_COMP_2YR_TRANS_YR2_RT       13256
FEMALE_WDRAW_ORIG_YR2_RT            7765
FEMALE_WDRAW_4YR_TRANS_YR2_RT      11652
FEMALE_WDRAW_2YR_TRANS_YR2_RT      11579
FEMALE_ENRL_ORIG_YR2_RT             5689
FEMALE_ENRL_4YR_TRANS_YR2_RT        9482
FEMALE_ENRL_2YR_TRANS_YR2_RT       11239
FEMALE_UNKN_ORIG_YR2_RT            10246
FEMALE_UNKN_4YR_TRANS_YR2_RT       13442
FEMALE_UNKN_2YR_TRANS_YR2_RT       13074
MALE_DEATH_YR2_RT                   8946
MALE_COMP_ORIG_YR2_RT              13753
MALE_COMP_4YR_TRANS_YR2_RT         13556
```

In [177]:
```
for i in repayment_features:
    repayment_df[i] = repayment_df.groupby(['UNITID'], sort=False)[i].apply(
```

```
C:\Users\Jesus Baquiax\anaconda3\envs\learn-env\lib\site-packages\numpy\lib\nan
functions.py:1117: RuntimeWarning: Mean of empty slice
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
```

In [178]:
```
repayment_df.isna().sum()
```

```
IND_ENRL_4YR_TRANS_YR2_RT           5841
IND_ENRL_2YR_TRANS_YR2_RT           5900

IND_UNKN_ORIG_YR2_RT                5526
IND_UNKN_4YR_TRANS_YR2_RT           7967
IND_UNKN_2YR_TRANS_YR2_RT           8250
FEMALE_DEATH_YR2_RT                 3459
FEMALE_COMP_ORIG_YR2_RT             3098
FEMALE_COMP_4YR_TRANS_YR2_RT        8524
FEMALE_COMP_2YR_TRANS_YR2_RT        8584
FEMALE_WDRAW_ORIG_YR2_RT            2374
FEMALE_WDRAW_4YR_TRANS_YR2_RT       6512
FEMALE_WDRAW_2YR_TRANS_YR2_RT       5772
FEMALE_ENRL_ORIG_YR2_RT             1962
FEMALE_ENRL_4YR_TRANS_YR2_RT        3921
FEMALE_ENRL_2YR_TRANS_YR2_RT        6183
FEMALE_UNKN_ORIG_YR2_RT             5421
FEMALE_UNKN_4YR_TRANS_YR2_RT        8195
FEMALE_UNKN_2YR_TRANS_YR2_RT        7442
MALE_DEATH_YR2_RT                   2934
MALE_COMP_ORIG_YR2_RT               6419
```

In [182]:
```
imputer5 = KNNImputer(n_neighbors=2, weights="uniform")
Repayment_KNN = pd.DataFrame(imputer5.fit_transform(repayment_df), columns =
```

In [183]:
```
Repayment_KNN.isna().sum()
```

```
FIRSTGEN_ENRL_4YR_TRANS_YR2_RT      0
FIRSTGEN_ENRL_2YR_TRANS_YR2_RT      0
FIRSTGEN_UNKN_ORIG_YR2_RT           0
FIRSTGEN_UNKN_4YR_TRANS_YR2_RT      0
FIRSTGEN_UNKN_2YR_TRANS_YR2_RT      0
NOT1STGEN_DEATH_YR2_RT              0
NOT1STGEN_COMP_ORIG_YR2_RT          0
NOT1STGEN_COMP_4YR_TRANS_YR2_RT     0
NOT1STGEN_COMP_2YR_TRANS_YR2_RT     0
NOT1STGEN_WDRAW_ORIG_YR2_RT         0
NOT1STGEN_WDRAW_4YR_TRANS_YR2_RT    0
NOT1STGEN_WDRAW_2YR_TRANS_YR2_RT    0
NOT1STGEN_ENRL_ORIG_YR2_RT          0
NOT1STGEN_ENRL_4YR_TRANS_YR2_RT     0
NOT1STGEN_ENRL_2YR_TRANS_YR2_RT     0
NOT1STGEN_UNKN_ORIG_YR2_RT          0
NOT1STGEN_UNKN_4YR_TRANS_YR2_RT     0
NOT1STGEN_UNKN_2YR_TRANS_YR2_RT     0
dtype: int64
```

Repayment Features - Correlation

In [184]:
```
Repayment_KNN['COMP_ORIG_YR4_RT'] = Repayment_KNN['COMP_ORIG_YR4_RT'].astype
```

In [492]:
```
corr_matrix9 = Repayment_KNN.corr()
repayment_top5 = corr_matrix9['COMP_ORIG_YR4_RT'].nlargest(n=6)
repayment_top5
```

Out[492]:
```
COMP_ORIG_YR4_RT              1.000000
COMP_ORIG_YR2_RT              0.110795
LOAN_COMP_ORIG_YR2_RT         0.096380
MALE_COMP_ORIG_YR2_RT         0.091781
FEMALE_COMP_ORIG_YR2_RT       0.087515
NOT1STGEN_COMP_ORIG_YR2_RT    0.077709
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [493]:
```
repayment_top5_list = list(corr_matrix9['COMP_ORIG_YR4_RT'].nlargest(n=6).in
repayment_top5_list
```

Out[493]:
```
['COMP_ORIG_YR2_RT',
 'LOAN_COMP_ORIG_YR2_RT',
 'MALE_COMP_ORIG_YR2_RT',
 'FEMALE_COMP_ORIG_YR2_RT',
 'NOT1STGEN_COMP_ORIG_YR2_RT']
```

Earnings Feature

In [189]:
```
earnings_features = ['count_nwne_p10', 'count_wne_p10', 'mn_earn_wne_p10', '
target_earnings = target + earnings_features
earnings_df = processed_df[target_earnings].copy()
earnings_df.isna().sum()
```

Out[189]:
```
UNITID                          0
COMP_ORIG_YR4_RT                0
count_nwne_p10               9450
count_wne_p10                9451
mn_earn_wne_p10              9615
md_earn_wne_p10              9615
pct10_earn_wne_p10           9878
pct25_earn_wne_p10           9878
pct75_earn_wne_p10           9878
pct90_earn_wne_p10           9878
sd_earn_wne_p10              9615
count_wne_inc1_p10           9568
count_wne_inc2_p10           9605
count_wne_inc3_p10          10141
count_wne_indep0_inc1_p10    9955
count_wne_indep0_p10         9634
count_wne_indep1_p10         9762
count_wne_male0_p10          9629
count_wne_male1_p10          9743
gt_25k_p10                   9615
mn_earn_wne_inc1_p10        11646
mn_earn_wne_inc2_p10        11643
mn_earn_wne_inc3_p10        11981
mn_earn_wne_indep0_inc1_p10 13281
mn_earn_wne_indep0_p10      11056
mn_earn_wne_indep1_p10      11019
mn_earn_wne_male0_p10       10495
mn_earn_wne_male1_p10       10495
count_nwne_p6                9354
count_wne_p6                 9355
mn_earn_wne_p6               9480
md_earn_wne_p6               9480
pct10_earn_wne_p6            9712
pct25_earn_wne_p6            9712
pct75_earn_wne_p6            9712
pct90_earn_wne_p6            9712
sd_earn_wne_p6               9480
count_wne_inc1_p6            9440
count_wne_inc2_p6            9473
count_wne_inc3_p6            9917
count_wne_indep0_inc1_p6     9748
count_wne_indep0_p6          9490
count_wne_indep1_p6          9692
count_wne_male0_p6           9510
count_wne_male1_p6           9570
gt_25k_p6                    9480
mn_earn_wne_inc1_p6         11512
mn_earn_wne_inc2_p6         11478
mn_earn_wne_inc3_p6         11610
mn_earn_wne_indep0_inc1_p6  12930
mn_earn_wne_indep0_p6       10889
```

```
mn_earn_wne_indep1_p6          10911
mn_earn_wne_male0_p6           10301
mn_earn_wne_male1_p6           10302
count_nwne_p7                  12425
count_wne_p7                   12428
mn_earn_wne_p7                 12505
sd_earn_wne_p7                 12505
gt_25k_p7                      12505
count_nwne_p8                   9403
count_wne_p8                    9403
mn_earn_wne_p8                  9538
md_earn_wne_p8                  9538
pct10_earn_wne_p8               9776
pct25_earn_wne_p8               9776
pct75_earn_wne_p8               9776
pct90_earn_wne_p8               9776
sd_earn_wne_p8                  9538
gt_25k_p8                       9538
count_nwne_p9                  12457
count_wne_p9                   12457
mn_earn_wne_p9                 12558
sd_earn_wne_p9                 12558
gt_25k_p9                      12558
dtype: int64
```

In [190]:

```
for i in earnings_features:
    earnings_df[i] = earnings_df.groupby(['UNITID'], sort=False)[i].apply(la
```

```
C:\Users\Jesus Baquiax\anaconda3\envs\learn-env\lib\site-packages\numpy\lib\nan
functions.py:1117: RuntimeWarning: Mean of empty slice
  return np.nanmean(a, axis, out=out, keepdims=keepdims)
```

In [191]:
```
1 earnings_df.isna().sum()
```

Out[191]:

UNITID	0
COMP_ORIG_YR4_RT	0
count_nwne_p10	520
count_wne_p10	526
mn_earn_wne_p10	775
md_earn_wne_p10	775
pct10_earn_wne_p10	1201
pct25_earn_wne_p10	1201
pct75_earn_wne_p10	1201
pct90_earn_wne_p10	1201
sd_earn_wne_p10	775
count_wne_inc1_p10	707
count_wne_inc2_p10	747
count_wne_inc3_p10	1545
count_wne_indep0_inc1_p10	1247
count_wne_indep0_p10	806
count_wne_indep1_p10	924
count_wne_male0_p10	847
count_wne_male1_p10	947
gt_25k_p10	775
mn_earn_wne_inc1_p10	3996
mn_earn_wne_inc2_p10	4025
mn_earn_wne_inc3_p10	4762
mn_earn_wne_indep0_inc1_p10	6767
mn_earn_wne_indep0_p10	3186
mn_earn_wne_indep1_p10	3158
mn_earn_wne_male0_p10	2286
mn_earn_wne_male1_p10	2286
count_nwne_p6	366
count_wne_p6	366
mn_earn_wne_p6	577
md_earn_wne_p6	577
pct10_earn_wne_p6	912
pct25_earn_wne_p6	912
pct75_earn_wne_p6	912
pct90_earn_wne_p6	912
sd_earn_wne_p6	577
count_wne_inc1_p6	491
count_wne_inc2_p6	556
count_wne_inc3_p6	1231
count_wne_indep0_inc1_p6	911
count_wne_indep0_p6	566
count_wne_indep1_p6	849
count_wne_male0_p6	623
count_wne_male1_p6	671
gt_25k_p6	577
mn_earn_wne_inc1_p6	3915
mn_earn_wne_inc2_p6	3861
mn_earn_wne_inc3_p6	4285
mn_earn_wne_indep0_inc1_p6	6301
mn_earn_wne_indep0_p6	3049
mn_earn_wne_indep1_p6	3077
mn_earn_wne_male0_p6	1971
mn_earn_wne_male1_p6	1977
count_nwne_p7	908

```
count_wne_p7                912
mn_earn_wne_p7             1107
sd_earn_wne_p7             1107
gt_25k_p7                  1107
count_nwne_p8               438
count_wne_p8                438
mn_earn_wne_p8              667
md_earn_wne_p8              667
pct10_earn_wne_p8          1051
pct25_earn_wne_p8          1051
pct75_earn_wne_p8          1051
pct90_earn_wne_p8          1051
sd_earn_wne_p8              667
gt_25k_p8                   667
count_nwne_p9               976
count_wne_p9                976
mn_earn_wne_p9             1250
sd_earn_wne_p9             1250
gt_25k_p9                  1250
dtype: int64
```

In [192]:
```
imputer6 = KNNImputer(n_neighbors=2, weights="uniform")
earnings_KNN = pd.DataFrame(imputer6.fit_transform(earnings_df), columns = l
```

Earnings Features - Correlation

In [193]:
```
earnings_KNN['COMP_ORIG_YR4_RT'] = earnings_KNN['COMP_ORIG_YR4_RT'].astype(i
```

In [330]:
```
corr_matrix = earnings_KNN.corr()
earnings_top5 = corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6)
earnings_top5
```

Out[330]:
```
COMP_ORIG_YR4_RT         1.000000
pct10_earn_wne_p10       0.099295
mn_earn_wne_indep0_p6    0.092026
pct10_earn_wne_p8        0.088269
pct25_earn_wne_p6        0.087437
md_earn_wne_p8           0.086629
Name: COMP_ORIG_YR4_RT, dtype: float64
```

In [332]:
```
earnings_top5_list = list(corr_matrix['COMP_ORIG_YR4_RT'].nlargest(n=6).inde
earnings_top5_list
```

Out[332]:
```
['pct10_earn_wne_p10',
 'mn_earn_wne_indep0_p6',
 'pct10_earn_wne_p8',
 'pct25_earn_wne_p6',
 'md_earn_wne_p8']
```

Feature / Correlation Summary

Below are the top five features for each category within the dataset. These features will be used as the starting point for modelling in the notebook. A dataframe will be created and saved using these features.

In [504]:

```
print(school_top5[1:],'\n',
      admission_top5[1:],'\n',
      percentage_top5[1:], '\n',
      offered_top5[1:], '\n',
      student_top5[1:], '\n',
      cost_top5[1:], '\n',
      aid_top4[1:],'\n',
      completion_top5[1:], '\n',
      repayment_top5[1:], '\n',
      earnings_top5[1:])
```

```
ZIP_60616-3878    0.408080
ZIP_97230-3099    0.333187
ZIP_10027-4649    0.235592
CITY_Needham      0.235592
ZIP_02492-1200    0.235592
Name: COMP_ORIG_YR4_RT, dtype: float64
 SATMTMID       0.025243
SATMT25        0.024867
SAT_AVG_ALL    0.024282
SATMT75        0.023885
ACTEN25        0.023854
Name: COMP_ORIG_YR4_RT, dtype: float64
 PCIP45    0.385909
PCIP23    0.384487
PCIP54    0.381086
PCIP16    0.319399
PCIP26    0.316918
Name: COMP_ORIG_YR4_RT, dtype: float64
 CIP51BACHL_1    0.023110
CIP26BACHL_1    0.008103
CIP39CERT4_2   -0.000233
CIP25CERT4_2   -0.000233
CIP29CERT4_2   -0.000233
Name: COMP_ORIG_YR4_RT, dtype: float64
 UGDS_NRA      0.063676
UGDS_ASIAN    0.027090
UGDS_UNKN     0.016609
UGDS_API      0.010668
UGDS          0.004897
Name: COMP_ORIG_YR4_RT, dtype: float64
 TUITIONFEE_OUT    0.670658
COSTT4_A          0.626523
TUITIONFEE_IN     0.621967
NPT45_PUB         0.579966
NPT4_75UP_PUB     0.573320
Name: COMP_ORIG_YR4_RT, dtype: float64
 CUML_DEBT_P10    0.209088
CUML_DEBT_P25    0.095919
DEBT_MDN         0.026732
IND_DEBT_MDN     0.006619
Name: COMP_ORIG_YR4_RT, dtype: float64
 LOAN_COMP_ORIG_YR2_RT      0.103054
COMP_ORIG_YR2_RT           0.102187
MALE_COMP_ORIG_YR2_RT      0.079000
```

```
FEMALE_COMP_ORIG_YR2_RT    0.076622
COMP_ORIG_YR3_RT           0.067808
Name: COMP_ORIG_YR4_RT, dtype: float64
 COMP_ORIG_YR2_RT             0.110795
LOAN_COMP_ORIG_YR2_RT         0.096380
MALE_COMP_ORIG_YR2_RT         0.091781
FEMALE_COMP_ORIG_YR2_RT       0.087515
NOT1STGEN_COMP_ORIG_YR2_RT    0.077709
Name: COMP_ORIG_YR4_RT, dtype: float64
 pct10_earn_wne_p10       0.099295
mn_earn_wne_indep0_p6    0.092026
pct10_earn_wne_p8        0.088269
pct25_earn_wne_p6        0.087437
md_earn_wne_p8           0.086629
Name: COMP_ORIG_YR4_RT, dtype: float64
```

Final Cleaning

In [395]:
```
offer_top5_series=pd.Series(offered_top5_list).replace({'CIP51BACHL_1':'CIP5
                           'CIP26BACHL_1':'CIP26BACHL',
                           'CIP39CERT4_2': 'CIP39CERT4',
                           'CIP25CERT4_2':'CIP25CERT4',
                           'CIP29CERT4_2':'CIP29CERT4'})

offered_top5_list = list(offer_top5_series)
```

In [438]:
```
zip_city = ['ZIP', 'CITY']
```

In [404]:
```
offered_top5_list
```

Out[404]:
```
['CIP51BACHL', 'CIP26BACHL', 'CIP39CERT4', 'CIP25CERT4', 'CIP29CERT4']
```

In [505]:
```
ered_top5_list+student_top5_list+cost_top5_list+aid_top4_list+completion_top5_lis
```

In [496]:
```
aid_top5_list
```

Out[496]:
```
['CUML_DEBT_P10', 'CUML_DEBT_P25', 'DEBT_MDN', 'IND_DEBT_MDN', 'UNITID']
```

In [506]:
```
1 top_features_list
```

Out[506]:
```
['UNITID',
 'COMP_ORIG_YR4_RT',
 'ZIP',
 'CITY',
 'SATMTMID',
 'SATMT25',
 'SAT_AVG_ALL',
 'SATMT75',
 'ACTEN25',
 'PCIP45',
 'PCIP23',
 'PCIP54',
 'PCIP16',
 'PCIP26',
 'CIP51BACHL',
 'CIP26BACHL',
 'CIP39CERT4',
 'CIP25CERT4',
 'CIP29CERT4',
 'UGDS_NRA',
 'UGDS_ASIAN',
 'UGDS_UNKN',
 'UGDS_API',
 'UGDS',
 'TUITIONFEE_OUT',
 'COSTT4_A',
 'TUITIONFEE_IN',
 'NPT45_PUB',
 'NPT4_75UP_PUB',
 'CUML_DEBT_P10',
 'CUML_DEBT_P25',
 'DEBT_MDN',
 'IND_DEBT_MDN',
 'LOAN_COMP_ORIG_YR2_RT',
 'COMP_ORIG_YR2_RT',
 'MALE_COMP_ORIG_YR2_RT',
 'FEMALE_COMP_ORIG_YR2_RT',
 'COMP_ORIG_YR3_RT',
 'COMP_ORIG_YR2_RT',
 'LOAN_COMP_ORIG_YR2_RT',
 'MALE_COMP_ORIG_YR2_RT',
 'FEMALE_COMP_ORIG_YR2_RT',
 'NOT1STGEN_COMP_ORIG_YR2_RT',
 'pct10_earn_wne_p10',
 'mn_earn_wne_indep0_p6',
 'pct10_earn_wne_p8',
 'pct25_earn_wne_p6',
 'md_earn_wne_p8']
```

In [507]:
```
top_features_df = processed_df[top_features_list].copy()
top_features_df.isna().sum()
```

Out[507]:
```
UNITID                          0
COMP_ORIG_YR4_RT                0
ZIP                             0
CITY                            0
SATMTMID                    10755
SATMT25                     10755
SAT_AVG_ALL                  9645
SATMT75                     10755
ACTEN25                     11738
PCIP45                          0
PCIP23                          0
PCIP54                          0
PCIP16                          0
PCIP26                          0
CIP51BACHL                      0
CIP26BACHL                      0
CIP39CERT4                      0
CIP25CERT4                      0
CIP29CERT4                      0
UGDS_NRA                       10
UGDS_ASIAN                   2859
UGDS_UNKN                      10
UGDS_API                     9574
UGDS                           10
TUITIONFEE_OUT                776
COSTT4_A                     6259
TUITIONFEE_IN                 776
NPT45_PUB                   14300
NPT4_75UP_PUB               13911
CUML_DEBT_P10                 772
CUML_DEBT_P25                 632
DEBT_MDN                      605
IND_DEBT_MDN                 5268
LOAN_COMP_ORIG_YR2_RT       10113
COMP_ORIG_YR2_RT             1276
MALE_COMP_ORIG_YR2_RT       13753
FEMALE_COMP_ORIG_YR2_RT      8612
COMP_ORIG_YR3_RT              411
COMP_ORIG_YR2_RT             1276
LOAN_COMP_ORIG_YR2_RT       10113
MALE_COMP_ORIG_YR2_RT       13753
FEMALE_COMP_ORIG_YR2_RT      8612
NOT1STGEN_COMP_ORIG_YR2_RT   4644
pct10_earn_wne_p10           9878
mn_earn_wne_indep0_p6       10889
pct10_earn_wne_p8            9776
pct25_earn_wne_p6            9712
md_earn_wne_p8               9538
dtype: int64
```

In [514]:
```
cat_columns = ['ZIP', 'CITY', 'CIP51BACHL', 'CIP26BACHL', 'CIP39CERT4', 'CIP

cont_columns_df = top_features_df.drop(cat_columns, axis=1).copy()

cat_columns_df = top_features_df[cat_columns].copy()
```

In [527]:
```
cat_columns_df_dummied = pd.get_dummies(cat_columns_df, drop_first=True)

```

In [516]:
```
cont_columns_list = list(cont_columns_df.columns)
cont_columns_list
```

Out[516]:
```
['UNITID',
 'COMP_ORIG_YR4_RT',
 'SATMTMID',
 'SATMT25',
 'SAT_AVG_ALL',
 'SATMT75',
 'ACTEN25',
 'PCIP45',
 'PCIP23',
 'PCIP54',
 'PCIP16',
 'PCIP26',
 'UGDS_NRA',
 'UGDS_ASIAN',
 'UGDS_UNKN',
 'UGDS_API',
 'UGDS',
 'TUITIONFEE_OUT',
 'COSTT4_A',
 'TUITIONFEE_IN',
 'NPT45_PUB',
 'NPT4_75UP_PUB',
 'CUML_DEBT_P10',
 'CUML_DEBT_P25',
 'DEBT_MDN',
 'IND_DEBT_MDN',
 'LOAN_COMP_ORIG_YR2_RT',
 'COMP_ORIG_YR2_RT',
 'MALE_COMP_ORIG_YR2_RT',
 'FEMALE_COMP_ORIG_YR2_RT',
 'COMP_ORIG_YR3_RT',
 'COMP_ORIG_YR2_RT',
 'LOAN_COMP_ORIG_YR2_RT',
 'MALE_COMP_ORIG_YR2_RT',
 'FEMALE_COMP_ORIG_YR2_RT',
 'NOT1STGEN_COMP_ORIG_YR2_RT',
 'pct10_earn_wne_p10',
 'mn_earn_wne_indep0_p6',
 'pct10_earn_wne_p8',
 'pct25_earn_wne_p6',
 'md_earn_wne_p8']
```

In [519]:
```
imputer = KNNImputer(n_neighbors=2, weights="uniform")
top_features_KNN = pd.DataFrame(imputer.fit_transform(cont_columns_df), colu
```

In [521]:
```
top_features_KNN.isna().sum()
```

Out[521]:
```
UNITID                0
COMP_ORIG_YR4_RT      0
SATMTMID              0
SATMT25               0
SAT_AVG_ALL           0
SATMT75               0
ACTEN25               0
PCIP45                0
PCIP23                0
PCIP54                0
PCIP16                0
PCIP26                0
UGDS_NRA              0
UGDS_ASIAN            0
UGDS_UNKN             0
UGDS_API              0
UGDS                  0
TUITIONFEE_OUT        0
COSTT4_A              0
TUITIONFEE_IN         0
```

In [528]:
```
top_features_KNN.shape, cat_columns_df_dummied.shape
```

Out[528]:
```
((18233, 41), (18233, 5509))
```

In [535]:
```
final_df = pd.concat([top_features_KNN.reset_index(drop=True), cat_columns_d
```

In [536]:
```
final_df.shape
```

Out[536]:
```
(18233, 5550)
```

In [537]:
```
X = final_df.drop('COMP_ORIG_YR4_RT',axis=1).copy()
y = final_df['COMP_ORIG_YR4_RT']
X.shape, y.shape
```

Out[537]:
```
((18233, 5549), (18233,))
```

In [538]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Baseline Model

In [539]:
```
dr = DummyRegressor(strategy='median')
dr.fit(X_train, y_train)
dr.predict(X_train)
dr.score(X_train, y_train)
```

Out[539]: -0.003910667228882003

First Model

In [540]:
```
linreg = LinearRegression()
linreg.fit(X_train, y_train)
```

Out[540]: LinearRegression()

In [541]:
```
linreg.score(X_train, y_train)
```

Out[541]: 0.9490044111047855

In [543]:
```
linreg.score(X_test, y_test)
```

Out[543]: 0.9066136894339951

In [ ]:
```

```