Does Your College Decision Determine Future Salary Potential?

Building a Random Forest Classifier

Random Forest utilizes ensemble learning - the opinion of the crowd. Instead of using only one decision tree a literal “forest” of decision trees are used to make the same prediction averaging the forest’s results. Each tree is unique because they are individually created from subsets of the dataset.

The first step in building a Random Forest Classifier in Python was to import portions of the scikit-learn machine learning library: metrics and RandomForestClassifier.

```
import sklearn.metrics as metrics
from sklearn.ensemble import RandomForestClassifier
```

We chose 100 trees with a maximum depth of 25, criterion=“gini” was the default setting.

```
model = RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=25, random_state=0)
```

We printed out the first few rows of the data sets we needed to join in order to run our Random Forest Classifier: College Type and College Salary data with Median Household Income (MHHI).

```
college_type.head()
```

	School Name	School Type	Starting Median Salary	Mid-Career Median Salary	Mid-Career 10th Percentile Salary	Mid-Career 25th Percentile Salary	Mid-Career 75th Percentile Salary	Mid-Career 90th Percentile Salary
0	Massachusetts Institute of Technology (MIT)	Engineering	72200.0	126000.0	76800.0	99200.0	168000.0	220000.0
1	California Institute of Technology (CIT)	Engineering	75500.0	123000.0	NaN	104000.0	161000.0	NaN
2	Harvey Mudd College	Engineering	71800.0	122000.0	NaN	96000.0	180000.0	NaN
3	Polytechnic University of New York, Brooklyn	Engineering	62400.0	114000.0	66800.0	94300.0	143000.0	190000.0
4	Cooper Union	Engineering	62200.0	114000.0	NaN	80200.0	142000.0	NaN

```
college_salary_data.head()
```

	school_name	school_region	starting_salary	mid_career_salary	median_hh_income
0	Stanford University	California	70400.0	129000.0	84189.571429
1	California Institute of Technology (CIT)	California	75500.0	123000.0	84189.571429
2	Harvey Mudd College	California	71800.0	122000.0	84189.571429
3	University of California, Berkeley	California	59900.0	112000.0	84189.571429
4	Occidental College	California	51900.0	105000.0	84189.571429

We merged the two data sets to create ‘merged_data_set’.

```
merged_data_set = college_salary_data.merge(college_type[['School Name','School Type']],left_on='school_name',right_on='School Name')
merged_data_set.head()
```

	school_name	school_region	starting_salary	mid_career_salary	median_hh_income	School Name	School Type
0	California Institute of Technology (CIT)	California	75500.0	123000.0	84189.571429	California Institute of Technology (CIT)	Engineering
1	Harvey Mudd College	California	71800.0	122000.0	84189.571429	Harvey Mudd College	Engineering
2	University of California, Berkeley	California	59900.0	112000.0	84189.571429	University of California, Berkeley	State
3	Occidental College	California	51900.0	105000.0	84189.571429	Occidental College	Liberal Arts
4	Cal Poly San Luis Obispo	California	57200.0	101000.0	84189.571429	Cal Poly San Luis Obispo	State

Next we set the Starting Salary ‘greater than $50,000’ as the value the Random Forest Classifier would initially split.

```
#y = 0+(merged_data_set.starting_salary > merged_data_set.median_hh_income.median())
# Starting Salary $50000
y = 0+(merged_data_set.starting_salary > 50000)
```

Generate dummy variables: turn each category with words into 0 or 1. The first line indicates a numerical value for MHHI, with School Region as California(1) and School Type as Engineering(1).

```
x_dummies = pd.get_dummies(x)
x_dummies.head()
```

	median_hh_income	school_region_California	school_region_Midwestern	school_region_Northeastern	school_region_Southern	school_region_Western	School Type_Engineering	School Type_Ivy League	School Type_Liberal Arts	School Type_Party	School Type_State
0	84189.571429	1	0	0	0	0	1	0	0	0	0
1	84189.571429	1	0	0	0	0	1	0	0	0	0
2	84189.571429	1	0	0	0	0	0	0	0	0	1
3	84189.571429	1	0	0	0	0	0	0	1	0	0
4	84189.571429	1	0	0	0	0	0	0	0	0	1

Fit the Classifier to the training data set. This enables it to learn how to make future predictions for new data points.

```
model.fit(x_dummies,y)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=25, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=100, n_jobs=None,
            oob_score=False, random_state=0, verbose=0, warm_start=False)
```

Imported Numpy and Matplotlib libraries for computing and generating charts.

```
import numpy as np
import matplotlib.pyplot as plt
```

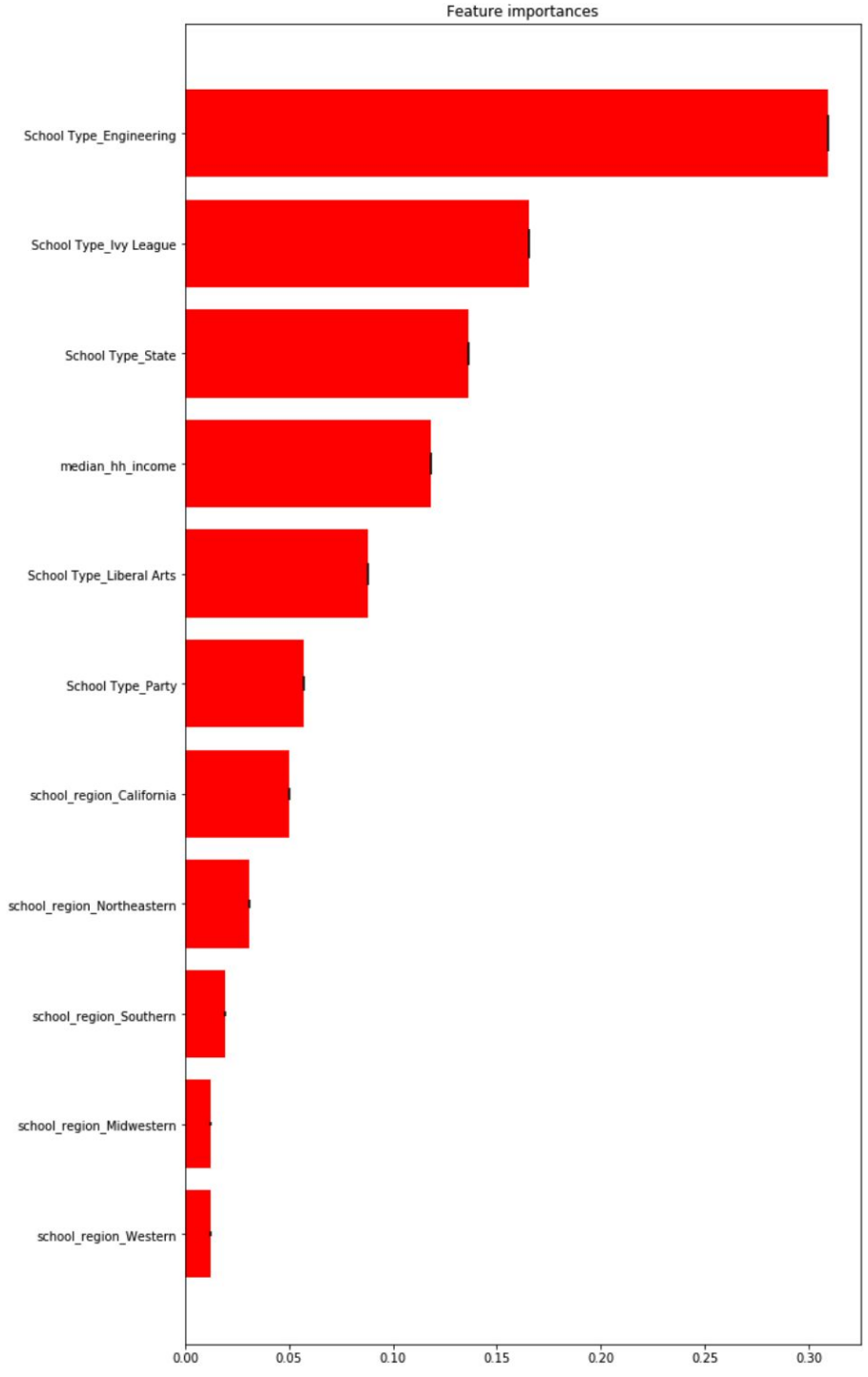
Generated a function to identify, display and plot the important features of the Random Forest Classifier.

```
importances = model.feature_importances_

std = np.std([tree.feature_importances_ for tree in model.estimators_],
             axis=0)
indices = np.argsort(importances)#[::-1]
print("Feature ranking:")

for f in range(x_dummies.shape[1]):
    # print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))
    print("%d. %s (%f)" % (f + 1, x_dummies.columns[indices[f]], importances[indices[f]]))
plt.figure(figsize=(10,20))
plt.title("Feature importances")
plt.barh(range(x_dummies.shape[1]), importances[indices],
       color="r", yerr=std[indices], align="center")
x_labels = [x_dummies.columns[indices[f]] for f in range(x_dummies.shape[1])]
plt.yticks(range(x_dummies.shape[1]), x_labels)
plt.ylim([-1, x_dummies.shape[1]])
plt.show()
```

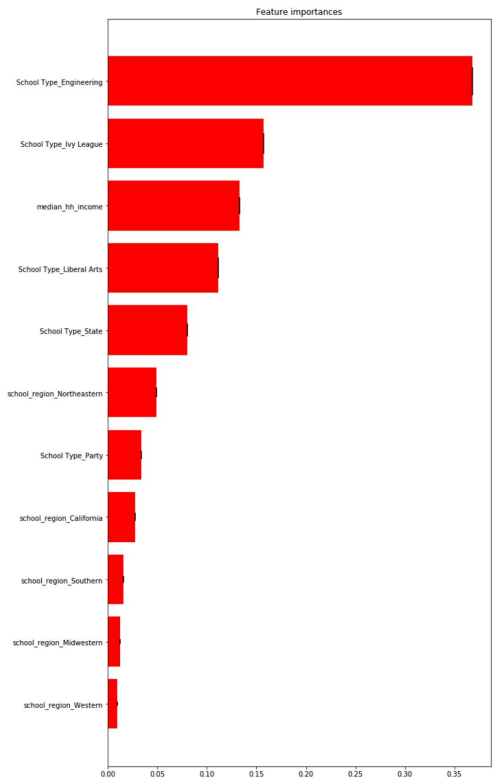
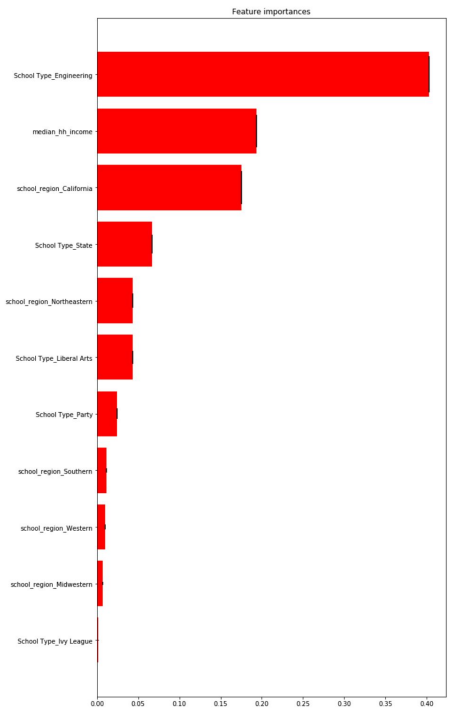
Salary greater than $50,000 in ascending order:

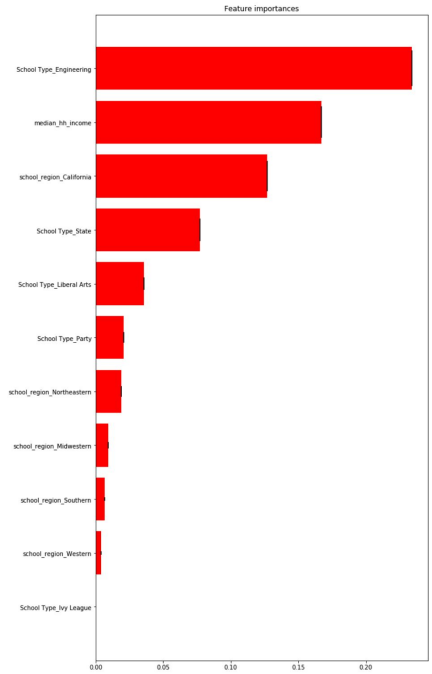
```
Feature ranking:
1. school_region_Western (0.012527)
2. school_region_Midwestern (0.012534)
3. school_region_Southern (0.019408)
4. school_region_Northeastern (0.030767)
5. school_region_California (0.050294)
6. School Type_Party (0.057042)
7. School Type_Liberal Arts (0.087987)
8. median_hh_income (0.118121)
9. School Type_State (0.136164)
10. School Type_Ivy League (0.165644)
11. School Type_Engineering (0.309511)
```

Bar chart for Starting Salary greater than $50,000.



We repeated the same process for starting salary $60,000, $70,000, $75,000, and $80,000. Our goal was to explore how the features would change in importance as Starting Salary increased in value.

Starting Salary > $60,000	Starting Salary > $70,000
	

Starting Salary > $75,000	Starting Salary > $80,000*
	

***** not enough people above $80,000 starting salary to perform the analysis

Table of the Top Five Important Features for Starting Salary:

Starting Salary	Starting Salary: Top 5 Important Features				
$50,000	**School Type Engineering (0.309511)**	School Type Ivy League (0.165644)	School Type State (0.136164)	Median HH Income (0.118121)	School Type Liberal Arts (0.087987)
$60,000	**School Type Engineering (0.368339)**	School Type Ivy League (0.157082)	Median HH Income (0.133210)	School Type Liberal Arts (0.111797)	School Type State (0.080680)
$70,000	**School Type Engineering (0.403373)**	Median HH Income (0.193489)	School Region California (0.175052)	School Type State (0.067024)	School Region Northeastern (0.043677)
$75,000	**School Type Engineering (0.234165)**	Median HH Income (0.166841)	School Region California (0.126346)	School Type State (0.077064)	School Type Liberal Arts (0.035637)
$80,000 *	0	0	0	0	0

***** not enough people above $80,000 starting salary to perform the analysis

Interpretation:

Notice the strength of School Type Engineering. It clearly dominates as the strongest feature across all the Starting Salary ranges. Also, as starting salaries increase the Median Household Income (MHHI) becomes second in importance closely followed by School Region: California.

Mid-Career Salary Model:

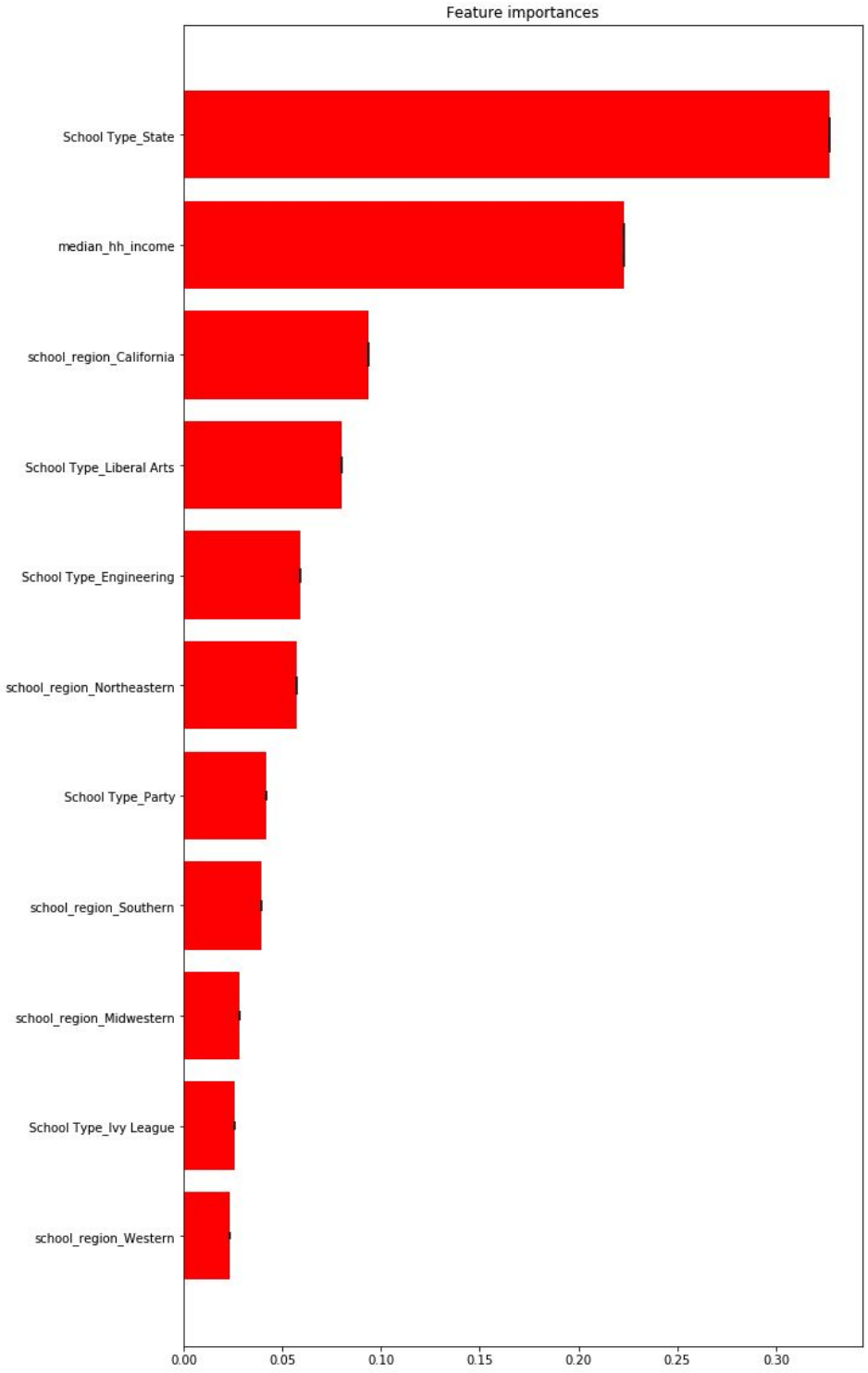
Our next Random Forest Classifier model was set on Mid-Career Salary. Again, our goal was to explore how the features would change in importance as starting salary increased in value.
We began with Mid-Career Salary greater than $80,000. We used the same model but updated the y variable.

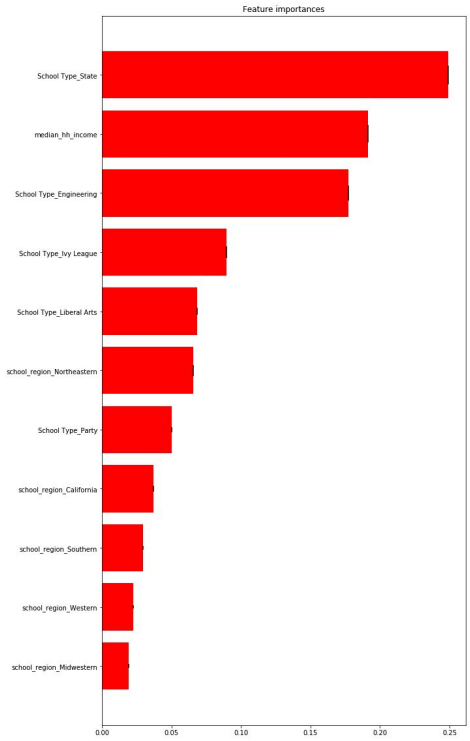
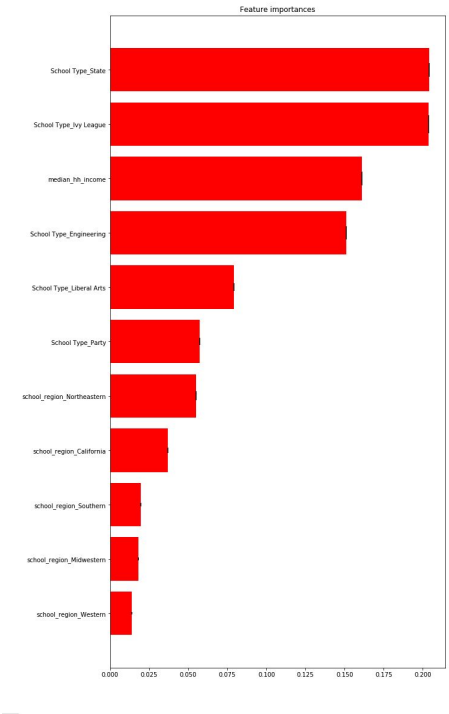
```
# Mid-Career Salary $80000
y = 0+(merged_data_set.mid_career_salary > 80000)
```

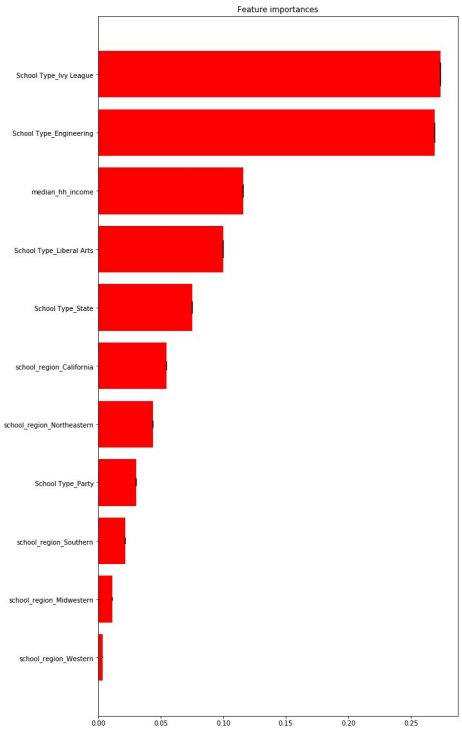
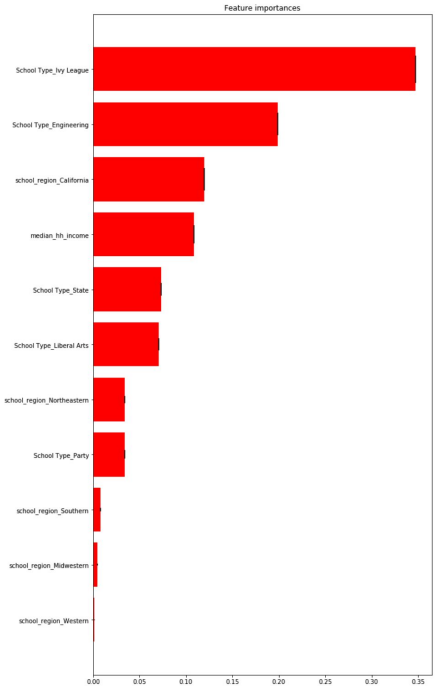
Mid-Career Salary greater than $80,000 feature ranking in ascending order:

```
Feature ranking:
1. school_region_Western (0.023448)
2. School Type_Ivy League (0.026025)
3. school_region_Midwestern (0.028494)
4. school_region_Southern (0.039542)
5. School Type_Party (0.041963)
6. school_region_Northeastern (0.057298)
7. School Type_Engineering (0.059458)
8. School Type_Liberal Arts (0.080048)
9. school_region_California (0.093664)
10. median_hh_income (0.222732)
11. School Type_State (0.327328)
```

Bar Chart for Mid-Career Salary greater than $80.000:



Mid-Career Salary > $90,000	Mid-Career Salary > $100,000
	

Mid-Career Salary > $110,000	Mid-Career Salary > $120,000
	

Mid-Career Salary > $130,000

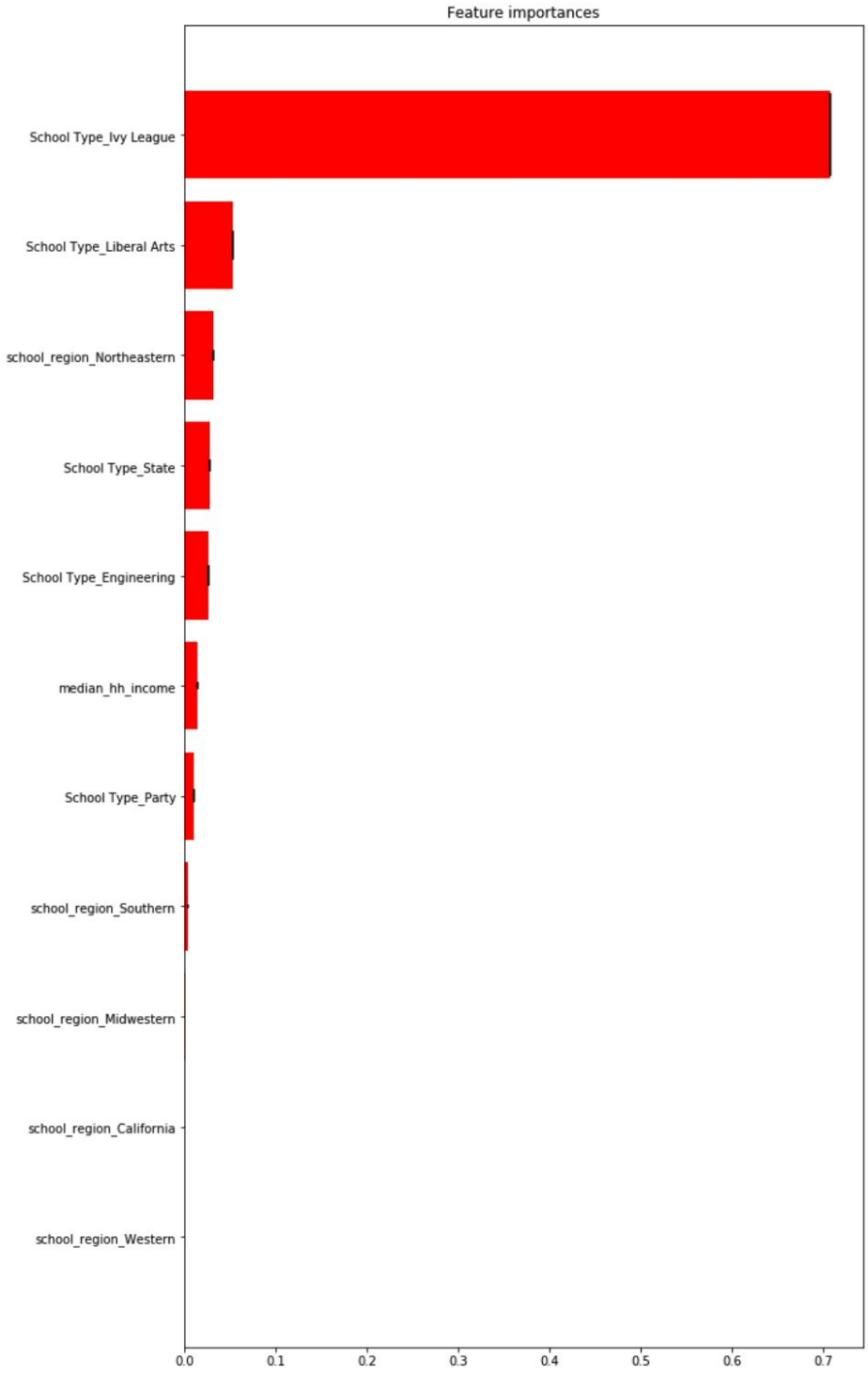


Table of the Top Five Important Features for Mid-Career Salary:

Mid-Career Salary	Mid-Career Salary: Top 5 Important Features				
$80,000	School Type State (0.327328)	Median HH Income (0.222732)	School Region California (0.093664)	School Type Liberal Arts (0.080048)	School Type Engineering (0.059458)
$90,000	School Type State (0.249189)	Median HH Income (0.191393)	School Type Engineering (0.177060)	**School Type Ivy League (0.089492)**	School Type Liberal Arts (0.068492)
$100,000	School Type State (0.204255)	**School Type Ivy League (0.203739)**	Median HH Income (0.161061)	School Type Engineering (0.150921)	School Type Liberal Arts (0.079248)
$110,000	**School Type Ivy League (0.273895)**	School Type Engineering (0.269272)	Median HH Income (0.115758)	School Type Liberal Arts (0.099967)	School Type State (0.075526)
$120,000	**School Type Ivy League (0.347543)**	School Type Engineering (0.198843)	School Region California (0.119591)	Median HH Income (0.108682)	School Type State (0.073566)
$130,000	**School Type Ivy League (0.708447)**	School Type Liberal Arts (0.053805)	School Region Northeastern (0.032031)	School Type State (0.027441)	School Type Engineering (0.027380)

Interpretation:

The $80-90,000 Mid-Career Salary range has State School Type as the strongest feature. This makes sense since the “Mid-Career Salary by College Type” histogram showed State with a heavy distribution in that salary range. However, as Mid-Career Salary increases in value the Ivy League school type becomes the strongest feature. In the $110-120,000 range Engineering School ranks second in importance. Between $120,000 to $130,000 Mid-Career Salary the Ivy League doubles it’s feature importance dwarfing the significance of the other four features.

Conclusion:

We are not sure if the Kolmogorov-Smirnov test was the best choice for comparing the Starting Salary, Mid-Career Salary and Median Household Income values by Region. Do we need to renormalize the data?
Fortunately, the Random Forest Classifier model provided an elegant illustration of how Starting and Mid-Career Salaries were strongly determined by your college type: Engineering, Ivy League, Liberal Arts, State, and Party.

The data strongly suggests that your college choice does make a difference in your starting and mid-career salaries. Ivy League and Engineering colleges definitively result in higher salaries over time. There are also substantial regional differences in salaries. What we aren't certain of is whether the higher salaries in the California and Northeastern regions truly result in a higher living standard or if those higher salary are mitigated by their regions' higher cost of living. Quite possibly, the ideal solution for a college educated individual would be to graduate from an Ivy League or Engineering college and live in a region with a lower cost of living.

All of this data will be of intense interest to our client, ACT, Inc. This will assist them with providing clear data on the importance of school choice to future salaries in their ACT Profile program.

References:

Quandl 3.4.5
Pandas 0.22.0
NumPy 1.14.6
Matplotlib 2.1.2
Statsmodels 0.8.0
Seaborn 0.7.1
Python 3.6.7
SciPy 1.2.0
Cufflinks 0.8.2.
Plotly 3.4.2

https://plot.ly/python/anova/

https://www.investopedia.com/exam-guide/cfa-level-1/quantitative-methods/hypothesis-testing.asp

https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.ks_2samp.html

https://www.kaggle.com/wsj/college-salaries

https://www.kaggle.com/census/estimate-of-median-household-income-group-series

http://www.act.org/content/act/en/products-and-services/act-profile.html

https://matplotlib.org/api/_as_gen/matplotlib.pyplot.xticks.html

https://matplotlib.org/examples/api/barchart_demo.html

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html

https://stackoverflow.com/questions/17506163/how-to-convert-a-boolean-array-to-an-int-array

https://stackoverflow.com/questions/11587782/creating-dummy-variables-in-pandas-for-python

https://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.merge.html

https://pandas.pydata.org/pandas-docs/stable/merging.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html