QUESTION 1

INPUT:
I have used the files 'Medicare_Provider_Charge_Inpatient_DRG100_FY2011.csv' and 'Medicare_Provider_Charge_Outpatient_APC30_CY2011_v2.csv' which have the most granular data available for the analysis.

Data Cleansing:
Used the xls file with tab as delimiter (Data has some comma inside some columns)

Approach:
1) Group the average costs by the procedure ids for all the procedure and for each procedure id relative variance = variance /mean (Refer to DatasciencesChallenge.zip Problem1 – analysis_1a.py and Hive.sql)

2)
a) Calculated the average estimated_average_submitted charge for given provider and procedure,
b) Calculated the max of the average estimated_average_submitted charge for each procedure.
Based on a) and b) calculated the top three providers having more max average_estimated submitted charge for procedures. (Refer to DatasciencesChallenge.zip Problem1 – analysis_1b.py and Hive.sql)

3)
a) For a given region and procedure calculated
(sum(Average_Estimated_Submitted_Charges*outpatient_services))/(sum(outpatient_services))
b) Calculated the max of the equation above.
c) Based on the a) and b) calculate the top three regions claimed the highest average amount for the largest number of procedures (Refer to DatasciencesChallenge.zip Problem1 – analysis_1c.R and Hive.sql)

4)
a) For a given provider the difference in estimated charges submitted and average total payments is calculated.
b) Calculated the max difference per procedure
c) Based on a) and b) calculate the top three providers have max difference per procedure for largest number of procedures. (Refer to DatasciencesChallenge.zip Problem1 – analysis_1d.py and analysis_1d.R and Hive.sql)

Time Spent: 8 hours

Software: Python and numpy libraries

Testing and Validation: is done by solving the same problem through R and Hive

Attached are the Python, R and Hive scripts

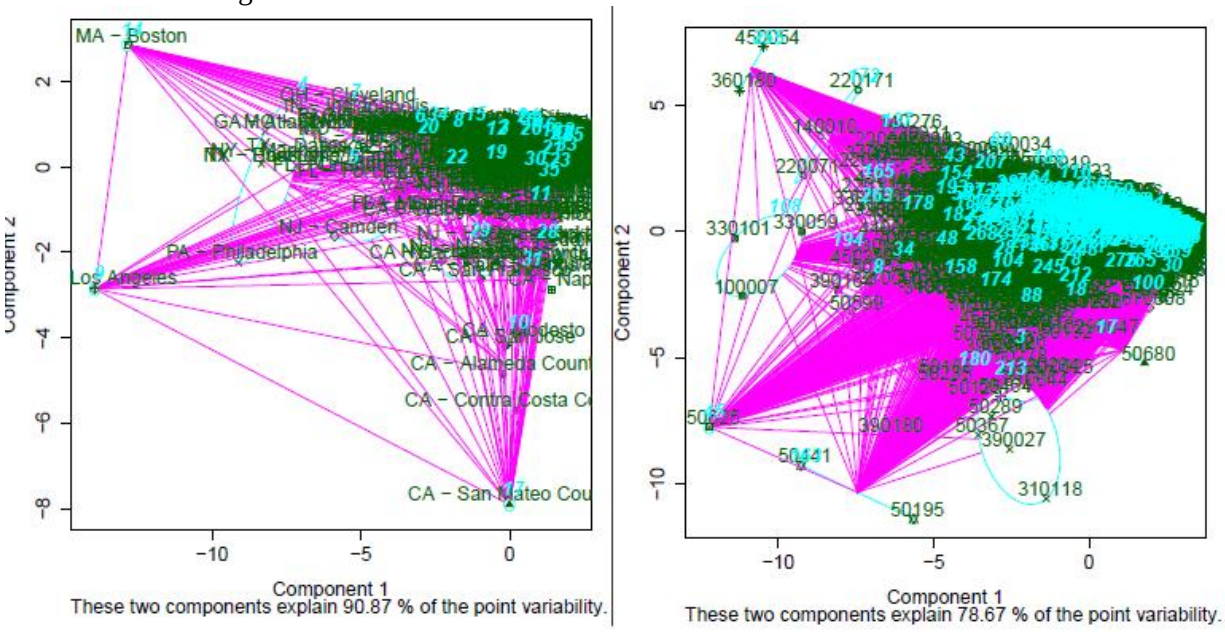
QUESTION 2

Approach: We can consider the problem no:2 as outlier detection problem where a particular provider or Region stands out different from other class points in one or more dimensions.

Feature selection: For each provider or Region we shall have additional important features like TotalCost , TotalPayment, TotalServices , TotalProcedures, TotalAPC, TotalDRG obtained by summing up over respective provider ids. We neglect the correlation between provider and Region.

Model Selection criteria: Since there are more than 4 features or dimensions possible for each provider or Region, each point cannot be understood or represented in 2d or 3d plot directly unless any transformations are applied to these points. histograms and box plot digrams would help us understand only the pairwise interactions between the variables. Hence we resort to applying dimension reduction techniques so that points can be represented on a graph and can be differentiated manually. We shall reduce this problem to finding outlier in 2 dimensional graph.
Algorithm: We apply principal component techniques to both part 2a and part 2b. The variability of each variable is different and hence data is scaled first.PCA is applied on scaled data. The results are interesting in the fact that the first two principal components capture 80% and 90% of the variability in the data for providers and regions respectively. Now to visualize the outliers on two dimensional graph, cluster analysis was helpful. We use scree plot i.e. a graph of "Group sum of squares" vs "number of clusters" to estimate the right number of clusters that helps in acheiving the classification. Here "Group sum of squares" is calculated using kmeans clustering.
Software : Applying princomp(data=data, COR = TRUE) method in R we shall get principal components which are fewer in number to dimensions. Since the top two principal components hold most of the data we plot 2d graph between them and identify the points that appear like outliers.
After estimating the appropriate number of clusters we use CLUSPLOT(x) function in R that gives a very good visualization of clusters with labels of points and clusters and clusters bounded by ellipses
Testing and Validation techniques: The outliers identified in pca analysis also appear as outliers in pairwise plots between variables.

The graphs with pair Wise interactions as well as principalComponents interactions are attached in the code. Also PCA diagrams are shown below in miniature size.



Refer to DatasciencesChallenge.zip Problem 2 – analysis_2a.R, analysis_2b.R, clusPlot_providers.pdf, clusPlot_Regions.pdf, PCAplotofRegions.pdf, PCAplotofprovider.pdf, scatterPlotofCostVsFeatures.pdf and TotalCostvsothersPerRegion.pdf

Total time spent: 2 Days

WHY THREE PROVIDERS ARE DIFFERENT:
50625,100007, 330101
The three providers mentioned in the part2a.csv appear as outliers in the PCA cluster analysis. When we look deeper for the reason it appears that total variability in the variable totalSumCost of providers dominates the other variables. Since the major principal components align along the direction of the maximal variability of the data the data points which have extreme values of totalSumCost stand out in the graph attached (ClusPlot_Providers.pdf)

WHY THREE REGIONS ARE DIFFERENT:
Los Angeles,CA
Boston,MA
San Mateo County,CA

PCA analysis gives the above three regions to be different from others. The first point Los Angeles,CA has very high TotalSumCost compared to other Regions. The second point Boston,MA has very high number of APC procedures and as a result has high number of services and TotalPayments. The third point San Mateo County,CA interestingly doesnt have any extreme values for any of the above variables. But it shows up as an extreme point in the graph. But one can recall that this point is also one of the answer for 1c i.e. ranks third among 336 regions in collecting highes average claim amount for maximum number of procedures. PA − Philadelphia was also a potential outlier as it stands far away from other cities in PA. Outliers can be seen in (ClusPlot_Regions.pdf)

Tools: R, Manual and Hive

QUESTION 3

PART A

Major Technologies used
1) **Python** – For data exploration on small scale, For running Classification algorithms and creating models and loading to HDFS using Hadoop streaming.
2) **Hive and Hadoop** – Data exploration on large scale, Data preparation, data validation and Data Analysis
3) **Shell Scripting** – For reading .ADT files and converting to .csv files.

STEP 1:

Convert the .ADT to csv file and move to hdfs
Snippet of code
```
for   line in 01 02 03 04 05 06 07 08 09 10 11 12
do
tr -c '\40-\176' ',' < PCDR11${line}.ADT | sed 's/,,,/\r\n/g'  > PCDR11${line}.csv
hadoop fs -put  PCDR11${line}.csv /tmp/ds/pcdr
done
```

STEP 2:

Hadoop streaming using Python - convert the patient XML to csv

Mapper – pntd_mapper.py (attached)
Reducer – pntd_reducer.py(attached)

To Run the Hadoop stream:
export HADOOP_STREAMING_LOCATION=/x/home/nchayapathi/hadoop2/pig-0.10.0-cdh4.2.0/test/e2e/pig/lib/
hadoop jar $HADOOP_STREAMING_LOCATION/hadoop-streaming.jar \
-D stream.non.zero.exit.is.failure=false -D mapred.reduce.tasks=0 \
-inputreader "StreamXmlRecordReader,begin=<rows>,end=</rows>" \
-input /tmp/ds/pntd/PNTSDUMP.XML -mapper mapper.py \
-file mapper.py -reducer reducer.py \
-file reducer.py -output /tmp/ds/tgt_pntd/pntd -numReduceTasks 1

STEP 3:

Data Preparation
Let review data be called as positive patient data and unlabeled data as negative data

Load the patient, procedure data and review data.
1. Patient data denormalize on age (each group as column) , gender (M and F) as column and income (each group as column)
2. Procedure data has 300000003 records and patient data has 100001224 (on an avg. 3 procedures on each patient.)
3. Denormalize the procedure data (each procedure as column -140 columns) and group by patient id – results in 100001224 records.
4. Join Procedure data and Patient data.
5. Join d) with review to get positive data (going forward I will refer review data as positive data
6. Get around 5 Million records which are not part of the review data which will be used in training and testing data.
7. Review data is split in the ratio of 70:30 (35,000 and 15,000)
8. 35,000 positive data + 1 million records from 5 Million records from 6) will be used as training data.
9. 15,000 positive data + 2 million records from 5 million records from 6) will be used as testing data

Basic checks for quality of data
1. Distinct values on age and gndr and inc has given value as expected and some null values. Apart from that there seems to be no noise.
2. count of patient data - 100001224 (patient information)
3. count of procedures - 300000003 (procedure information)
4. count distinct id's in procedure data - 100001224
5. So no data loss in any transformations or transition of data across systems
6. There are around 230K records which have date as 0002-MM-DD

Data Imputation:
All the null fields in the positive data after joining with procedure data are imputated by most occuring value in that column so that the record adds value to the training data.
Similarly missing columnar value in unlabeled data is imputated with most occuring value in that column.

Tackling Approach:
Since this a classic problem of Positive and large scale Unlabeled learning(PU Learning), I have referred to "Learning Classifiers from Only positive and Unlabeled Data" by Charles Elkan.
I have compared their approach (PU algorithm) to that of a normal binary classifier(Random Forests) and found that PU has a better F1 score compared to that of any normal binary classifier.

But the question only asks for 10,000 out of 100 million points which is 0.01% of total data set. So feature engineering was done with emphasis on improving precision than recall of the classifier.

Feature Engineering:

The demographic features(age, gender, inc) are almost identically distributed between all the review patients. The procedure distribution showed very interesting distinctions between positive and unlabeled data with numberofDRG's being much higher for positive patients and with mostly zero APC procedures.

Model Selection criteria:

This hypothesis is tested out with RandomForest classifier in python using sklearn library and feature importance methods available with RandomForest classifier.

All the four features (numberOfDRG, numberOfAPC, hasAPC, hasDRG) have a signficant feature importance value and put together gave better precision. Since there are lot of data points for training set I have tried to include as many features as possible along with the four. The possibility of overfitting is eliminated by choosing bagging in RandomForest. Ensembling helped a lot in choosing the best features and give more precision. RandomForest also has predict probability method which is very helpful at the end to improve the precision.

Data Leakage:

Out of curiosity, I have checked if PatientId adds anything meaningful to the classifier and infact it stands out as an excellent distinguishing feature to classify the data. I think it is a data leakage or preparation problem where all review patients are taken from similar or close ids.

STEP 4:

Creating & Running Classification

1. Since I have to get 10,000 records which needs review goal is to make sure negative data should not be predicted as positive
2. Decided to compare between RandomForestClassifier, Naive Bayes and SGD classifiers.
3. Tried different ratios of positive data vs negative data for training and found out good results on 35,000 positive data + 1M negative data since I have make sure negative data is not predicted as positive.
4. On the test data I got good f1, precision and recall values.
5. Ran the RandomForest Classification model on 100 M records and based on the high probability factor have taken 50,000 records.
6. With 50,000 as the input data ran the RandomForest and Naive Bayes models and got the best 10,000 records and probability of prediction is more than 0.97 for these records.

STEP 5

Data Validation and manual testing

Cross Validation is taken care by ensembling techniques (bootstrapping) in randomForest

1. On the selected 10,000 records made sure there is no noise.
2. Checked the probability of predictions on review data and 10,000 records and made sure there are of high probability numbers.
3. Ran different models on this positive data and made sure all these classifiers predicted as positive with high probability.

STEP6

Performance and scalability

1) Based on the training data Model is created which stored as "classification.pkl". (Python as well Apache Mahout I am able to create the model file)
2) Since the model has to run on large scale data (in our case 100 Million records). Converted the Python scripts in Hadoop Streaming and the model file is copied as Distributed cache on all the nodes.
3) Creating the Model based on the training data (35,000+1M) took around 20 minutes and using Mahout it took around 5 minutes (running the classifier using mahout didn't work).
4) Runn
5) ing the classifier on python using Hadoop Streaming for 100 Million on randomforest model took around 6 hours (4 nodes machine, 64GB RAM on each node with other jobs running on this servers).
6) Using Naive Bayes Classification the model creation was way faster and running model on the entire data took around 1.5 hours (using Hadoop streaming).
7) The accuracy, Recall and Precision values are high for RandomForest compared to Naive Bayes and SGD (tested on smaller subset of data).

Scripts in DatasciencesChallenge.zip problem3 Folder:
Hive.sql – code related to Data preparation, validation and Data Analysis
generatModel.py – for creating the model based on the train data and has capability of running the test data.
runClassifier.py – for running the test data.
runClassifierMapper.py – Hadoop Streaming (Mapper)
runClassifierReducer.py – (Reducer)
importantcommands.txt – Mahout and Hadoop streaming commands.

PART B

The model has given high confidence on some of the features like number Of DRG, number Of APC, hasAPC, hasDRG. Manually verified the same and confirmed the data the number of DRGs being high for the positive data and almost negligible for negative data.

Review Data

- Number of APCs greater than 1 for patients around 500
- DRGs less than APCs – around 500

Result Data

- Number of APCs greater than 1 for patients is 0 and more 1 is 230
- DRGs less than APCs - 0

SYSTEM CONFIGURATION:

- 4 –node
- 64GB RAM on each node, 2-cores, 32 threads
- Linux Operating System
- Python version 2.6.*
- Hadoop (hdp2.0)
- Hive 0.9
- R 3.0.2
- Mahout 0.9

TOTAL TIME SPENT:

While 50% of the time was spent of preparing data convenient enough to run a classifier, 40% of the time was spent on feature selection comparing the distributions of the features between the two labels.

Remaining 10% time was spent on comparison between the classifiers and choosing the randomForest classifier as the most suitable one and evaluating the probabilities.

15+ Days