In [737]:
```
import pandas as pd

data = pd.read_csv("C:\\Users\\virin\\Desktop\\edx\\data\\final\\public\\train_va
data1= pd.read_csv("C:\Users\\virin\\Desktop\\edx\\data\\final\\public\\train_lab
```

In [738]:
```
data.head(5)
```

Out[738]:

| | row_id | academics__program_assoc_agriculture | academics__program_assoc_architecture | academi |
|---|---|---|---|---|
| **0** | 0 | 0.0 | 0.0 | |
| **1** | 1 | 0.0 | 0.0 | |
| **2** | 3 | 0.0 | 0.0 | |
| **3** | 4 | 0.0 | 0.0 | |
| **4** | 5 | 0.0 | 0.0 | |

5 rows × 298 columns

In [739]:
```
data1.head(5)
```

Out[739]:

| | row_id | income |
|---|---|---|
| **0** | 0 | 46.9 |
| **1** | 1 | 26.7 |
| **2** | 3 | 28.1 |
| **3** | 4 | 41.6 |
| **4** | 5 | 34.3 |

In [740]:
```
final_data = pd.merge(data, data1, on='row_id')
```

In [741]:
```
final_data.head(5)
```

Out[741]:

| | row_id | academics__program_assoc_agriculture | academics__program_assoc_architecture | academi |
|---|---|---|---|---|
| **0** | 0 | 0.0 | 0.0 | |
| **1** | 1 | 0.0 | 0.0 | |
| **2** | 3 | 0.0 | 0.0 | |
| **3** | 4 | 0.0 | 0.0 | |
| **4** | 5 | 0.0 | 0.0 | |

5 rows × 299 columns

In [742]:
```
final_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17107 entries, 0 to 17106
Columns: 299 entries, row_id to income
dtypes: float64(288), int64(2), object(9)
memory usage: 39.2+ MB
```

In [743]:
```
final_data.describe()
```

Out[743]:

| | row_id | academics__program_assoc_agriculture | academics__program_assoc_architecture |
|---|---|---|---|
| **count** | 17107.000000 | 16393.000000 | 16393.000000 |
| **mean** | 13259.717835 | 0.074483 | 0.019582 |
| **std** | 7577.385374 | 0.266713 | 0.139876 |
| **min** | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 6715.500000 | 0.000000 | 0.000000 |
| **50%** | 13290.000000 | 0.000000 | 0.000000 |
| **75%** | 19835.500000 | 0.000000 | 0.000000 |
| **max** | 26296.000000 | 2.000000 | 2.000000 |

8 rows × 290 columns

In [744]:
```
corr = final_data.corr()
```

In [745]:
```
corr["income"].sort_values(ascending = False)
```

Out[745]:
```
income                                                    1.000000
admissions__act_scores_25th_percentile_math               0.587557
admissions__act_scores_midpoint_math                      0.586934
admissions__sat_scores_25th_percentile_math               0.570555
admissions__sat_scores_midpoint_math                      0.565676
admissions__act_scores_75th_percentile_math               0.555352
school__degrees_awarded_predominant_recoded               0.542336
admissions__sat_scores_75th_percentile_math               0.538334
admissions__act_scores_25th_percentile_cumulative         0.537562
admissions__sat_scores_average_by_ope_id                  0.535409
admissions__sat_scores_average_overall                    0.529695
school__faculty_salary                                    0.527972
admissions__sat_scores_25th_percentile_writing            0.527952
admissions__act_scores_midpoint_cumulative                0.525491
admissions__sat_scores_midpoint_writing                   0.517546
admissions__act_scores_25th_percentile_english            0.502382
student__share_firstgeneration_parents_somecollege        0.494568
admissions__act_scores_75th_percentile_cumulative         0.493378
admissions__sat_scores_75th_percentile_writing            0.489239
admissions__sat_scores_25th_percentile_critical_reading   0.488383
admissions__act_scores_midpoint_english                   0.477511
cost__tuition_out_of_state                                0.471151
admissions__sat_scores_midpoint_critical_reading          0.468217
admissions__act_scores_75th_percentile_english            0.434218
admissions__sat_scores_75th_percentile_critical_reading   0.429264
cost__tuition_in_state                                    0.390713
admissions__act_scores_25th_percentile_writing            0.375003
academics__program_bachelors_health                       0.359164
admissions__act_scores_midpoint_writing                   0.357413
academics__program_bachelors_computer                     0.354342
                                                             ...
academics__program_assoc_personal_culinary               -0.079466
academics__program_assoc_mechanic_repair_technology      -0.082736
academics__program_certificate_lt_1_yr_construction      -0.085878
academics__program_certificate_lt_1_yr_business_marketing -0.088682
student__demographics_age_entry                          -0.089189
academics__program_certificate_lt_1_yr_precision_production -0.091576
academics__program_certificate_lt_2_yr_security_law_enforcement -0.094032
academics__program_certificate_lt_2_yr_computer          -0.094034
academics__program_certificate_lt_2_yr_family_consumer_science -0.095671
academics__program_certificate_lt_2_yr_precision_production -0.102292
academics__program_certificate_lt_1_yr_mechanic_repair_technology -0.103767
academics__program_certificate_lt_2_yr_engineering_technology -0.105071
academics__program_certificate_lt_2_yr_construction      -0.114249
academics__program_certificate_lt_2_yr_mechanic_repair_technology -0.119474
student__share_25_older                                  -0.122160
academics__program_certificate_lt_4_yr_personal_culinary -0.150937
student__retention_rate_lt_four_year_part_time           -0.157655
academics__program_certificate_lt_2_yr_business_marketing -0.157801
academics__program_certificate_lt_1_yr_health            -0.163507
academics__program_certificate_lt_2_yr_health            -0.200592
student__share_independent_students                      -0.220619
student__demographics_female_share                       -0.250458
admissions__admission_rate_overall                       -0.288275
admissions__admission_rate_by_ope_id                     -0.294944
```

```
academics__program_certificate_lt_1_yr_personal_culinary          -0.311912
academics__program_certificate_lt_2_yr_personal_culinary          -0.376604
academics__program_percentage_personal_culinary                   -0.420519
student__share_firstgeneration                                    -0.504396
student__demographics_first_generation                            -0.504396
student__share_firstgeneration_parents_highschool                 -0.522094
Name: income, Length: 290, dtype: float64
```

In [746]:
```
print(final_data["admissions__sat_scores_average_overall"].isnull().sum())
```

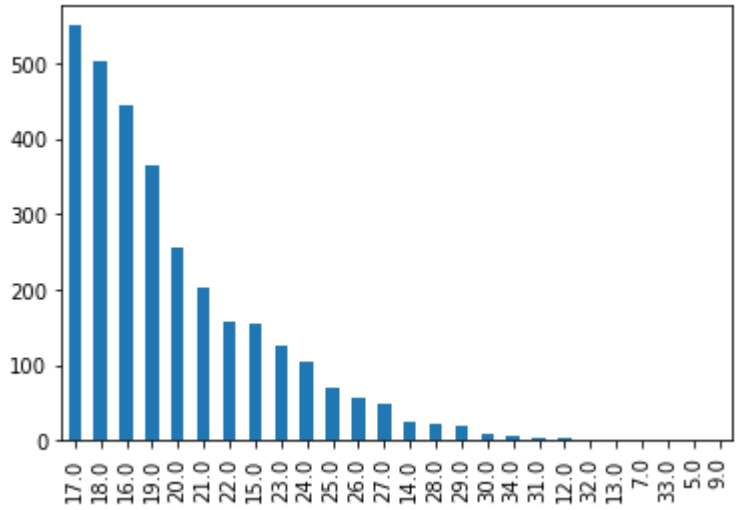
```
13082
```

In [747]:
```
count_1 = pd.value_counts(final_data["admissions__act_scores_25th_percentile_math
```

In [748]:
```
print(count_1)
```

```
17.0    549
18.0    503
16.0    444
19.0    364
20.0    255
21.0    203
22.0    157
15.0    155
23.0    126
24.0    104
25.0     69
26.0     57
27.0     48
14.0     25
28.0     23
29.0     19
30.0     10
34.0      7
31.0      5
12.0      3
32.0      2
13.0      2
7.0       1
33.0      1
5.0       1
9.0       1
Name: admissions__act_scores_25th_percentile_math, dtype: int64
```

In [749]:
```
import matplotlib.pyplot as plt
count_1.plot(kind = "bar")
plt.show()
```




In [750]:
```
from sklearn.preprocessing import Imputer
import numpy as np

imputer = Imputer(strategy = "median")
x =final_data.iloc[:, np.r_[229:298]]


final_data= final_data.fillna((x.median()), inplace=True)
final_data["admissions__act_scores_25th_percentile_math"].head(5)
```

Out[750]:
```
0    18.0
1    18.0
2    18.0
3    19.0
4    18.0
Name: admissions__act_scores_25th_percentile_math, dtype: float64
```

In [751]:
```
corr.loc["income", "cost__tuition_in_state"]
```

Out[751]:
```
0.39071261247853128
```

In [752]:
```
import statsmodels.api as sm

X = final_data["school__degrees_awarded_predominant_recoded"]
y = final_data["income"]

# Note the difference in argument order
model = sm.OLS(y, X).fit()
predictions = model.predict(X) # make the predictions by the model

# Print out the statistics
model.summary()
```

Out[752]:

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | income | **R-squared:** | 0.875 |
| **Model:** | OLS | **Adj. R-squared:** | 0.875 |
| **Method:** | Least Squares | **F-statistic:** | 1.197e+05 |
| **Date:** | Tue, 03 Oct 2017 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 02:53:06 | **Log-Likelihood:** | -66105. |
| **No. Observations:** | 17107 | **AIC:** | 1.322e+05 |
| **Df Residuals:** | 17106 | **BIC:** | 1.322e+05 |
| **Df Model:** | 1 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **school__degrees_awarded_predominant_recoded** | 13.9962 | 0.040 | 345.922 | 0.000 | 13.917 | 14.075 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 3708.288 | **Durbin-Watson:** | 1.896 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 15628.445 |
| **Skew:** | 1.014 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 7.221 | **Cond. No.** | 1.00 |

In [753]:
```
some_data = final_data["income"]
input_data = final_data["school__degrees_awarded_predominant_recoded"]
```

In [754]:
```
pred = model.predict(input_data)
print(model.predict(input_data))
print(some_data)
from sklearn.metrics import mean_squared_error
lin_mse = mean_squared_error(pred, some_data)
rmse = np.sqrt(lin_mse)
rmse
```

```
0        13.996193
1        41.988578
2        13.996193
3        41.988578
4        41.988578
5        27.992385
6        41.988578
7        41.988578
8        13.996193
9        13.996193
10       13.996193
11       41.988578
12       13.996193
13       13.996193
14       27.992385
15       13.996193
16       13.996193
17       27.992385
18       41.988578
19       41.988578
```

In [793]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn import preprocessing

some_data = (final_data["income"])
print(type(some_data))
input_data = final_data[["school__degrees_awarded_predominant_recoded","student__
                         "student__share_firstgeneration_parents_highschool","sch

x = input_data.values #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
input_data = pd.DataFrame(x_scaled)

tree_reg = RandomForestRegressor(max_depth = 30, random_state=2)
tree_reg.fit(input_data,some_data)
tree_pred = tree_reg.predict(input_data)
print(pd.DataFrame(tree_pred))
print(some_data)
dec_mse = mean_squared_error(tree_pred, some_data)
rmse = np.sqrt(dec_mse)
rmse
```

```
17089    22.6
17090    14.5
17091    22.2
17092    25.8
17093    33.2
17094    29.0
17095    32.8
17096    25.4
17097    24.5
17098    20.4
17099    22.5
17100    22.0
17101    33.0
17102    19.5
17103    20.7
17104    28.0
17105    22.4
17106    24.8
Name: income, Length: 17107, dtype: float64
```

Out[793]: 3.9153201542813144

In [794]:
```
test_data = pd.read_csv("C:\\Users\\virin\\Desktop\\edx\\data\\final\\public\\tes

x =test_data.iloc[:, np.r_[229:298]]
test_data= test_data.fillna((x.median()), inplace=True)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
```

In [795]:
```
from sklearn.model_selection import cross_val_score
scores = cross_val_score(tree_reg,input_data,some_data,scoring = "neg_mean_square
rmse_scores = np.sqrt(-scores)
```

In [796]:
```
def display_scores(scores):
    print("Scores:", scores)
    print("Mean:", scores.mean())
    print("SD:",scores.std())
```

In [797]:
```
display_scores(rmse_scores)
```

```
('Scores:', array([ 7.94263458,  7.81807622,  8.10031048,  7.56083547,  8.06440
528,
        8.198119  ,  8.05086697,  7.67651998,  8.08752756,  7.71258032]))
('Mean:', 7.9211875873039599)
('SD:', 0.20447441427727969)
```

In [756]:
```
degree_count = pd.value_counts(final_data["school__degrees_awarded_highest"])
print(degree_count)
```

```
Certificate degree     4757
Associate degree       4693
Graduate degree        4558
Bachelor's degree      2153
Non-degree-granting     946
Name: school__degrees_awarded_highest, dtype: int64
```

In [757]:
```
def degree_to_numeric(x):
    if x=="Graduate degree":
        return 4
    if x=="Bachelor's degree":
        return 3
    if x=="Associate degree":
        return 2
    if x=="Certificate degree":
        return 1
    if x=="Non-degree-granting":
        return 0
```

In [758]:
```
final_data['highest_degree_num'] = final_data['school__degrees_awarded_highest'].
final_data['highest_degree_num']
```

Out[758]:
```
0        1
1        3
2        1
3        4
4        4
5        2
6        4
7        4
8        1
9        1
10       1
11       3
12       1
13       1
14       2
15       1
16       3
17       2
18       2
19       3
20       2
21       4
22       1
23       4
24       1
25       2
26       4
27       2
28       2
29       1
        ..
17077    1
17078    1
17079    4
17080    4
17081    4
17082    2
17083    1
17084    1
17085    2
17086    1
17087    1
17088    2
17089    1
17090    1
17091    2
17092    2
17093    4
17094    3
17095    2
17096    1
17097    2
17098    1
17099    1
```

```
17100    1
17101    3
17102    2
17103    2
17104    2
17105    1
17106    1
Name: highest_degree_num, Length: 17107, dtype: int64
```

In [759]:
```
degree_count1 = pd.value_counts(final_data["school__degrees_awarded_predominant"])
print(degree_count1)
```

```
Predominantly certificate-degree granting    6541
Predominantly bachelor's-degree granting     5452
Predominantly associate's-degree granting    4167
Not classified                                761
Entirely graduate-degree granting             186
Name: school__degrees_awarded_predominant, dtype: int64
```

In [760]:
```
def degree_to_numeric(x):
    if x=="Entirely graduate-degree granting":
        return 4
    if x=="Predominantly bachelor's-degree granting":
        return 3
    if x=="Predominantly associate's-degree granting":
        return 2
    if x=="Predominantly certificate-degree granting":
        return 1
    if x=="Not classified":
        return 0
```

In [761]:
```
final_data['highest_degree_num1'] = final_data['school__degrees_awarded_predomina
final_data['highest_degree_num1']
```

Out[761]:
```
0        1
1        3
2        1
3        3
4        3
5        1
6        4
7        3
8        1
9        1
10       1
11       3
12       1
13       1
14       2
15       1
16       1
17       2
18       2
19       3
20       1
21       3
22       1
23       3
24       1
25       1
26       3
27       1
28       1
29       1
        ..
17077    1
17078    1
17079    3
17080    3
17081    3
17082    1
17083    1
17084    1
17085    2
17086    1
17087    1
17088    1
17089    1
17090    1
17091    2
17092    2
17093    3
17094    3
17095    2
17096    1
17097    2
17098    1
17099    1
```

```
17100    1
17101    2
17102    2
17103    1
17104    1
17105    1
17106    1
Name: highest_degree_num1, Length: 17107, dtype: int64
```

In [762]:
```
degree_count1 = pd.value_counts(final_data["school__main_campus"])
print(degree_count1)
```

```
Main campus        12489
Not main campus     4618
Name: school__main_campus, dtype: int64
```

In [763]:
```
def campus_to_numeric(x):
    if x=="Main campus":
        return 1
    if x=="Not main campus":
        return 0
```

In [764]:
```
final_data['school__main_campus'] = final_data['school__main_campus'].apply(campu
final_data['school__main_campus']
```

Out[764]:
```
0        0
1        1
2        1
3        1
4        1
5        1
6        1
7        1
8        0
9        1
10       0
11       1
12       1
13       1
14       1
15       0
16       1
17       1
18       1
19       0
20       1
21       1
22       1
23       1
24       0
25       0
26       1
27       1
28       0
29       0
        ..
17077    1
17078    1
17079    1
17080    1
17081    0
17082    1
17083    1
17084    0
17085    1
17086    1
17087    1
17088    1
17089    1
17090    1
17091    1
17092    1
17093    1
17094    0
17095    1
17096    0
17097    1
```

```
17098    0
17099    0
17100    0
17101    0
17102    1
17103    1
17104    0
17105    0
17106    1
Name: school__main_campus, Length: 17107, dtype: int64
```

In [765]:
```
def degree_to_numeric_t(x):
    if x=="Entirely graduate-degree granting":
        return 4
    if x=="Predominantly bachelor's-degree granting":
        return 3
    if x=="Predominantly associate's-degree granting":
        return 2
    if x=="Predominantly certificate-degree granting":
        return 1
    if x=="Not classified":
        return 0
```

In [766]:
```
test_data['highest_degree_num'] = test_data['school__degrees_awarded_predominant'
test_data['highest_degree_num']
```

Out[766]:
```
0       4
1       3
2       3
3       1
4       0
5       0
6       2
7       2
8       3
9       3
10      0
11      2
12      3
13      2
14      1
15      1
16      2
17      1
18      2
19      1
20      3
21      2
22      0
23      1
24      3
25      1
26      3
27      2
28      2
29      3
       ..
9162    0
9163    3
9164    3
9165    3
9166    0
9167    0
9168    0
9169    2
9170    3
9171    2
9172    2
9173    2
9174    3
9175    3
9176    1
9177    3
9178    3
9179    1
9180    3
9181    2
9182    1
9183    2
9184    1
```

```
9185    2
9186    0
9187    3
9188    1
9189    3
9190    3
9191    3
Name: highest_degree_num, Length: 9192, dtype: int64
```

In [767]:
```
def campus_to_numeric(x):
    if x=="Main campus":
        return 1
    if x=="Not main campus":
        return 0
```

In [768]:
```
test_data['school__main_campus'] = test_data['school__main_campus'].apply(campus_
test_data['school__main_campus']
```

Out[768]:
```
0       1
1       1
2       1
3       1
4       0
5       0
6       0
7       0
8       0
9       1
10      0
11      0
12      1
13      0
14      0
15      0
16      1
17      1
18      1
19      0
20      0
21      1
22      1
23      1
24      0
25      0
26      1
27      0
28      1
29      1
       ..
9162    0
9163    1
9164    0
9165    1
9166    0
9167    1
9168    1
9169    0
9170    1
9171    0
9172    0
9173    1
9174    1
9175    1
9176    1
9177    1
9178    1
9179    1
9180    1
9181    1
9182    1
9183    0
9184    0
```

```
9185    1
9186    0
9187    1
9188    1
9189    1
9190    1
9191    1
Name: school__main_campus, Length: 9192, dtype: int64
```

In [798]:
```
test_input_data = test_data[["school__degrees_awarded_predominant_recoded","stude
                             "student__share_firstgeneration_parents_highschool","sch
```

In [799]:
```
tree_pred = pd.DataFrame(tree_reg.predict(test_input_data), columns = ["income"])
tree_pred.head(5)
```

Out[799]:

| | income |
|---|---|
| **0** | 51.32 |
| **1** | 65.47 |
| **2** | 67.60 |
| **3** | 51.32 |
| **4** | 60.20 |

In [786]:
```
test_data_id = pd.DataFrame(test_data["row_id"])
test_data_id.head(5)
```

Out[786]:

| | row_id |
|---|---|
| **0** | 2 |
| **1** | 8 |
| **2** | 9 |
| **3** | 10 |
| **4** | 11 |

In [787]:
```
output_data=pd.concat([test_data_id, tree_pred], axis = 1)
```

In [788]:
```
output_data.to_csv('output_data2.csv', index = False)
```

In [ ]:

In [ ]:

In [ ]: