BIG DATA ANALYTICS

Midterm

Balamanikandan Gopalakrishnan – Rachan Hegde

Contents

Case 1 Classification & Regression

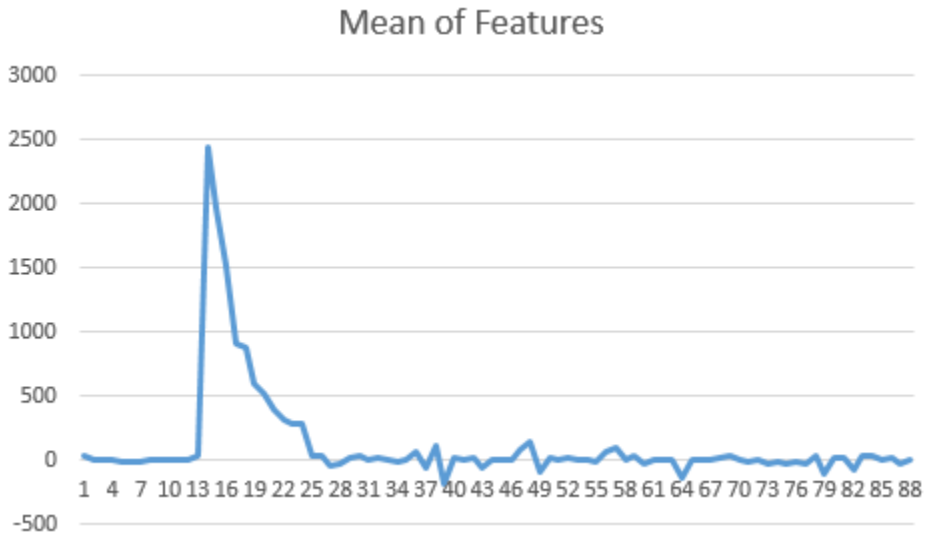
Pre-Processing Steps

- The data was analyzed for missing/ invalid data but the given data set was clean and didn't require any missing data handling
- On detailed analysis of data, it was identified that the #movies before 1965 were relatively lesser than the #movies after 1965. This resulted in poor prediction of movies before 1965 just by using random split
- **Stratified Sampling** with duplicates was used to increase the occurrences of #movies before 1965. This resulted in better prediction results for movies <1965 than the predictions made by using random split

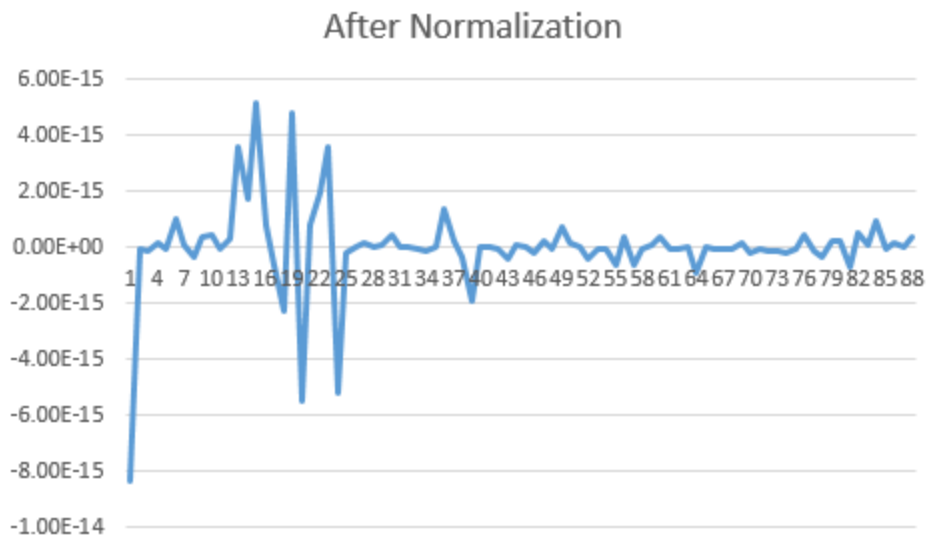
Feature Engineering

Feature Scaling

- The data had to be normalized for consistency and **Scalar normalization** was chosen to normalize the data. Few other normalization techniques were applied but the results were similar



Summarization displaying mean of features before normalization



Summarization displaying mean of features after normalization

Feature Selection

PCA (Principal Component Analysis)

The MLIB features functionality was used to perform principal component analysis to choose top 10 features. This has a signification performance boost and resulted in results similar to that of using the entire sample set.

Note: Class '0' i.e. the movies < 1965 were better classified when No Feature Selection was performed.

Model Selection - Classification

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Step Size: .00001 (For SGD)

The step size was set to .00001 and on increase or decrease of this value for SGD algorithms, the error was increasing

Convergence Tolerance (For LBFGS)

A convergence tolerance was set to .001 from a default of 1E-4 which resulted in reduced error rate.

Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

	PCA (10)	No Feature Selection

SVM With SGD	Training Error: .10 Precision: .89 Recall: .89 Precision (1): .89 Recall (1): 1 Precision (0): 0 Recall (0): 0 ROC: .5 Test Error: .10 Precision: .89 Recall: .89 Precision (1): .89 Recall (1): 1 Precision (0): 0 Recall (0): 0 ROC: .5	
LR With LBFGS	Training Error: .10 Precision: .89 Recall: .89 Precision (1): .90 Recall (1): .98 Precision (0): .42 Recall (0): .09 ROC: .54 Test Error: .10 Precision: .89 Recall: .89 Precision (1): .90 Recall (1): .98 Precision (0): .42 Recall (0): .09 ROC: .54	Training Error: .09 Precision: .90 Recall: .90 Precision (1): .92 Recall (1): .97 Precision (0): .55 Recall (0): .29 ROC: .63 Test Error: .09 Precision: .90 Recall: .90 Precision (1): .92 Recall (1): .97 Precision (0): .56 Recall (0): .29 ROC: .63
LR With SGD	Training Error: .10 Precision: .89 Recall: .89 Precision (1): .89 Recall (1): 1 Precision (0): 0 Recall (0): 0 ROC: .5 Test Error: .10 Precision: .89	

	Recall: .89 Precision (1): .89 Recall (1): 1 Precision (0): 0 Recall (0): 0 ROC: .5	

ML Pipeline Classification

The predictions using ML Pipeline were almost similar to MLIB

- Reg Param (0.1, 0.01)
- Max Iterations (5, 10, 20)
- Convergence Tolerance (.00001, .001, .0001)
- Number Folds (4)

	PCA (10)
Logistic Regression	Training Error: .10 Precision: .89 Recall: .89 Precision (1): .89 Recall (1): .99 Precision (0): .38 Recall (0): .012 ROC: .5 Test Error: .10 Precision: .89 Recall: .89 Precision (1): .89 Recall (1): .99 Precision (0): .38 Recall (0): .014 ROC: .5

Model Selection - Regression

The following configurations were used for the algorithms tested using

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Step Size: .001

The step size was set to .001 and on increase or decrease of this value, the RMS Error was increasing.

Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

MLIB	
LR with SGD	Training RMS Error: 5.28 Test RMS Error: 5.27 (Step Size: .01)

ML Pipeline Regression - Regression

Regression predictions were better for ML Pipeline.

The following were the attributes used for cross validation

- REGPARAM (0.1, 0.01)
- Max Iterations (5,10,20,100)
- Convergence Tolerance (.00001, .001, .0001)
- Number of folds (4)

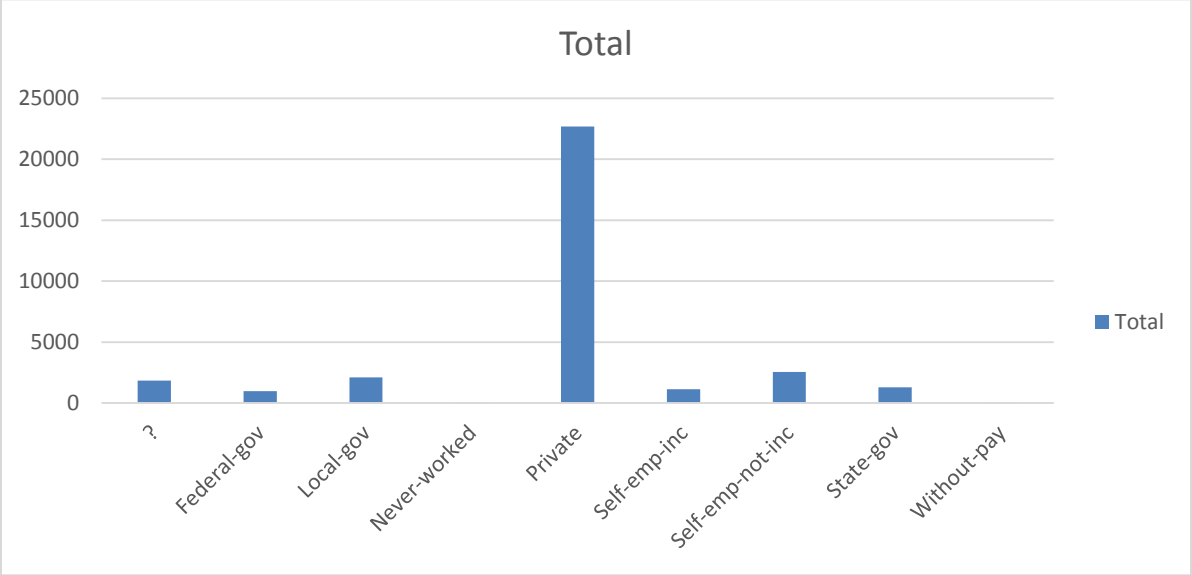
Spark ML	
Linear Regression	Training RMS Error: .28 Test RMS Error: .28

Case 2 Classification

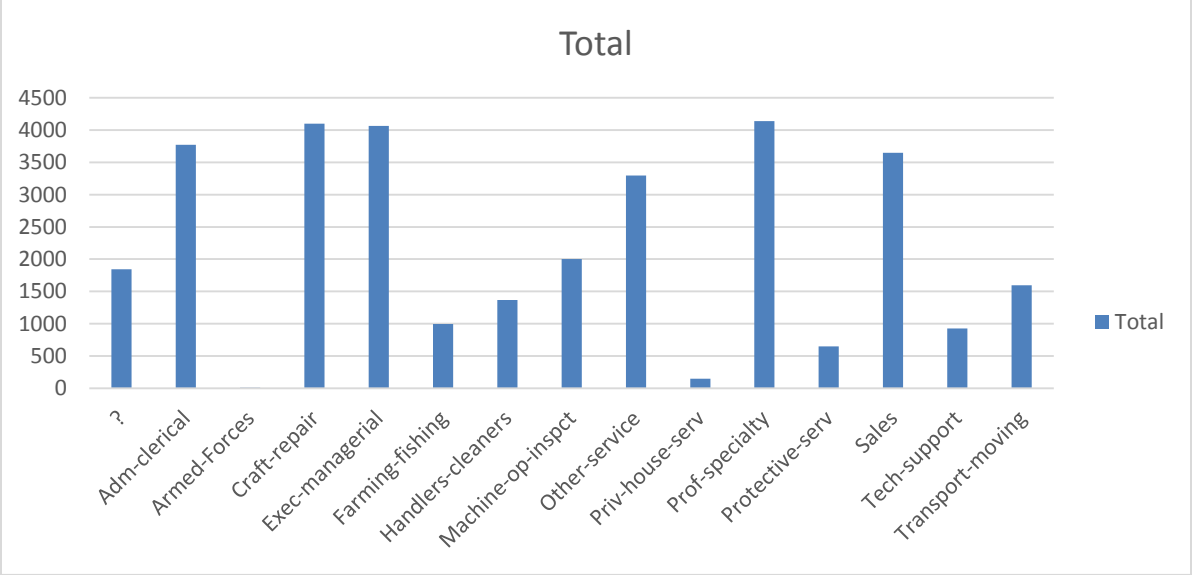
Classification

Pre-Processing Steps

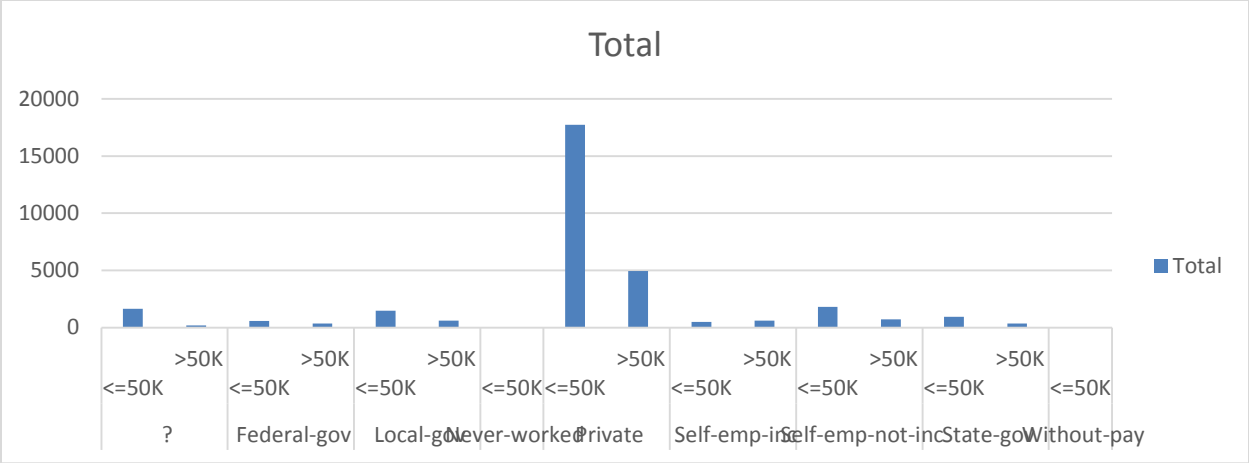
- The data was analyzed for missing/ invalid data and the following columns had missing values
  - Country
  - Work Class
  - Occupation



Split by work class



Split by occupation



Split by work class and salary

On detailed analysis of data,

- The missing work classes were replaced as 'Private' as there were larger pool of private data and the characteristics seemed to match and would cause minimal impact to the predictions
- The missing occupations were replaced as 'Other Services'
- The missing countries were replaced as 'United States' as it had the most number of records (Mode was considered in this case)

Feature Engineering

Feature Selection

The categorical variables were indexed using map functions and categorical information was passed to the decision trees which handled the transformation of categorical data into columns for prediction

Model Selection - Classification

Data Split: 60, 40

The data was randomly split and 60% was used for training the model and 40% was used for testing the created model.

Step Size: .01 (For SGD)

The step size was set to .01 and on increase or decrease of this value for SGD algorithms, the error was increasing

Intercept: True

This optimization attribute was set to true to enable algorithm to use intercepts while creating the model which reduced the error. This is also enabled to reduce the prediction skewedness.

Decision Trees	Training Error: .22 Precision: .85 Recall: .85

	Precision (1): .77 Recall (1): .53 Precision (0): .86 Recall (0): .95 ROC: .74 Test Error: .14 Precision: .85 Recall: .85 Precision (1): .77 Recall (1): .53 Precision (0): .86 Recall (0): .95 ROC: .74

ML Pipeline – Classification

The predictions in ML Pipeline were slightly better than that of MLIB. Scalar Indexer and Vector Indexer were used to convert categorical variables

- Max Depth (5)
- Max Bins (64)
- Min Instances Per Node (1)
- Min Info Gain (0)
- Checkpoint Interval (10)
- Max Categories (41)

Decision Trees	Training Error: .14 Precision: .85 Recall: .85 Precision (1): .79 Recall (1): .52 Precision (0): .86 Recall (0): .95 ROC: .74 Test Error: .14 Precision: .85 Recall: .85 Precision (1): .77 Recall (1): .53 Precision (0): .86 Recall (0): .95 ROC: .74

Case 3 Clustering

Pre Processing

- The data from different sources were aggregated into a single RDD in LIBSVM format.
- There were some missing data but the data type (LIBSVM) automatically handles null values.

Summarization

K-Means clustering algorithm was used on the consolidated data for clusters. The number of clusters was initially set to 2 and was iteratively increased till 10 and the corresponding WSSSE values were observed.

K-Means

K	2	3	4	5	6	7	8	9	10
WSSSE	2.1025	1.27832	9.4	8	6.55	5.86	5.41	5.2	4.72

This represents the K vs WSSSE (Elbow Chart) and when K is 5, the WSSSE seem to stabilize.