Big-Data Systems and Intelligent Analytics

Team 4:
Samir Sharan
Harshit Shah
Jeevan Reddy

Case 1: Predict the Year of the song

- 2 Approaches
    - Classification
        - Less than 1965
        - Greater than 1965
    - Regression
        - Predict accurate year

Data Exploration & Summarization

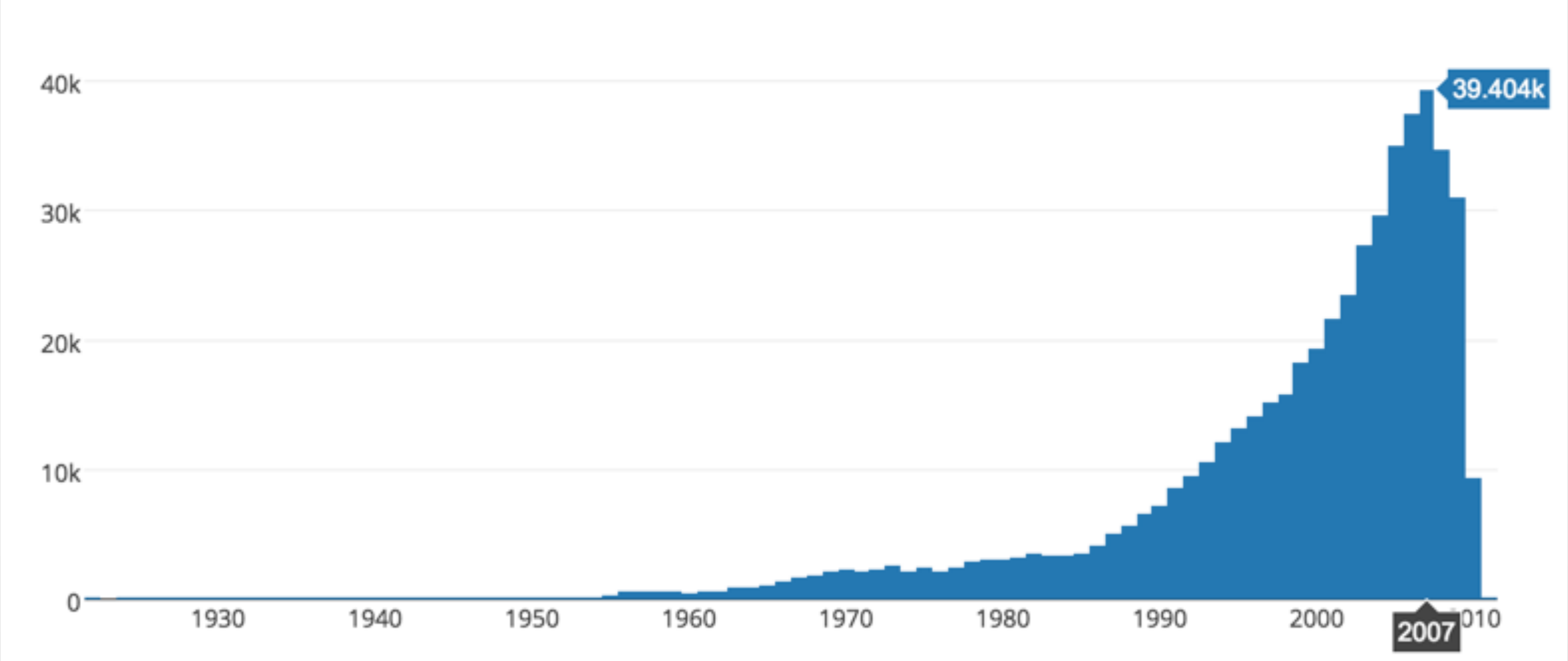
- Sampling
  - 100 records

- Summary Statistics
  - Mean: 1999
  - Min: 1964
  - Max: 2010

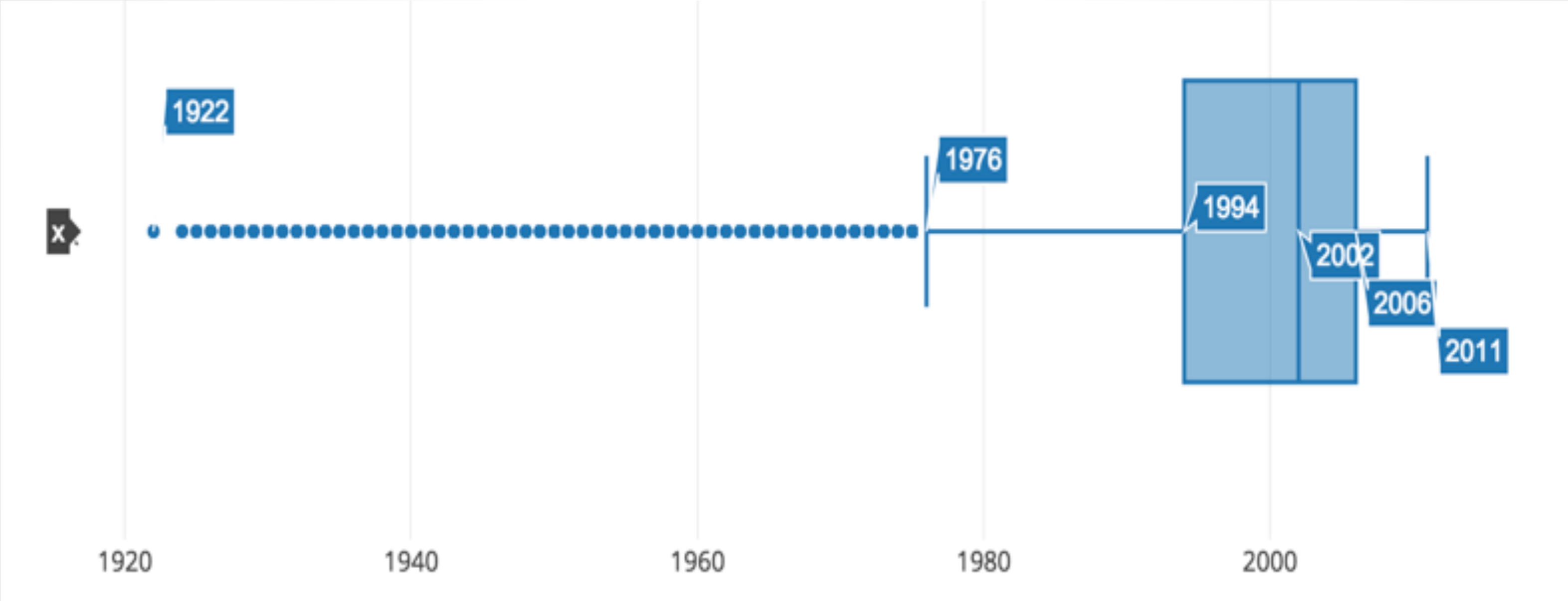
- Understand the distribution
  - How?

Distribution

https://plot.ly/~harshitshah/5/year/



Identifying Outliers: BoxPlot



https://plot.ly/~harshitshah/45/

Feature Engineering

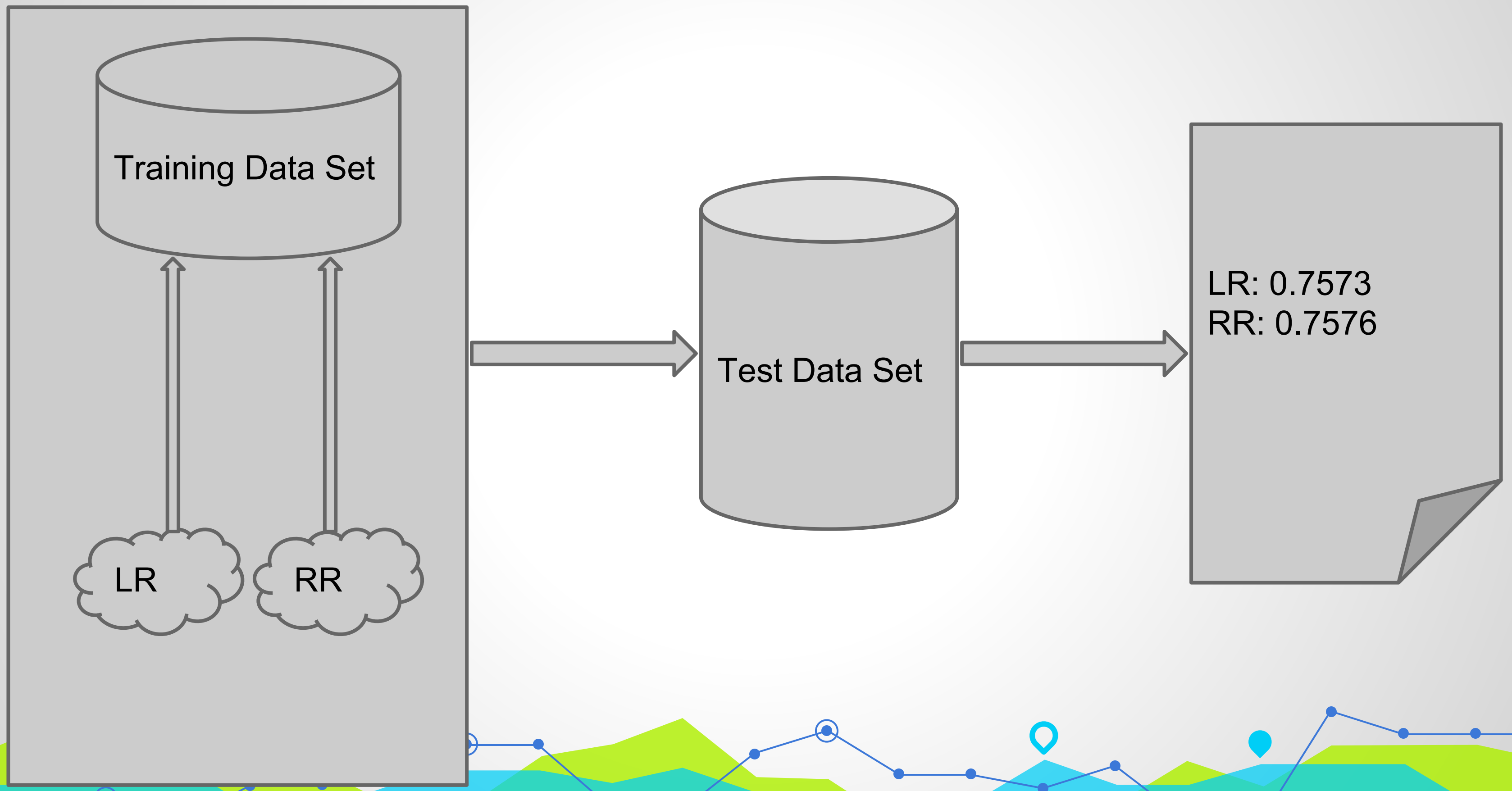
- Normalization
  - Z-score
    - (obs - mean) / stdev

[8 rows x 91 columns]

	c1	c2	c3	c4	c5	c6	c7	\
0	48.73215	18.42930	70.32679	12.94636	-10.32437	-24.83777	8.76630	
1	50.95714	31.85602	55.81851	13.41693	-6.57898	-18.54940	-3.27872	
2	48.24750	-1.89837	36.29772	2.58776	0.97170	-26.21683	5.05097	

	c1	c2	c3	c4	c5	c6	c7	\
0	0.880921	0.332293	1.748545	0.721828	-0.164946	-1.191172	0.765678	
1	1.247623	0.592599	1.337179	0.750657	-0.001111	-0.702100	-0.060917	
2	0.801045	-0.061804	0.783688	0.087218	0.329178	-1.298427	0.510712	

Model Selection & Evaluation



Case 2: Determine Income Category

- Age
- Workclass
- Education
- Race
- Occupation
- Marital Status
- Native Country
- .
- .
- .
- .
- .

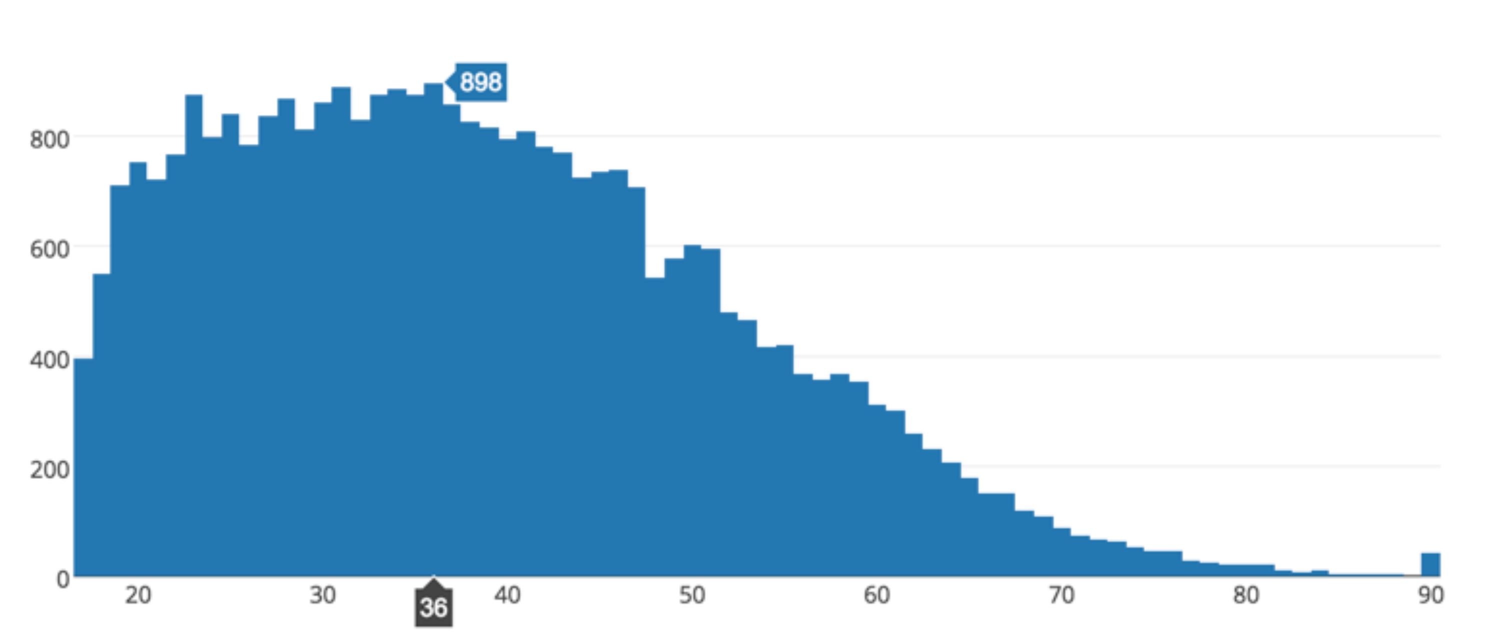
CLASSIFY

- Income
  - <= 50k
  - >50K

Data Exploration

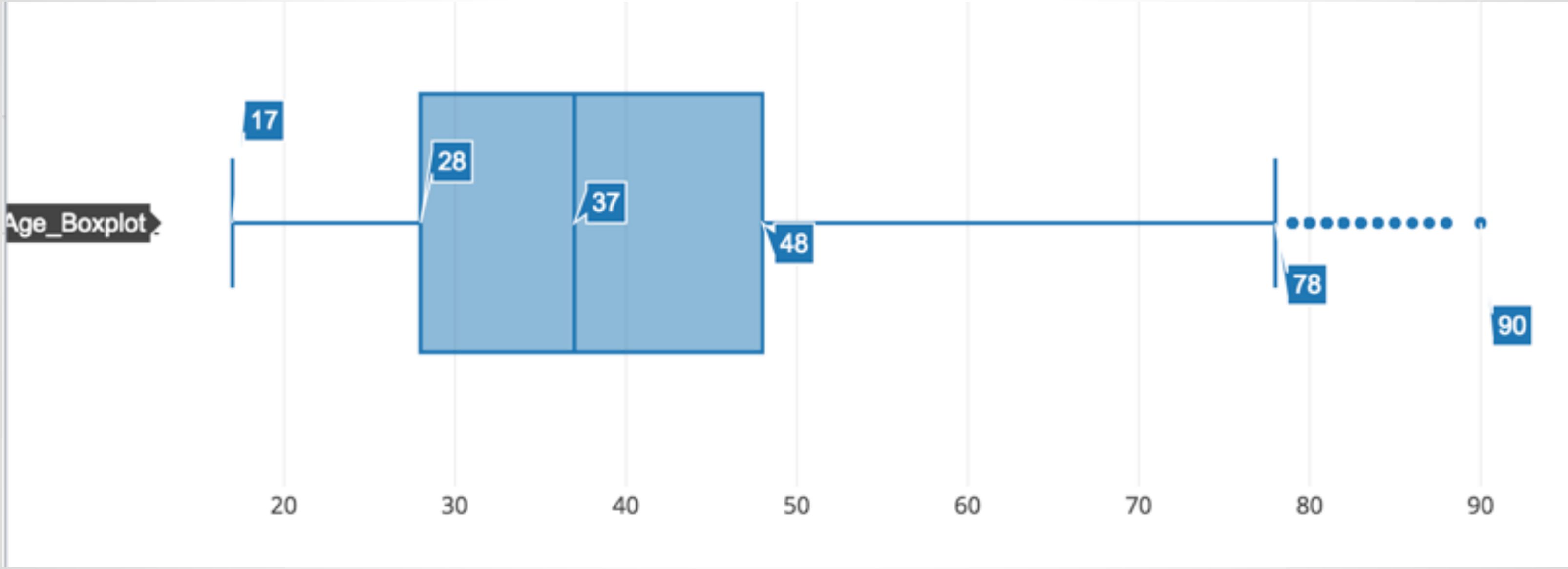
AGE

https://plot.ly/~harshitshah/5/age/



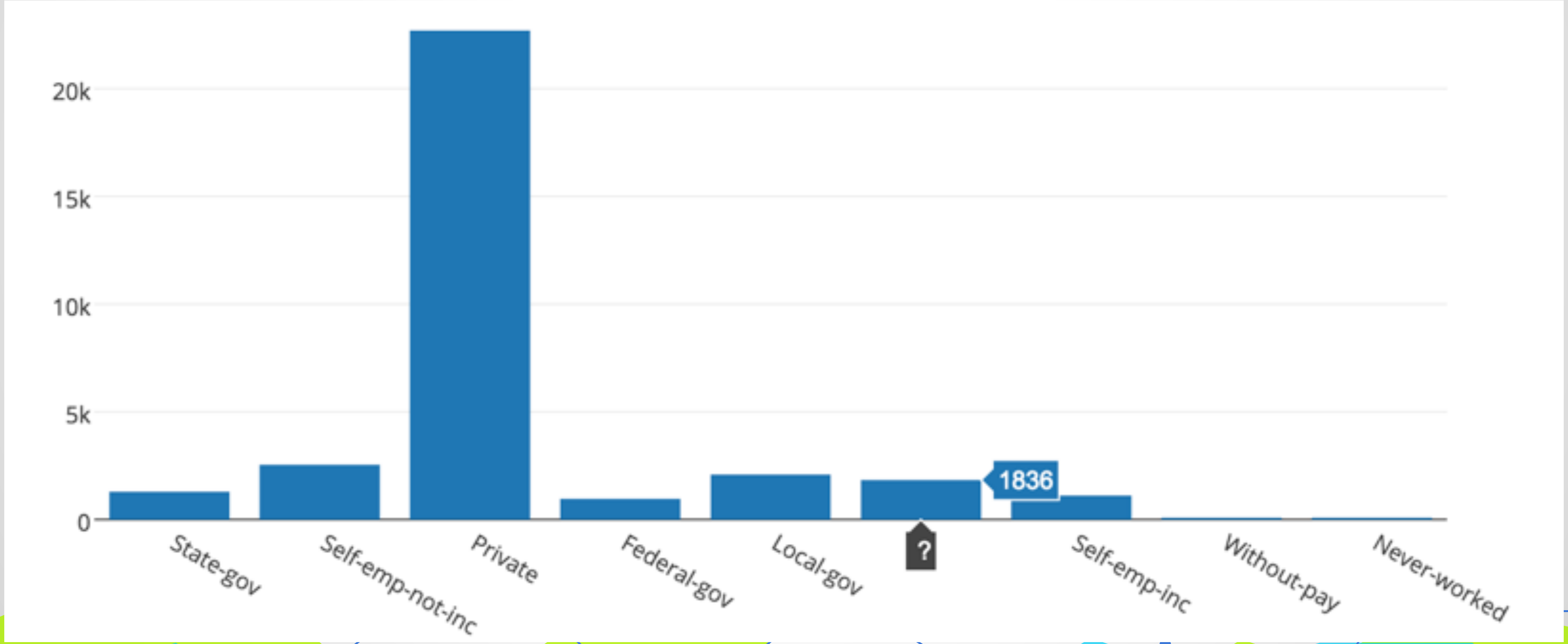
Detecting Outliers

AGE Boxplot https://plot.ly/~harshitshah/45/age-boxplot/



Finding Missing Values

Workclass https://plot.ly/~harshitshah/5/workclass/



Imputation

- Columns with Missing Values
    - Workclass
    - Marital Status
    - Occupation
    - Relationship
    - Race
    - Sex
    - Native Country

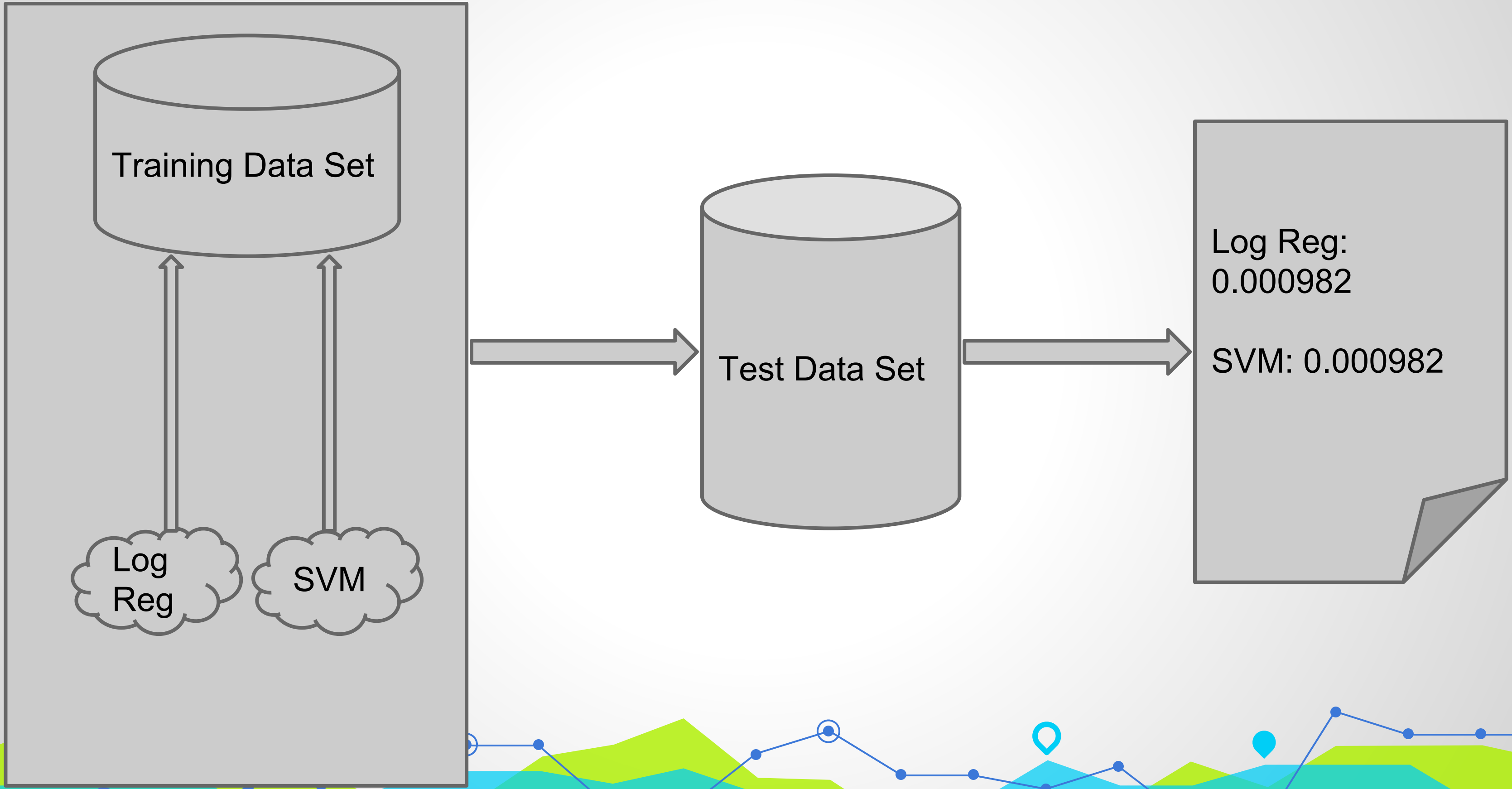
- Mode Imputation
    - Replace Missing Value with Mode of the feature

Feature Engineering

- Dummy Variables
  - Convert categorical variables to Numerical
  - Binary Variables

- Scaling
  - Scale whole data set on a scale of 0 to 1

Model Selection & Evaluation



Case-3 – Clustering

Step 1: Load the data

Step 2: Extract the features from the libsvm file

Step 3: After we have the parsed data we run the K-means model with number of iterations , number of clusters and the parsed data as the argument.

Step 4: After the clusters are formed we run the compute cost method to generate the Within Set Sum of Squared Error (WSSSE).

Code:

```
object Midterm_Q3 {

def main(args: Array[String]) {
    val conf = new SparkConf().setAppName("Spark Pi")
        /** Create the SparkContext */
        val sc = new SparkContext(conf)

        // Load and parse the data
        val data = MLUtils.loadLibSVMFile(sc, "/Users/insignia/Desktop/Big_Data_Analytics/Midterm/TV_News_Channel_Commerc
        val parsedData = data.map(s => Vectors.dense(s.features.toArray)).cache()

        val normalizer1 = new Normalizer()
        val data1 = parsedData.map(x => (normalizer1.transform(x)))
        // Cluster the data into two classes using KMeans
        val numClusters = 6
        val numIterations = 20
        val clusters = KMeans.train(data1, numClusters, numIterations)

        val numClusters1 = 8
        val numIterations1 = 20
        val clusters1 = KMeans.train(data1, numClusters1, numIterations)

        val numClusters2 = 10
        val numIterations2 = 20
        val clusters2 = KMeans.train(data1, numClusters2, numIterations)


        // Evaluate clustering by computing Within Set Sum of Squared Errors
        val WSSSE = clusters.computeCost(data1)
        val WSSSE1 = clusters1.computeCost(data1)
        val WSSSE2 = clusters2.computeCost(data1)
        println("Within Set Sum of Squared Errors 6 = " + WSSSE)
        println("Within Set Sum of Squared Errors 8 = " + WSSSE1)
        println("Within Set Sum of Squared Errors 10 = " + WSSSE2)
```

6 Clusters:

```
15/07/11 01:37:24 INFO TaskSchedulerImpl: Removed TaskSet 55.0, whose tasks have all completed, from pool
15/07/11 01:37:24 INFO DAGScheduler: Job 34 finished: sum at KMeansModel.scala:70, took 6.489288 s
Within Set Sum of Squared Errors = 6.556307464354119E10
15/07/11 01:37:24 INFO SparkContext: Invoking stop() from shutdown hook
```

10 Clusters:

```
15/07/11 02:05:05 INFO TaskSchedulerImpl: Removed TaskSet 55.0, whose tasks have all completed, from pool
15/07/11 02:05:05 INFO DAGScheduler: Job 34 finished: sum at KMeansModel.scala:70, took 10.376243 s
Within Set Sum of Squared Errors = 4.743400823740218E10
15/07/11 02:05:05 INFO SparkContext: Invoking stop() from shutdown hook
```

14 Clusters:

```
15/07/11 02:31:14 INFO DAGScheduler: ResultStage 55 (sum at KMeansModel.scala:70) finished in 12.249 s
15/07/11 02:31:14 INFO DAGScheduler: Job 34 finished: sum at KMeansModel.scala:70, took 12.259085 s
Within Set Sum of Squared Errors = 3.791888541643442E10
15/07/11 02:31:14 INFO SparkContext: Invoking stop() from shutdown hook
```

18 Clusters:

```
15/07/11 02:50:38 INFO DAGScheduler: Job 34 finished: sum at KMeansModel.scala:70, took 12.413750 s
Within Set Sum of Squared Errors = 3.192079357108762E10
15/07/11 02:50:38 INFO SparkContext: Invoking stop() from shutdown hook
```

Elbow Chart: Cluster 10 (Best K)