University of Texas at Austin

Center for Professional Education

Data Science Capstone Report – Predictive Modeling with Python

November 18, 2019

Rory S. Langran

Summary:

The Data Science Capstone project is designed to demonstrate proficiency in utilizing best practices in data wrangling, exploratory data analysis (EDA), feature engineering, feature selection, classifier training and prediction, and classifier parameter tuning. All of these steps are conducted through the Python programming language and leverage powerful modules that provide additional capabilities. Lastly, the Python script and visualizations are published through the GitHub portal, which is a code-sharing platform for users to view open source projects.

The overall objective of the Capstone project is to train, test, and tune a predictive Machine Learning algorithm that can provide an accurate and repeatable prediction (over 85%) using a large dataset. Nested under this objective are three goals: 1) To identify and remove outliers in the data that will improve the accuracy of the model, 2) To improve the accuracy of the predictive model through parameter tuning, and 3) To identify aspects of the data that could potentially improve the accuracy of the model through additional modifications.

Dataset:

I used a census dataset hosted by the University of California – Irvine, Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/Census+Income. This dataset consists of 48,842 records with 15 attributes. It was extracted from 1994 United States census data. The prediction attribute indicates whether the individual earned more than $50K or less than $50K per year. Figure A provides more detail on each of the 15 attributes:

Figure A

#	Dataset Attribute	Data Type	# of Classes	Range	Used in EDA?
1	age	integer	n/a	17-90	yes
2	workclass	string	9	n/a	yes
3	fnlwgt	integer	n/a	12285-1490400	no
4	education	string	16	n/a	yes
5	education-num	integer	n/a	1-16	no
6	marital-status	string	7	n/a	yes
7	occupation	string	15	n/a	yes
8	relationship	string	6	n/a	yes
9	race	string	5	n/a	yes
10	sex	string	2	n/a	yes
11	capital-gain	integer	n/a	0-99999	yes
12	capital-loss	integer	n/a	0-4356	yes
13	hours-per-week	integer	n/a	1-99	yes
14	native-country	string	42	n/a	yes
15	**income**	**string**	**2**	**n/a**	**yes**

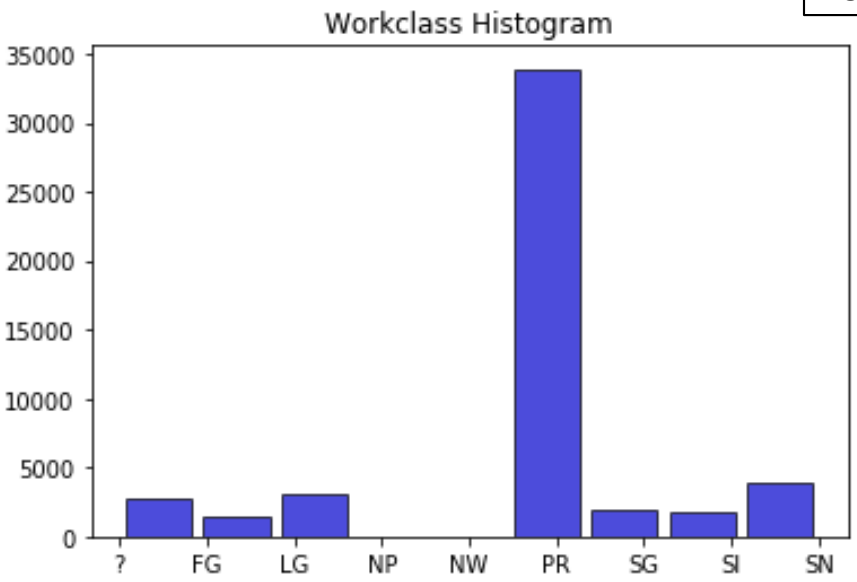
The attributes, “finlwgt” and “education-num” were not used for the subsequent EDA. The “income” attribute represents the predictor of income over or under $50K per year.

Exploratory Data Analysis (EDA):

I created histograms for all of the string attributes in order to visualize the distribution of data. For ease of reading, I relabeled the classes with two or three letter acronyms. There were instances of the character “?” in three of the attributes: Workclass, occupation, and native-country. For the purpose of the EDA these were left in the dataset but are removed as part of creating the final training and testing dataset.

Workclass:

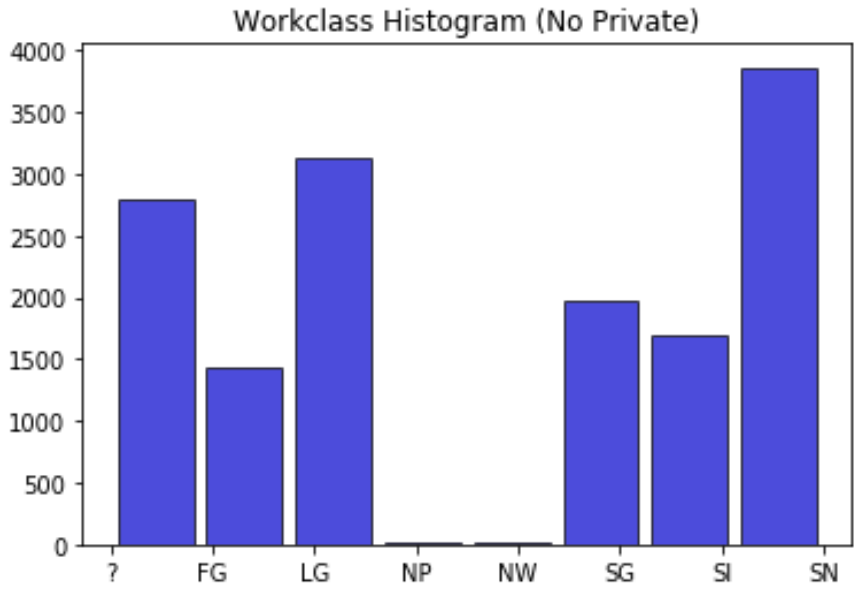
Figure B



Abbr	Full Name
FG	federal gov
LG	local gov
NW	never worked
NP	without pay
PR	private
SG	state gov
SI	self emp - inc
SN	self emp - no inc

The class “PR” (Private) constitutes by far the greatest number of instances with 33,906 in Figure B. If we remove PR from the histogram, it provides a more illustrative comparison for the other classes (Figure C):

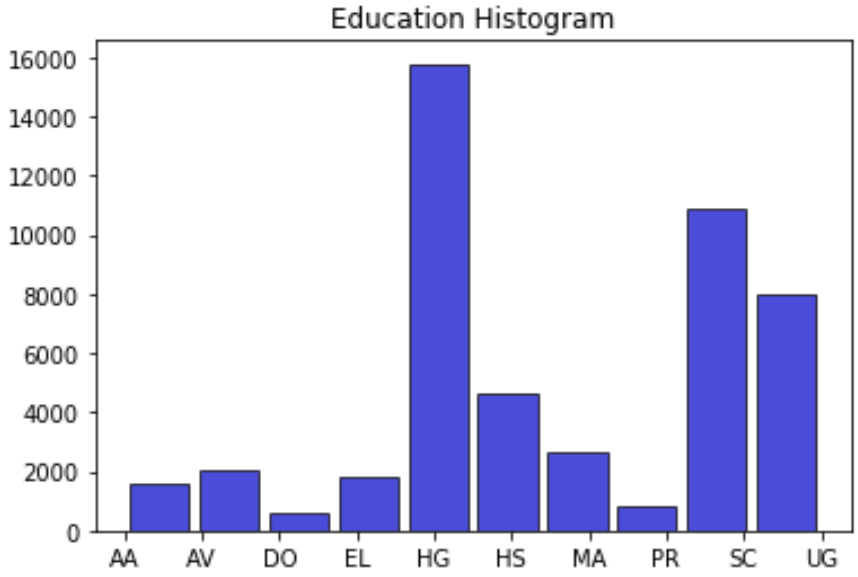
Figure C



The "?" class has 2,799 instances (about 6% of all records), while "NP" (Without Pay) and "NW" (Never Worked) represent only a handful of records (31).

Education:

Figure D

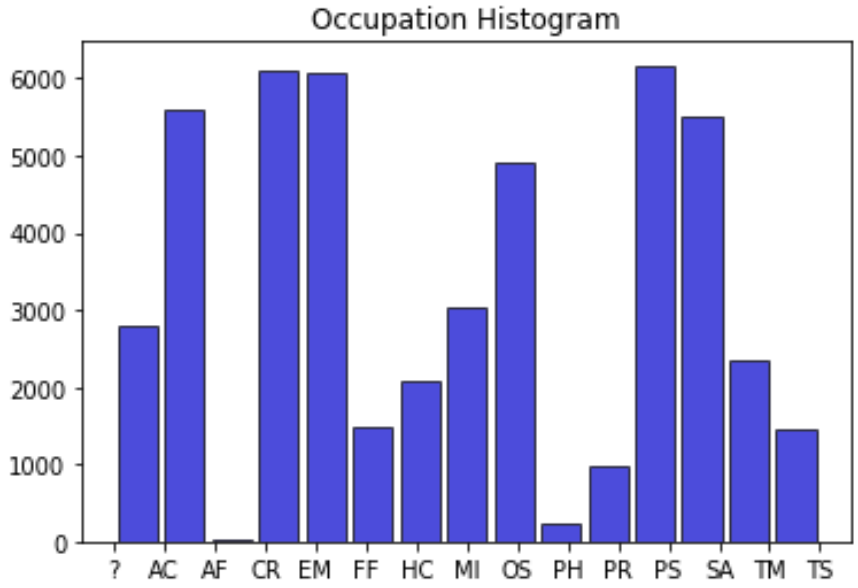


Abbr	Full Name
AA	assoc arts
AV	assoc voc
DO	doctorate
EL	elementary
HG	high grad
HS	high school
MA	masters
PR	professional
SC	some college
UG	undergrad

Education is somewhat more evenly distributed than workclass, but "HG" (High School Graduate) has a significantly larger share of records than the other nine attributes (15,784).

Occupation:

Figure E

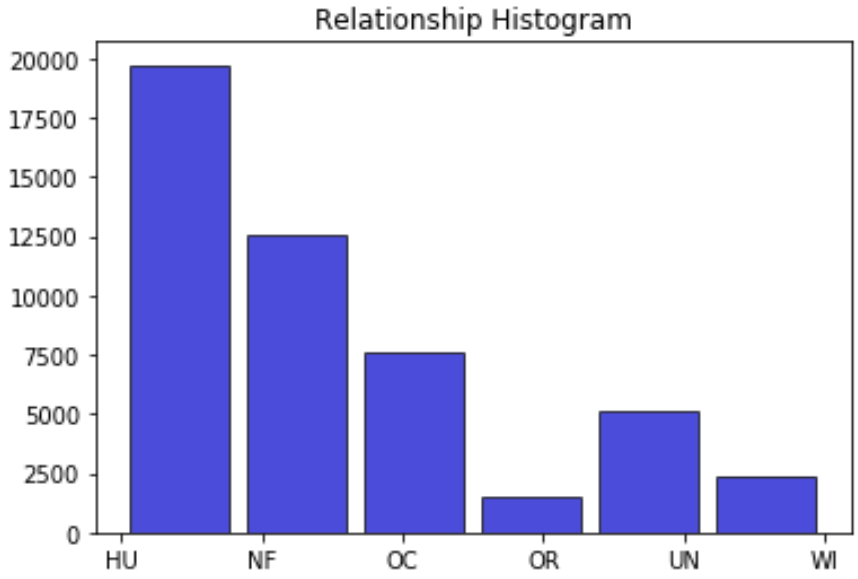


Abbr	Full Name
AC	Adm-clerical
AF	Armed-Forces
CR	Craft-repair
EM	Exec-managerial
FF	Farming-fishing
HC	Handlers-cleaners
MI	Machine-op-inspct
OS	Other-service
PH	Priv-house-serv
PR	Protective-serv
PS	Prof-specialty
SA	Sales
TM	Transport-moving
TS	Tech-support

There were 2,809 records with "?" under Occupation and only 15 instances of "AF" (Armed Forces).

Relationship:

Figure F

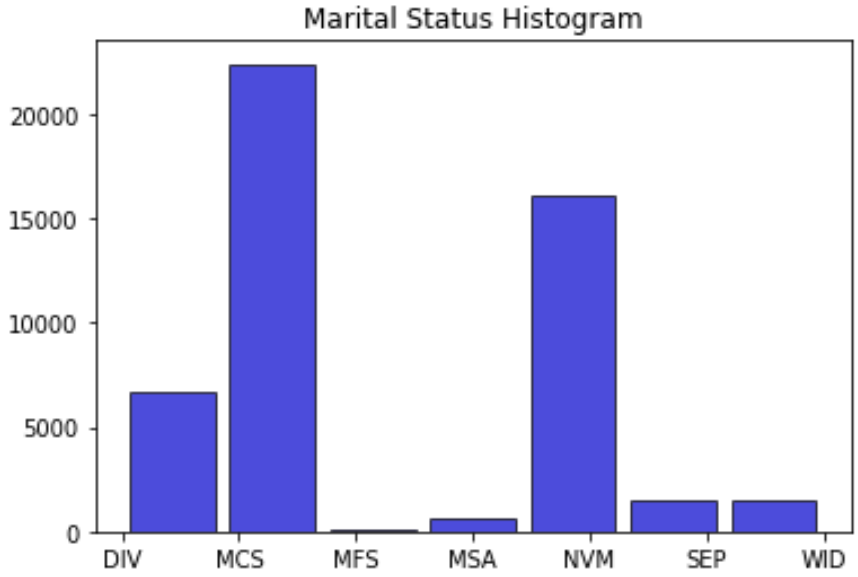


Abbr	Full Name
HU	husband
NF	not in family
OC	only child
OR	other relative
UN	unmarried
WI	wife

"HU" (Husband) had a much larger portion of records under Relationship with 19,716, while "OR" (Other Relative) and "WI" (Wife) had 1,506 and 2,331 respectively.

Marital Status:

Figure G

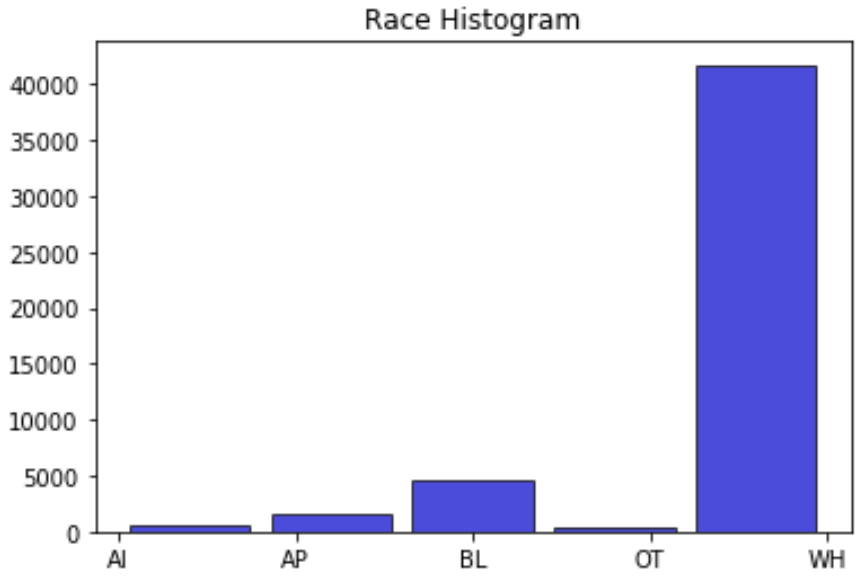


Abbr	Full Name
SEP	separated
DIV	divorced
MCS	married-civ-spouse
MFS	married-af-spouse
MSA	married-spouse-absent
NVM	never married
WID	widowed

Within the Marital Status field, "MCS" (Married Civ Spouse) and "NVM" (Never Married) comprised by far the largest number of records with 22,379 and 16,117 records respectively – these two categories represent almost 79% of all records.

Race:

Figure H

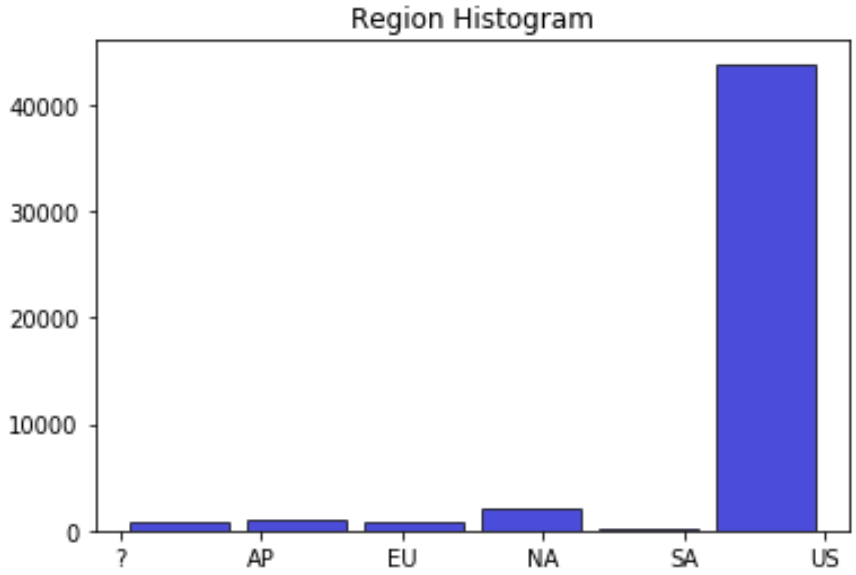


Abbr	Full Name
AI	american indian
AP	asia pacific
BL	black
OT	other
WH	white

The vast majority of records under the Race attribute are comprised of "WH" (White). This category represents 41,762 out of 48,842 records (about 86%).

Native Country:

Figure I

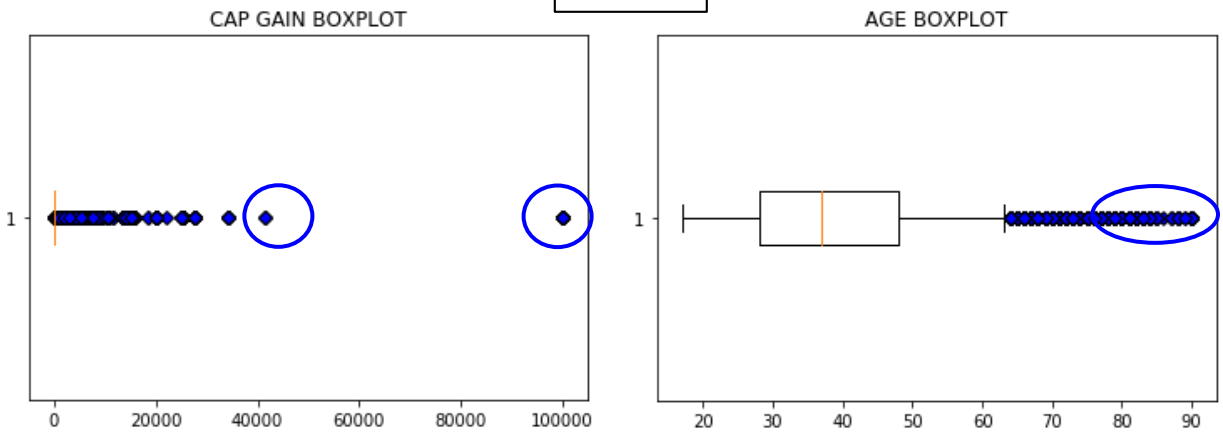


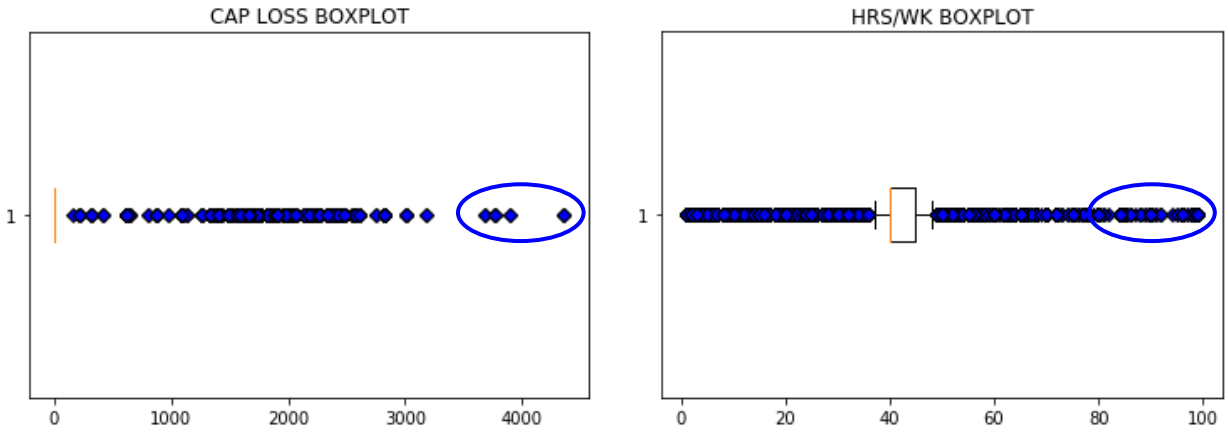
Abbr	Full Name
EU	Europe
AP	Asia Pacific
NA	North America
SA	South America
US	United States

Due to the large number of classes (42) in Native Country, I bundled these into Regions: US (United States), NA (North America), SA (South America), Asia-Pacific (AP), and EU (Europe). As expected, United States dominated the number of records under Native Country with 43,855 records (about 90%). The "?" category consisted of 857 records. Note that there were no countries listed under Native Country from the continent of Africa, nor were the countries of Russia, Brazil, Indonesia, or Australia.

Box Plots:

Figure J





The Capital Gain attribute contained some large outliers (99,999), which was more than 13 standard deviations above the mean (1,079). Capital Loss maximum was 11 standard deviations above the mean (88), followed by Hours per Week (4.9) and Age (3.6).

Feature Engineering:

Feature Scaling:

Prior to data wrangling, I created three new fields with feature scaling from: capital-gain, capital-loss, and hours-per-week. I used the "MinMax" feature scaler with a range of 0 to 1 – this feature scaler is from the sklearn-preprocessing package. The feature scaling provides a structured range of values for the predictive model to work with.

Data Discretization:

The only attribute that I discretized was "age." I established 5 bins and used string labels for each bin. This step is seamlessly accomplished using the pandas package with the "cut" function.

Data Wrangling:

As mentioned previously, there were a number of "?" categories within the "workclass", "occupation," and "native-country" fields. I performed a progressive filter where a new dataframe was create after each cut, thereby narrowing down the dataset based on how many remaining "?" instances there were in each field. This method ensured that I did not miss any records where "?" appeared in two or more of the fields. Figure K shows the number of records removed from the dataset from each field:

Figure K

```
# of ? removed from workclass:  2799
# of ? removed from occupation:  10
# of ? removed from native-country:  811
```

The additional feature engineering consisted of removing outliers. Based on the box plot figures and after initial train-test predictions, I ended up removing the following outliers shown in Figure L:

Figure L

#	Dataset Attribute	Mean	Std Dev	Max	SD above Mean	Outlier Range	# of Records Removed
1	capital-gain	1079	7452	99999	13.3	50001-99999	229
2	capital-loss	88	403	4356	10.6	2501-4356	56
3	hours-per-week	40	12	99	4.9	81-99	291
4	age	39	14	90	3.6	81-90	113

The record count for the final dataset was 44,533 (4,309 records removed).

Recursive Feature Engineering (RFE):

I used two different RFEs: 1) Random Forest with Cross Validation model (RFECV) using a linear regression for the estimator and scoring set to Negative Mean Squared Error 2) RFECV using SVC linear for the estimator and scoring set to Accuracy. The first RFE indicated that all of the attributes were relevant to use for the predictive model. However, the second RFE recommended that I remove "native-country" from the dataset. After experimenting with removing Native Country from the X_train and X_test datasets, I found that the accuracy of the classifiers actually decreased slightly so, I kept this attribute in the final dataset.

Predictive Model Training and Testing:

The train and test datasets were established using a one-step process from the sklearn package – "train_test_split." I split the test set with 20% of the records and used a Random State of 42.

I used three different Machine Learning algorithms: Service Vector Machine (SVM), k Nearest Neighbors (kNN), and Random Forest (RF).

The out of the box results from the training datasets are shown in Figure M on the right:

Census Dataset Number of Training Records: 35,626
Census Dataset Number of Testing Records: 8,907

SVM Model Results

	Over $50K	Under $50K	Total
Over $50K	6345	406	6751
Under $50K	1023	1133	2156
Total	7368	1539	8907
Accuracy:	**83.956%**		

kNN Model Results

	Over $50K	Under $50K	Total
Over $50K	6281	470	6751
Under $50K	1099	1057	2156
Total	7380	1527	8907
Accuracy:	**82.385%**		

RF Model Results

	Over $50K	Under $50K	Total
Over $50K	6241	510	6751
Under $50K	818	1338	2156
Total	7059	1848	8907
Accuracy:	**85.090%**		

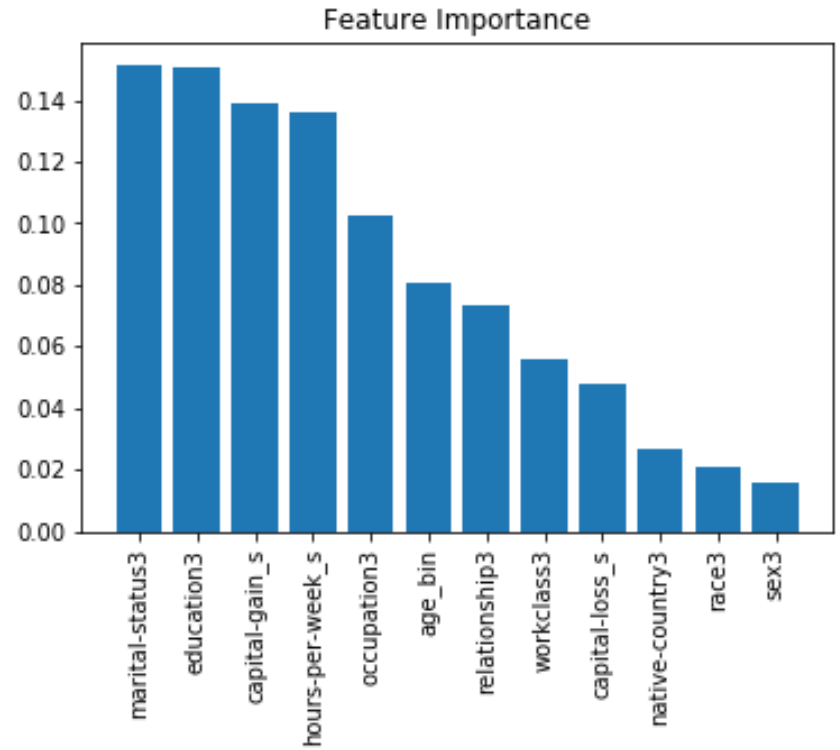
Figure M

I chose the Random Forest model for additional parameter tuning in determine if I could increase the accuracy.

Feature Importance:

Before proceeding with parameter tuning, I ran the "feature_importances" function and created a visualization in order to review the relative importance of each of the 12 features (shown in Figure N):

Figure N



If speed was a consideration for the predictive model, I could consider eliminating "sex," "race," and "native-country" from the dataset and evaluating the impact on accuracy. As it stands, I have already achieved over 85% accuracy with the RF model, so I am satisfied with the final dataset.

Parameter Tuning:

In the "Towards Data Science" portal, there is an excellent article on "hyperparameter" tuning: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74.

I leveraged the technique in the article in order to conduct a search for the best parameters for the RF model. This technique calls for setting up a Random Grid for the parameters and then evaluating the results based on a random selection of parameters that lie between the boundaries. In this case, the following parameters were evaluated:

'n_estimators': Evaluate starting at 200 until reaching 2000 with 10 steps
'max_features': Auto or SqRt
'max_depth': Evaluate starting at 10 until reaching 110 with 11 steps
'min_samples_split': 2, 5, 10
'min_samples_leaf': 1, 2, 4
'bootstrap': True or False

This process is time-consuming due to the number of iterations based on the parameter choices. After completion, the model recommended features that I could use for the final grid:

'n_estimators': 1400, 1600, 1800
'max_features': SqRt
'max_depth': 10, 20, 30
'min_samples_split': 8, 10, 12
'min_samples_leaf': 1, 2, 3
'bootstrap': True

Based on the final parameter grid search, the final settings are:

'n_estimators': 1600
'max_features': SqRt
'max_depth': 20
'min_samples_split': 12
'min_samples_leaf': 1
'bootstrap': True

Before and after parameter tuning results in Figure O shown on right:

Figure O

RF Model Before Parameter Tuning

	Over $50K	Under $50K	Total
Over $50K	6241	510	6751
Under $50K	818	1338	2156
Total	7059	1848	8907
Accuracy:	**85.090%**		

RF Model Using Final Parameters

	Over $50K	Under $50K	Total
Over $50K	6390	361	6751
Under $50K	833	1323	2156
Total	7223	1684	8907
Accuracy:	**86.595%**		

Final Results of RF Model:

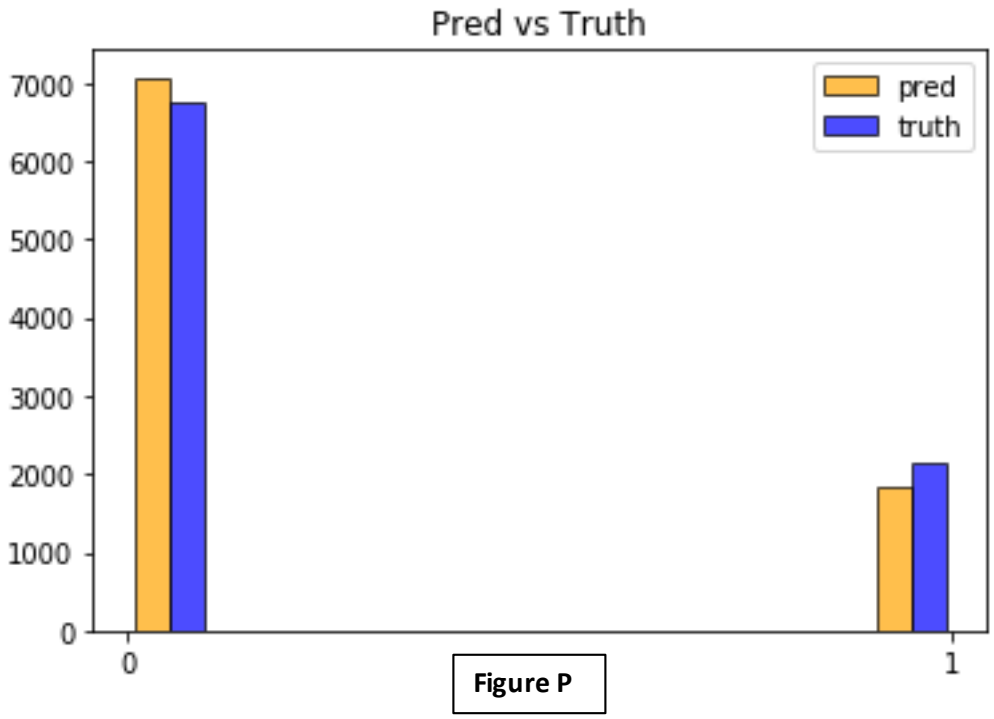


Figure P

Conclusions and Recommendations:

All objectives were addressed through this Data Science Capstone project. Through some robust feature engineering and outlier removal, I was able to achieve an out of the box accuracy of over 85% with the Random Forest classifier.

Objective 1 – Identify and Remove Outliers: During EDA, I found box plots to be very effective in providing an initial visualization of outliers. Additionally, the pandas function "describe" is a simple, one-step command that provides all of the data that pertains to the Five Number Summary. Based on these functions, I was able to further refine the range of outliers by simply doing a count of records based on my filter threshold. While I automatically included the max values, I was also interested in capturing lower values that were still above three times the interquartile range (IQR). Through the use of counting outliers based on experimenting with the ranges, I was able to filter a very small number of records, but significantly reduce the distribution of the results.

Objective 2 – Improve the Model Accuracy Through Parameter Tuning: The "Towards Data Science" article that I cited earlier was instrumental in determining the best parameters to use for tuning the Random Forest model. While each step in the process, i.e. Random Tuning and Parameter Tuning, took about 30 minutes each, this was still a much more efficient process then if I selected each parameter configuration and generated the results and recorded them. While the final accuracy increased by only

1.5%, this was still proof that parameter tuning can generate a more accurate model without changing anything to the dataset.

Objective 3 – Identify Aspects of Data That Could Improve the Accuracy of the Model Through Additional Modifications: Based on the visualizations from the histograms, it was obvious that "native-country" and "race" were overly represented. The category "White" in the Race attribute represented 86% of all the records, while "United States" in the Native Country attribute represented about 90% of the records. I feel that both of these attributes could be refined through additional and/or refined attributes and categories in order to reduce this over representation. For example, the White category includes all of the Hispanic population. In 1994, the Hispanic population was 26.1 million out of 260.3 million (about 10%). The White population as a whole was 216.5 million (about 83%) out of the total U.S. population.[1] Without excluding Hispanics, the White population percentage is still overstated by 3% - if we split White into Hispanics and All Other White, then it will provide a more refined categorization of the race attribute. In terms of Native Country, a better attribute would be one that describes the country of ethnicity, rather than where the individual was originally born. This would eliminate "United States" as a category, but the downside is that it would introduce a large of number of categories, such as "Scots-Irish," "Welsh," "Pomeranian," etc. Even if there were 50 or more categories, it might be preferable than using an attribute where the category represents 90% of all the records. Additionally, the Native Country attribute does not capture a large number of countries – currently there are 54 countries in Africa, including Nigeria (200M), Ethiopia (110M), Egypt (101M) not included. In addition to Africa, the Census data does not include Russia, Brazil, Indonesia, or Australia.

1- Byerly, Edwin and Deardorff, Kevin, "CURRENT POPULATION REPORTS – National and State Estimates 1990-1994," U.S. Department of Commerce, Economics and Statistics Administration, Bureau of the Census. https://www.census.gov/prod/1/pop/p25-1127.pdf