Homework #5

Rohit Garla – rg3365

1 PROBLEM DEFINITION

The goal of this project is to predict salary of a person in the United States based on some features recorded about the person. This is a regression problem and so we need to predict a quantity which in this case is going to be an integer representing the salary in dollars. Some of the possible algorithm we can use to solve a regression problem are the following: linear regression, neural network, random forests, k-NN. We decided to focus on three algorithms: linear regression, neural network and random forest.

2 FEATURE ENGINEERING

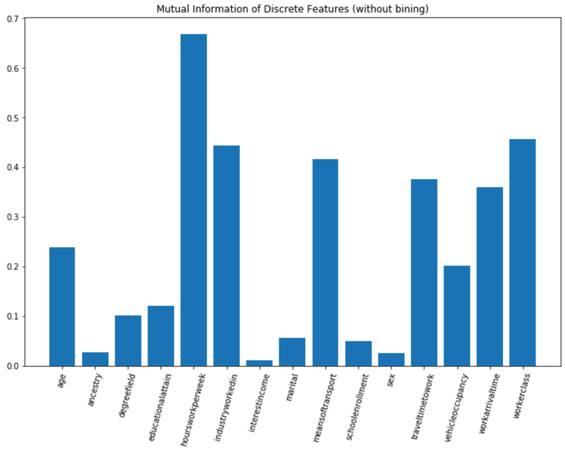
2.1 Dropped Features

Dropped the columns= ‘idnum', 'interestincome', 'schoolenrollment', 'ancestry', 'sex'.

Reason:
Idnum: Idnum is just an additional attributed used for convenience to identify a sample of the dataset, so it will definitely not be included.

Rest of the column: The mutual info of the above columns was low and were removed from the training data and test data.

Mutual Information - Sci-kit learn’s mutual_info_regression () function was used to get the mutual information of each feature with the target variable. The following plot was obtained:



2.2 Data Cleaning (NULL VALUE DEALING)

1 Worker class: “?” was replaced by ‘10’ for people with age less than 16 and ‘11’ for people who never worked and not in labour force and last worked more than 5 years ago

2 Travel time to work: “?” was replaced by 0 for people who work from home. This was calculated using the column of means of transport.

3 Vehicle occupancy: “?” was replaced by 0 for not a worker and were assigned the values 01,02,03,04..10 by finding out the vehicle occupancy from the means of transport column

4 Means of transport: “?” was replaced by ‘0’ for not a worker, not in the labor force,
including persons under 16 years; unemployed, employed, with a job but not at work, Armed Forces with a job but not at work)

5 Educational attain: “?” was replaced by ‘0’ for less than three years old

6 Work arrival time: “?” was replaced by ‘0’ for people who worked from home and ‘1’ for not a worker

7 Hours work per week: “?” was replaced by 0

8 Degree field: “?” was replaced by 0 for less than a bachelors degree

9 industry worked in: “?” was replaced by 0

2.3 Binning

Created bins to convert continuous values to discrete. An categorical value was generated, to represent the range. Following were the bins chosen for each feature:

(a) Work arrival time: 1 - first 2 hours, 2 - second 2 hrs….

(b) Industry worked in: 1 - AGR industry, 2 - EXT industry, 3 – UTL industry ….

2.4 Feature Encoding

The training and test sets contains several categorical features, so they must be converted to a numerical form. To do so we have adopted two different options: integer encoding and one-hot encoding.

Integer encoding is good enough for features whose a natural ordered relationship between each other is present. As an example, hoursperweek has a numerical value, so it was encoded to an integer directly. Also, in the case of columns like worker -class, degree field the representation in the data did not represent any order or hierarchy. So in this case one hot encoding was used.

One hot encoded column: Worker class, vehicle occupancy, means of transport, marital, education attained, sex, work arrival time, degree field, industry worked in.

2.4 Scaling

Scaling – After data frame analysis, it seemed evident that scaling is required in this case as some attributes like ‘interest income’, ‘traveltimetowork’, ‘age’, ’hours per week’ etc. were numeric but very large values. In addition, since most Machine learning models work better on scaled data, scaling every feature was required. Sci-kit learn’s RobustScaler() function was used to scale the features and make the mean 0. This is called “robust” because it also treats the outliers, which can vary for each feature.

3 MACHINE LEARNING MODELS

After the Feature Engineering, the following models were trained and tested on the training data.

3.1 Linear Regression

Linear Regression() package from sci-kit learn was used to implement Linear Regression.

Root mean Squared error for 10 folds – 145387495335.66458

3.2 Neural Network

Neural Network MLP regressor was used from sci kit learn to create a neural network.

For “Relu” activation Function
Root mean Squared error for 10 folds – 38199.845839599184

For “Logistic” sctivation
Root mean Squared error for 10 folds – 47365.18991106058

3.3 Random Forest

RandomForestRegressor( ) package from sci-kit learn was used to implement a Random Forest.
Root mean Squared error for 10 folds – 25964.47067368035

4 HYPER PARAMETER TUNING

After training and testing the models with their default parameters on the training data. Random forest model was chosen for further analysis and prediction.

- Random Search CV was used with CV=5
- {'bootstrap': True, 'max_depth': 30, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 1400} were the best params found

4.1 Random Search

Random Search was performed to obtain the best parameters for the random forest. After hyper parameter tuning it can be observed that there was a significant jump in accuracy.

```
CV SCORES - [1.28281098e+09 2.26557566e+09 7.03266670e+08 1.13683069e+10
 4.91655714e+09 1.01702350e+09 3.12639798e+09 4.14196233e+08
 7.98463083e+09 1.04403915e+09]
```

Root mean Squared error for 10 folds – 20351.811540350336

5 FURTHER IMPROVEMENTS

Classification for 0 and non - 0 wages

- Random Forest Classifier was used to predict the wage when it’s a zero and this prediction was combined with the prediction of the regressor.
- Random Search CV was used with CV=5, to find the best parameters for classification

- Both the outputs were combined as whenever the classier predicted a 0, it would be selected and for non zero prediction the prediction of the regressor was preferred.

References

RobustScaler( ) - http://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html#sklearn. preprocessing.RobustScaler

Mutual_info_regression( ) - http://scikitlearn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression. html#sklearn.feature_selection.mutual_info_regression

Mean_Squared_error( ) - http://scikitlearn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

Linear Regression( ) - http://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

Neural Network – https://scikit-learn.org/stable/modules/neural_networks_supervised.html

RandomForestRegressor( ) - http://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html