

PROJECT REPORT

1. OVERVIEW:

Animal shelters in USA have always been working hard to groom and make the animals adoption ready. Yet there are several “instances” where despite their valiant efforts the animal is not adopted and has to be transferred or worst put to sleep. Now that the shelters have gone IT and digitalized their data, we have an opportunity to study this data, observe the trends and most importantly can make possible prediction on the outcome for a particular animal coming into the shelter. This will help the shelter identify the weaker ones that have lesser chances of adoption and channelize their efforts on them to improve adoption numbers.

The GOAL of this project is to analyze Animal Shelter Data, observe the existing patterns in the given dataset and predict outcomes for new data by using various machine learning techniques. These insights could help shelters focus their energy on specific animals who need a little extra help finding a new home.

2. ABOUT THE DATASET:

2.1. Animal Shelter Outcomes:

The following dataset is taken from Austin Animal Shelter from October 1st, 2013 to March, 2016 which is used to predict the outcome for each animal.

The data shows that close to 30% of companion animals that come to the adoption shelters end up being unwanted and are euthanized which is the major area of concern.

Following is the snapshot of the dataset:

	A	B	C	D	E	F	G	H	I	J
1	AnimalID	Name	DateTime	OutcomeType	OutcomeSubtype	AnimalType	SexuponOutcome	AgeuponOutcome	Breed	Color
2	A671945	Hambone	2/12/2014 18:22	Return_to_owner		Dog	Neutered Male	1 year	Shetland Sheepdog Mix	Brown/White
3	A656520	Emily	10/13/2013 12:44	Euthanasia	Suffering	Cat	Spayed Female	1 year	Domestic Shorthair Mix	Cream Tabby
4	A686464	Pearce	1/31/2015 12:28	Adoption	Foster	Dog	Neutered Male	2 years	Pit Bull Mix	Blue/White
5	A683430		7/11/2014 19:09	Transfer	Partner	Cat	Intact Male	3 weeks	Domestic Shorthair Mix	Blue Cream
6	A667013		11/15/2013 12:52	Transfer	Partner	Dog	Neutered Male	2 years	Lhasa Apso/Miniature Poodle	Tan

2.1.1. SUMMARY:

- *Number of Instances* = 26729
- *Number of Attributes* = 10

2.1.2. ATTRIBUTES:

- AnimalID
- Name
- DateTime
- OutcomeType: represent the status of animals as they leave the Animal Center.

Adoption	Died	Euthanasia	Return to owner	Transfer
----------	------	------------	-----------------	----------

- OutcomeSubtype
- AnimalType
- SexuponOutcome
- AgeuponOutcome
- Breed
- Color

2.1.3. PRE-PROCESSING:

We did some pre-processing before classifying with machine learning algorithms:

- DateTime:
This attribute is split into
 - Year
 - Month
 - TimeofDay: Morning, Afternoon, Evening, Night
- SexuponOutcome:
This attribute is split into:
 - Sex: Male or Female
 - Intact: Neutered, Spayed, Intact
- AgeuponOutcome:
This attribute is converted into AgeinDays which represents age in terms of days. AgeinDays is further converted to AgeType which classifies an animal as 'baby' or 'adult' depending on age.
- Breed:
This attribute contains multiple breeds separated by '/' for some instances. The first breed is taken in this case and stored in SimpleBreed.
- Color:
This attribute contains multiple breeds separated by '/' for some instances.

- The first color is taken in this case and stored in SimpleColor.
- Attributes that will be considered for training: Year, Month, TimeofDay, OutcomeType, AnimalType, Sex, Intact, AgeType, Breed, Color
- Null Values: Null values in attributes are classified as 'Unknown'

3. MACHINE LEARNING QUESTIONS:

Following are some of the major ML questions that we wanted to answer using this project.

Question 1 →

What could be the potential outcome for a new dog coming to the shelter?

Question 2 →

What could be the potential outcome for a new cat coming to the shelter?

Here are some other interesting questions that could be predicted, but are currently out of the scope of this solution.

Question →

In which season/days are animals more likely to be adopted so that the shelter could make advance plans?

Question →

For a given un-named animal (cat or a dog), does giving the name increase the chances of adoption?

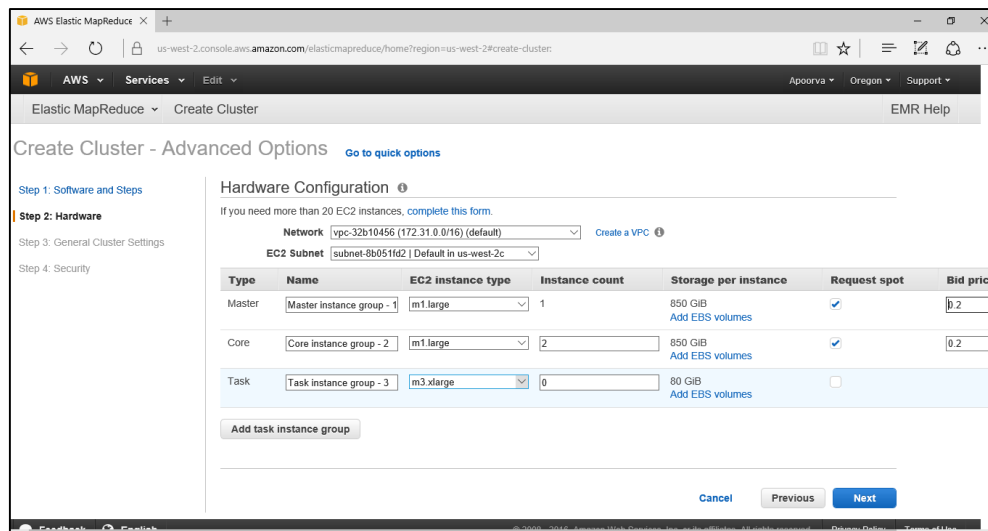
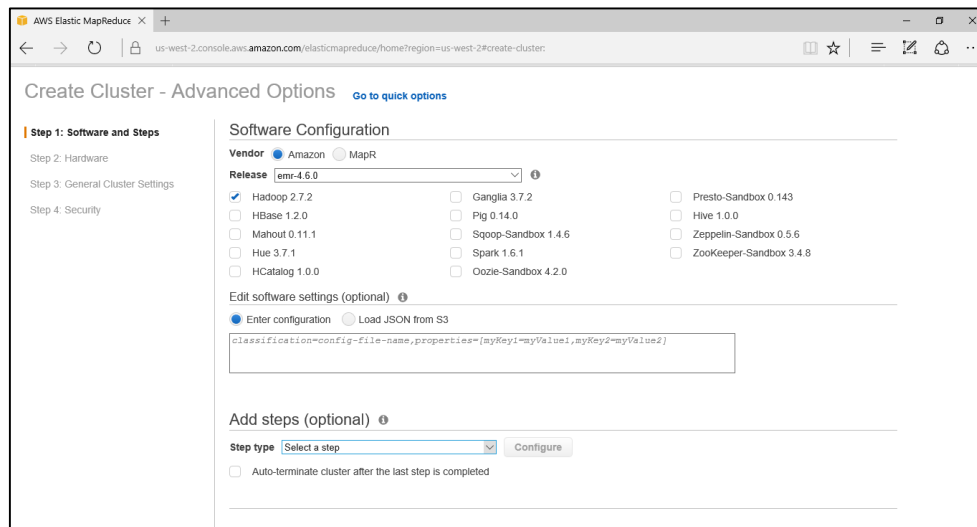
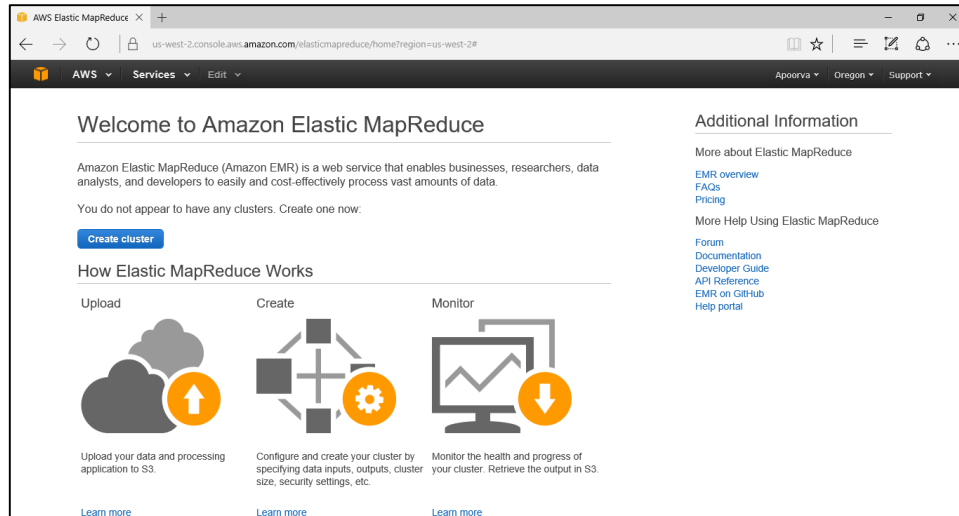
4. Using Hadoop - Map Reduce to ease the analysis

4.1. Hadoop – MapReduce

We use Hadoop MapReduce framework for aggregating the large dataset so that we can use the summarized dataset for analysis.

Enabling MapReduce using AWS:

Stepwise process for enabling and running an Amazon Elastic MapReduce job:



AWS Elastic MapReduce x +

us-west-2.console.aws.amazon.com/elasticmapreduce/home?region=us-west-2#create-cluster

AWS Services Edit Apoorva Oregon Support

Elastic MapReduce Create Cluster EMR Help

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

General Options

Cluster name

☒ Logging ⓘ

S3 folder

☒ Debugging ⓘ

☒ Termination protection ⓘ

Tags ⓘ

Key	Value (optional)
Project	Big Data
Add a key to create a tag	

Additional Options

☐ EMRFS consistent view ⓘ

AWS Elastic MapReduce x EC2 Management Console +

us-west-2.console.aws.amazon.com/elasticmapreduce/home?region=us-west-2#create-cluster

AWS Services Edit Apoorva Oregon Support

Create Cluster - Advanced Options [Go to quick options](#)

Step 1: Software and Steps
Step 2: Hardware
Step 3: General Cluster Settings
Step 4: Security

Security Options

EC2 key pair

☒ Cluster visible to all IAM users in account ⓘ

Permissions ⓘ

☒ Default ☐ Custom

Use default IAM roles. If roles are not present, they will be automatically created for you with managed policies for automatic policy updates.

EMR role

EC2 instance profile

EC2 Security Groups

An EC2 security group acts as a virtual firewall for your cluster nodes to control inbound and outbound traffic. There are two types of security groups you can configure, **EMR managed security groups** and **additional security groups**. EMR will **automatically update** the rules in the EMR managed security groups in order to launch a cluster. [Learn more](#).

Type	EMR managed security groups	Additional security groups
Master	<input type="text" value="Create ElasticMapReduce-master"/>	<input type="text" value="sg-69242e0e (default)"/>
Core & Task	<input type="text" value="Create ElasticMapReduce-slave"/>	<input type="text" value="sg-69242e0e (default)"/>

[Create a security group](#)

Encryption Options

AWS Elastic MapReduce x EC2 Management Console x EC2 Management Console +

us-west-2.console.aws.amazon.com/elasticmapreduce/home?region=us-west-2#cluster-details-j37UE84QGQ8C0

AWS Services Edit Apoorva Oregon Support

Elastic MapReduce Cluster List Cluster Details

Add step Resize Clone Terminate AWS CLI export

Cluster: MyCluster Starting

Connections: --
Master public DNS: --
Tags: Project = Big Data [View All / Edit](#)

Summary	Configuration Details	Network and Hardware	Security and Access
<p>ID: j37UE84QGQ8C0</p> <p>Creation date: 2016-05-23 15:27 (UTC-7)</p> <p>Elapsed time: --</p> <p>Auto-terminate: No</p> <p>Termination protection: On Change</p>	<p>Release label: emr-4.8.0</p> <p>Hadoop distribution: Amazon 2.7.2</p> <p>Applications: --</p> <p>Log URI: s3://aws-logs-302350776389-us-west-2/elasticmapreduce</p> <p>EMRFS consistent view: Disabled</p>	<p>Availability zone: --</p> <p>Subnet ID: subnet-b0a518c2</p> <p>Master: Provisioning 1 m1.large (Spot 0.2)</p> <p>Core: Provisioning 2 m1.large (Spot 0.2)</p> <p>Task: --</p>	<p>Key name: ApoorvaKey</p> <p>EC2 instance profile: EMR_EC2_DefaultRole</p> <p>EMR role: EMR_DefaultRole</p> <p>Visible to all users: All Change</p> <p>Security groups for ElasticMapReduce-master: More</p> <p>Master: --</p> <p>Security groups for ElasticMapReduce-slave: More</p> <p>Core & Task: --</p>

Monitoring

Hardware

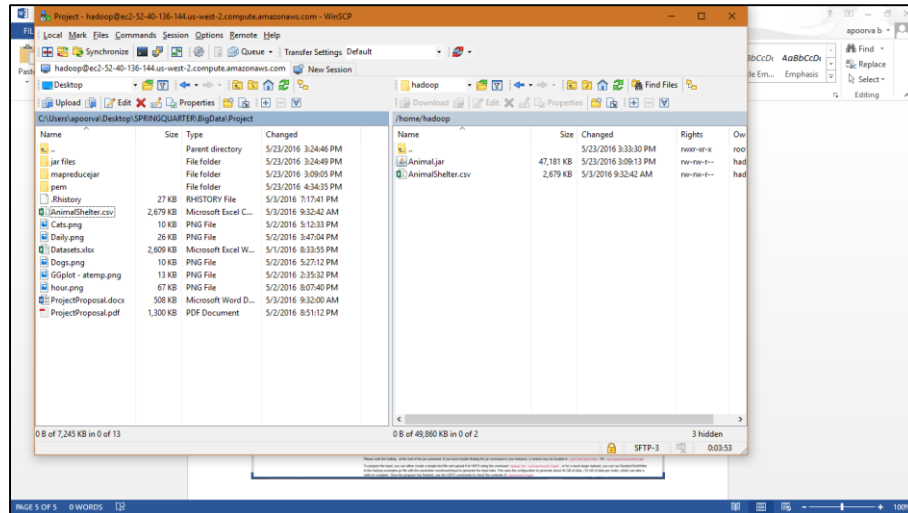
Steps

Configurations

Bootstrap Actions

Feedback English

© 2016 - 2017 Amazon Web Services, Inc. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#)



Running the following commands to run our MapReduce program:

```
[hadoop@ip-172-31-9-69 ~]$ ls -l
total 49864
-rw-rw-r-- 1 hadoop hadoop 48313087 May 23 22:09 Animal.jar
-rw-rw-r-- 1 hadoop hadoop 2742881 May 3 16:32 AnimalShelter.csv
hadoop@ip-172-31-9-69 ~$
```

Creating an input folder and copying the Animal Shelter dataset:

```
[hadoop@ip-172-31-9-69 ~]$ hadoop fs -ls /
Found 4 items
drwxr-xr-x - hadoop hadoop 0 2016-05-23 22:34 /mnt
drwxrwxrwt - hdfs hadoop 0 2016-05-23 22:34 /tmp
drwxr-xr-x - hdfs hadoop 0 2016-05-23 22:34 /user
drwxr-xr-x - hdfs hadoop 0 2016-05-23 22:34 /var
[hadoop@ip-172-31-9-69 ~]$ hadoop fs -mkdir /input
[hadoop@ip-172-31-9-69 ~]$ hadoop fs -put AnimalShelter.csv /input
[hadoop@ip-172-31-9-69 ~]$ hadoop fs -ls /input
Found 1 items
-rw-r--r-- 1 hadoop hadoop 2742881 2016-05-23 23:42 /input/AnimalShelter.csv
```

Running the jar :

```

hadoop@ip-172-31-9-69:~$ hadoop jar Animal.jar /input/output
16/05/24 00:12:10 INFO client.RMProxy: Connecting to ResourceManager at ip-172-31-9-69.us-west-2.compute.internal/172.31.9.69:8032
16/05/24 00:12:11 INFO metrics.MetricsSaver: MetricsConfigRecord disabledInCluster: false instanceEngineCycleSec: 60 disableClusterEngine: true maxMemoryMb: 3072 maxI
instanceCount: 500 lastModified: 1464042873196
16/05/24 00:12:11 INFO metrics.MetricsSaver: Created MetricsSaver j-37UE8E4GQ8C0:i-0777a647fe197c80e:RunJar:27230 period:60 /mnt/var/em/raw/i-0777a647fe197c80e_20160524_RunJar_27230_raw.bin
16/05/24 00:12:12 INFO input.FileInputFormat: Total input paths to process : 1
16/05/24 00:12:12 INFO lzo.LzoNativeCodeLoader: Loaded native lzo library
16/05/24 00:12:12 INFO lzo.LzoCodec: Successfully loaded & initialized native-lzo library [hadoop-lzo rev 426d94a07125cf9447bb0c2b336cf10b4c254375]
16/05/24 00:12:13 INFO mapreduce.JobSubmitter: number of splits=1
16/05/24 00:12:13 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1464042858673_0003
16/05/24 00:12:14 INFO impl.YarnClientImpl: Submitted application application_1464042858673_0003
16/05/24 00:12:14 INFO mapreduce.Job: The url to track the job: https://ip-172-31-9-69.us-west-2.compute.internal:20888/proxy/application_1464042858673_0003/
16/05/24 00:12:14 INFO mapreduce.Job: Running job: job_1464042858673_0003
16/05/24 00:12:26 INFO mapreduce.Job: Job job_1464042858673_0003 running in uber mode : false
16/05/24 00:12:26 INFO mapreduce.Job: map 0% reduce 0%
16/05/24 00:12:35 INFO mapreduce.Job: map 100% reduce 0%
16/05/24 00:12:47 INFO mapreduce.Job: map 100% reduce 20%
16/05/24 00:12:48 INFO mapreduce.Job: map 100% reduce 40%
16/05/24 00:12:55 INFO mapreduce.Job: map 100% reduce 60%
16/05/24 00:12:56 INFO mapreduce.Job: map 100% reduce 100%
16/05/24 00:12:57 INFO mapreduce.Job: Job job_1464042858673_0003 completed successfully
16/05/24 00:12:58 INFO mapreduce.Job: Counters: 50
File System Counters
  FILE: Number of bytes read=31214
  FILE: Number of bytes written=815541
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=2743023
  HDFS: Number of bytes written=96792
  HDFS: Number of read operations=18
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=10
Job Counters
  Killed reduce tasks=1
  Launched map tasks=1
  Launched reduce tasks=5
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=163536
  Total time spent by all reduces in occupied slots (ms)=3253632
  Total time spent by all map tasks (ms)=6814
  Total time spent by all reduce tasks (ms)=67784
  Total vcore-milliseconds taken by all map tasks=6814
  Total vcore-milliseconds taken by all reduce tasks=67784
  Total megabyte-milliseconds taken by all map tasks=5233152
  Total megabyte-milliseconds taken by all reduce tasks=104116224
Map-Reduce Framework
  Map input records=26730
  Map output records=26730
  Map output bytes=1014530
  Map output materialized bytes=31194
  Input split bytes=142
  Combine input records=26730
  Combine output records=2416
  Reduce input groups=2416
  Reduce shuffle bytes=31194
  Reduce input records=2416
  Reduce output records=2416

```

```

hadoop@ip-172-31-9-69:~$ hadoop jar Animal.jar /input/output
16/05/24 00:12:58 INFO mapreduce.Job: Counters: 50
File System Counters
  FILE: Number of bytes read=31214
  FILE: Number of bytes written=815541
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
  HDFS: Number of bytes read=2743023
  HDFS: Number of bytes written=96792
  HDFS: Number of read operations=18
  HDFS: Number of large read operations=0
  HDFS: Number of write operations=10
Job Counters
  Killed reduce tasks=1
  Launched map tasks=1
  Launched reduce tasks=5
  Data-local map tasks=1
  Total time spent by all maps in occupied slots (ms)=163536
  Total time spent by all reduces in occupied slots (ms)=3253632
  Total time spent by all map tasks (ms)=6814
  Total time spent by all reduce tasks (ms)=67784
  Total vcore-milliseconds taken by all map tasks=6814
  Total vcore-milliseconds taken by all reduce tasks=67784
  Total megabyte-milliseconds taken by all map tasks=5233152
  Total megabyte-milliseconds taken by all reduce tasks=104116224
Map-Reduce Framework
  Map input records=26730
  Map output records=26730
  Map output bytes=1014530
  Map output materialized bytes=31194
  Input split bytes=142
  Combine input records=26730
  Combine output records=2416
  Reduce input groups=2416
  Reduce shuffle bytes=31194
  Reduce input records=2416
  Reduce output records=2416
  Spilled Records=4832
  Shuffled Maps=5
  Failed Shuffles=0
  Merged Map outputs=5
  GC time elapsed (ms)=753
  CPU time spent (ms)=9240
  Physical memory (bytes) snapshot=1594040320
  Virtual memory (bytes) snapshot=10987483136
  Total committed heap usage (bytes)=1512570880
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=2742881
File Output Format Counters
  Bytes Written=96792
[hadoop@ip-172-31-9-69 ~]$

```

The output:

```

[hadoop@ip-172-31-9-69 ~]$ hadoop fs -ls /output
Found 6 items
-rw-r--r-- 1 hadoop hadoop 0 2016-05-24 00:12 /output/_SUCCESS
-rw-r--r-- 1 hadoop hadoop 20600 2016-05-24 00:12 /output/part-r-00000
-rw-r--r-- 1 hadoop hadoop 18401 2016-05-24 00:12 /output/part-r-00001
-rw-r--r-- 1 hadoop hadoop 18359 2016-05-24 00:12 /output/part-r-00002
-rw-r--r-- 1 hadoop hadoop 20566 2016-05-24 00:12 /output/part-r-00003
-rw-r--r-- 1 hadoop hadoop 18866 2016-05-24 00:12 /output/part-r-00004

```

Merging the result:

```

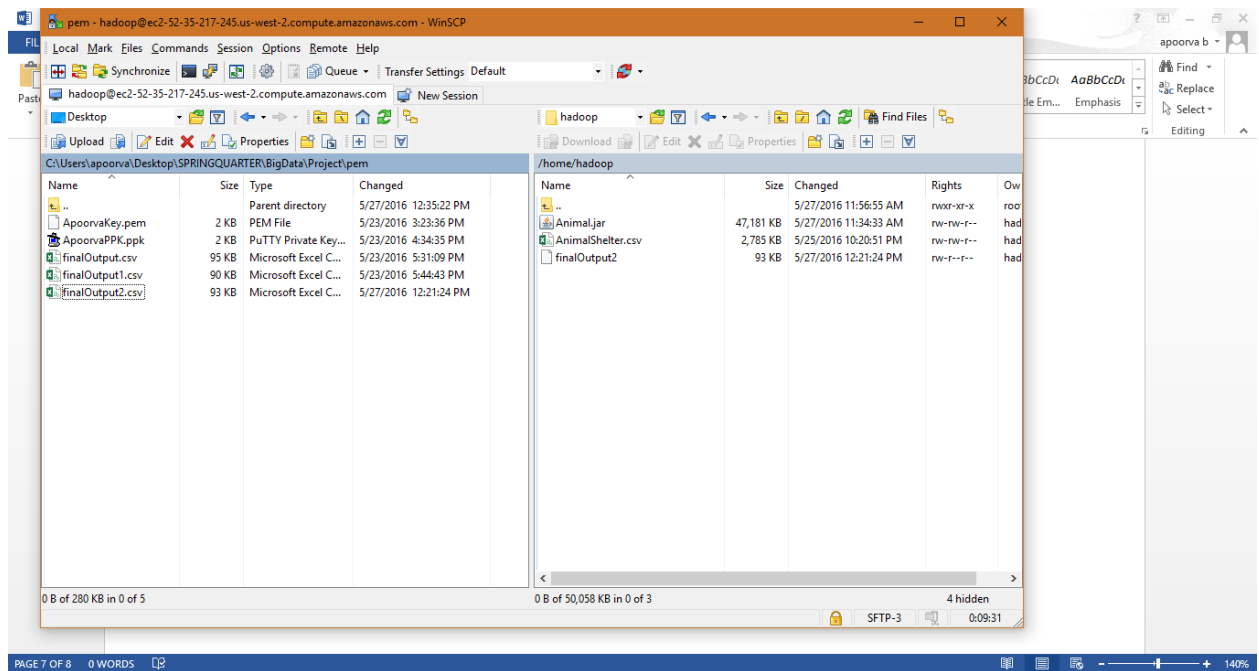
[hadoop@ip-172-31-9-69 ~]$ hadoop fs -getmerge /output ~/finalOutput

```

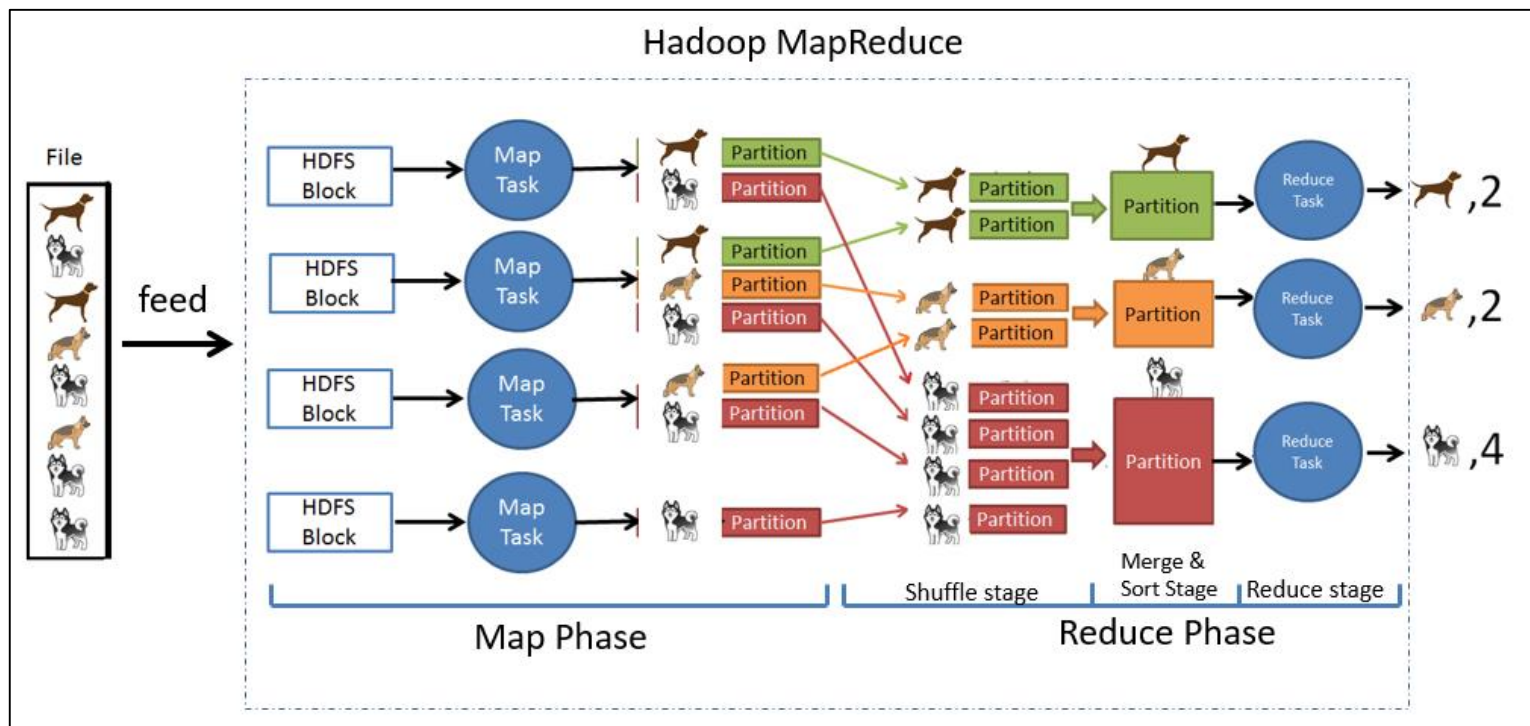


```
[hadoop@ip-172-31-9-69 ~]$ ls -l
total 49964
-rw-rw-r-- 1 hadoop hadoop 48313085 May 23 23:48 Animal.jar
-rw-rw-r-- 1 hadoop hadoop 2742881 May 3 16:32 AnimalShelter.csv
-rw-rw-r-- 1 hadoop hadoop 96792 May 24 00:17 finalOutput
```

Transferring to local system:



MapReduce follows the following workflow:-



1. Input Processor: Data from the file is processed into Key/Value pairs which is provided to a Map function.
2. Mapper: The map function outputs a transformed (writable) set of Key/Value pairs, which are subsequently processed by a Reduce function.

```
public static class AnimalMapper extends Mapper<LongWritable, Text, Text, IntWritable> {
    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();
    @Override
    public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {
        String line = value.toString();
        String[] lineArray = line.split(",");
        if (lineArray != null) {
            if (lineArray[8] != null || !lineArray[8].equals("")) {
                word.set(lineArray[8] + "," + lineArray[3] + ",");
                context.write(word, one);
            }
        }
    }
}
```

3. Shuffle and Sort: The next step is the Shuffle and Sort, where the results emitted from each mapper are sorted by key, and partitioned into one of the reducers. This is done when each map task completes to avoid an overload of traffic at the end of the final mapper's operation. This data is written to the local disk and passed into the memory of a waiting reduce task.
4. Reducer: This function is called once for each unique key emitted from the mapper. The Reducer has an iterator for all values for each key. This can be used to aggregate results, and finally returns another output in the desired Output Format.

```
public static class AnimalReducer extends Reducer<Text, IntWritable, Text, IntWritable> {
    /**
     * method that iterates all values over a particular key, which is the
     * text assigned in Mapper, and sums up the values
     */
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
        throws IOException, InterruptedException {
        int sum = 0;
        // sum all values by iterating all values over a particular key
        for (IntWritable value : values) {
            sum += value.get();
        }
        context.write(key, new IntWritable(sum));
    }
}
```

In our project, we used to find the outcome of dogs and cats based on Breed type.

	A	B	C
1	Key		Value
2	Abyssinian Mix	Adoption	2
3	Airedale Terrier Mix	Adoption	1
4	Airedale Terrier Mix	Return_to_owner	2
5	Airedale Terrier	Transfer	1
6	Akita	Adoption	2
7	Akita/Labrador Retriever	Transfer	1
8	Alaskan Husky/Australian	Transfer	1
9	Alaskan Malamute Mix	Return_to_owner	4
10	American Bulldog/Labrad	Transfer	1
11	American Eskimo	Transfer	1
12	American Pit Bull Terrier M	Euthanasia	4
13	American Pit Bull Terrier	Euthanasia	1
14	American Pit Bull Terrier/A	Return_to_owner	1
15	American Pit Bull Terrier/C	Return_to_owner	1
16	American Pit Bull Terrier/F	Adoption	1
17	American Shorthair Mix	Transfer	7

This summarized data can now be easily consumed for further analysis.

4.2.Resultant Analysis:

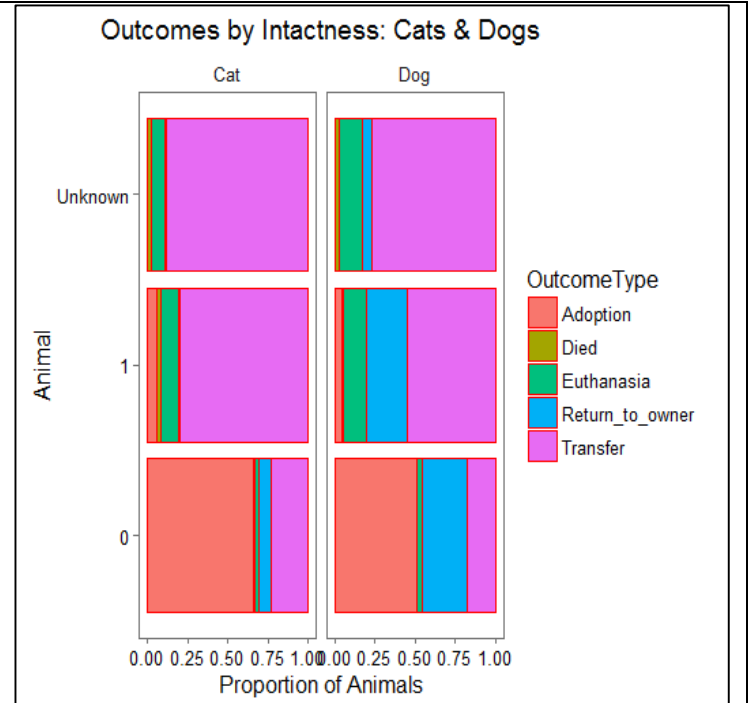
Using the resultant aggregated dataset (in a csv format) we can now fit the data into R and perform analysis.

The following points show various analysis made on this dataset:

1. Impact of Intactness on Outcome:

Plot

```
ggplot(intact, aes(x = Intact, y =
num_animals, fill = OutcomeType))
+geom_bar(stat = 'identity', position
= 'fill', colour = 'Red')
+facet_wrap(~AnimalType) +
coord_flip() + labs(y = 'Proportion of
Animals', x = 'Animal', title =
'Outcomes by Intactness: Cats &
Dogs') + theme_few()
```



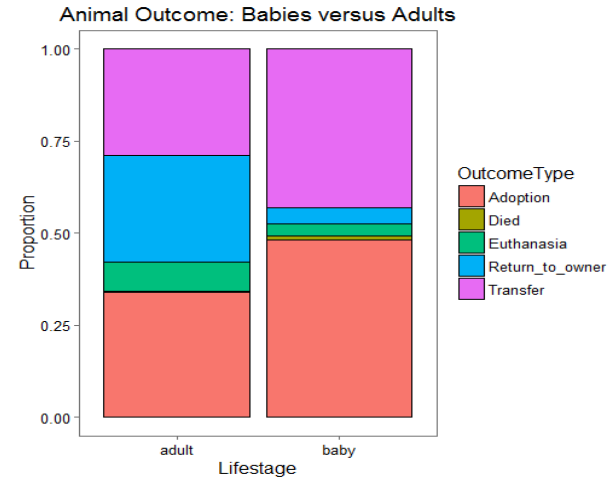
From this graph we can see that adoption rate is more for Spayed or Neutered animals in case of both Cats and Dogs. Intact animals are more likely to be transferred.

2. Baby vs Adult :

(Age<1 Year is considered as Baby)

Plot

```
ggplot(train[1:26729, ], aes(x =
Lifestage, fill = OutcomeType)) +
geom_bar(position = 'fill', colour =
'black') + labs(y = 'Proportion', title =
'Animal Outcome: Babies versus
Adults') + theme_few()
```



We see that adoptions and transfers are more for baby animals compared to an adult animal. On the other hand, euthanasia is more prominent for adult animal compared to baby animal.

3. Analyzing the impact of age and Animal Type on the Outcome.

CATS

#Filtering cat data from the dataset

```
> cat_data <- filter(Animal_data, AnimalType == "Cat")
```

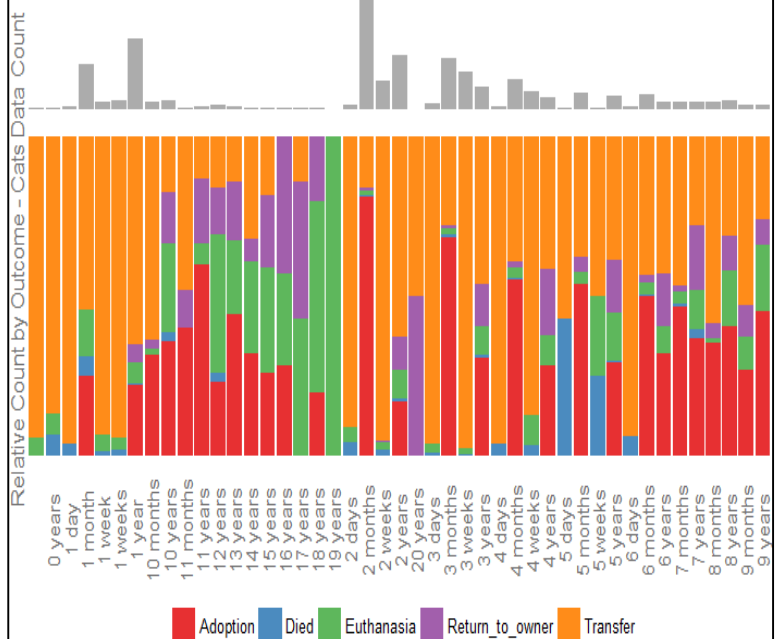
#How does the OutcomeType affect the age upon Outcome

```
> cat_data %>% count(AgeuponOutcome) %>%
+ ggplot(aes(x = AgeuponOutcome, y = n))
+ geom_bar(stat = "identity", alpha = 0.5)
+ theme_void()
+ ylab("Count")
+ ggtitle("Effect of OutcomeType by Age for Cats Data")
+ theme(axis.title.y = element_text(angle = 90, color = "#737373")) -> g1
```

Relative Count by Outcome – Cats Data

```
> ggplot(cat_data, aes(x = AgeuponOutcome, fill = OutcomeType))
+ geom_bar(stat = "count", position = "fill", alpha = 0.9)
+ scale_fill_brewer(palette = "Set1")
+ theme_void()
+ ylab("Relative Count by Outcome") +
+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 1),
+ color = "#737373"), legend.position = "bottom", axis.title.y = element_text(
+ angle = 90, color = "#737373")) -> g2
# To bring up the above 2 Graphs on the same page
> grid.arrange(g1, g2, ncol = 1, heights = c(1,3))
```

Effect of OutcomeType by Age for Cats Data

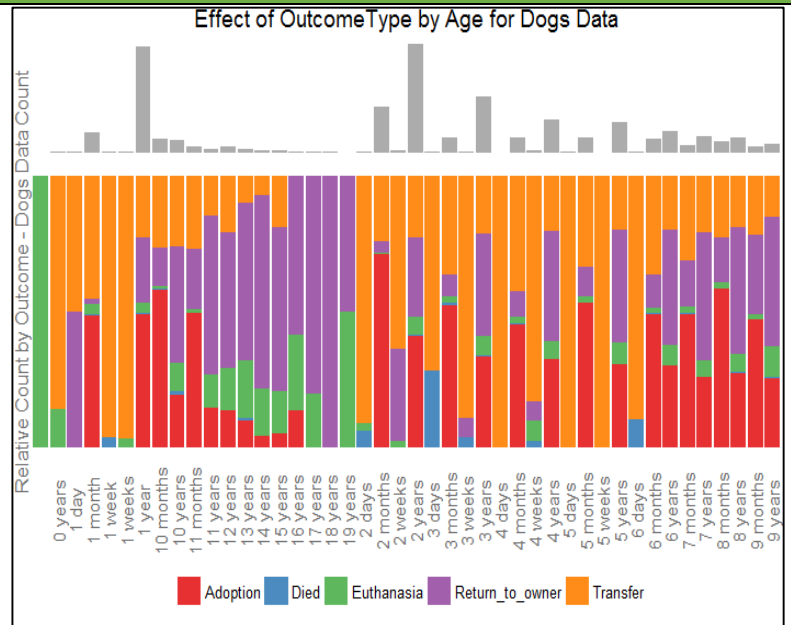


OBSERVATION- Age does not seem to have much impact on adoption rates (unlike for dogs – see below).

DOGS

Filtering cat data from the dataset

```
> dog_data <- filter(Animal_data, AnimalType == "Dog")
#How does the OutcomeType affect the age upon Outcome
> dog_data %>% count(AgeuponOutcome) %>%
+ ggplot(aes(x = AgeuponOutcome, y = n))
+ geom_bar(stat = "identity", alpha = 0.5)
+ theme_void()
+ ylab("Count")
+ ggtitle("Effect of OutcomeType by Age for dogs data")
+ theme(axis.title.y = element_text(angle = 90, color = "#737373")) -> g3
# Relative Count by Outcome – Dogs Data
> ggplot(dog_data, aes(x = AgeuponOutcome, fill = OutcomeType))
+ geom_bar(stat = "count", position = "fill", alpha = 0.9)
+ scale_fill_brewer(palette = "Set1")
+ theme_void()
+ ylab("Relative Count by Outcome") +
+ theme(axis.text.x = element_text(angle = 90, vjust = 0.5, size = 11, color = "#737373"),
+       legend.position = "bottom",
+       axis.title.y = element_text(angle = 90, color = "#737373"))
-> g4
# To bring up the above 2 Graphs on the same page
> grid.arrange(g3, g4, ncol = 1, heights = c(1,3))
```

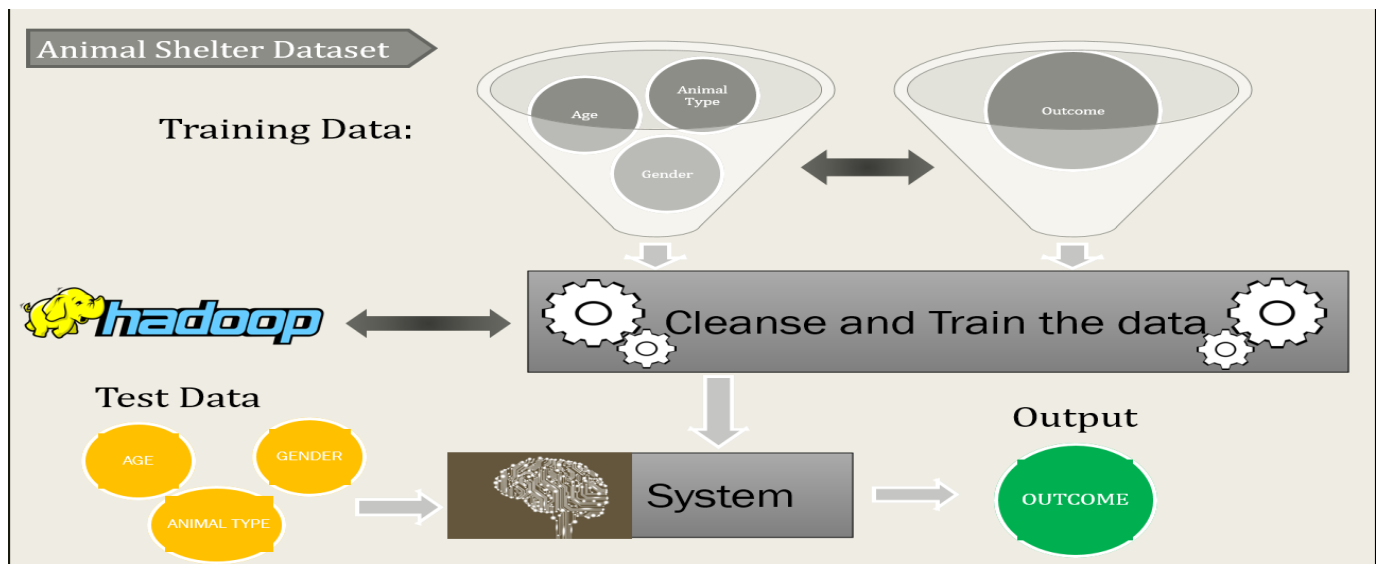


OBSERVATION-

1. The adoption rate is very high for young dogs and puppies. As the age increases the adoption rate falls.
2. Older dogs have higher chances of getting returned to owner.

4.3.SOLUTION ARCHITECTURE & ML TECHNIQUES:

The following diagram shows the solution architecture for predicting Animal Shelter Outcome for the given Dataset:

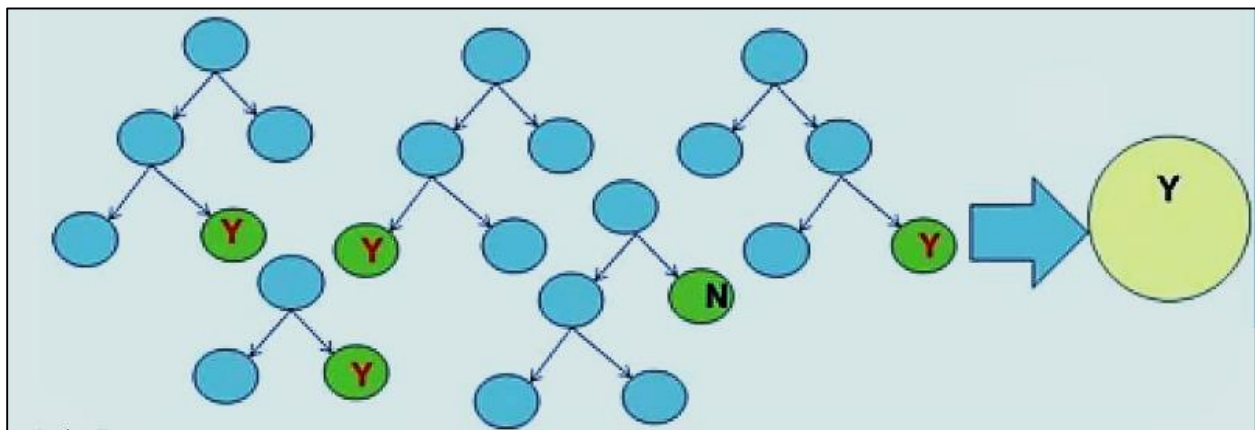


We shall be taking the animal type, age, gender, breed and other suitable attributes as input data which will give us the outcome as output. We will be training this data using *Random Forest algorithm* to predict the outcome. We will use Hadoop for aggregating Raw data for analysis.

5. MACHINE Learning Techniques and Predictions:

- Using Random Forest

A random forest is an ensemble of decision trees which will output a prediction value, in this case "outcome type". Each decision tree is constructed by using a random subset of the training data.



We then consolidate the outcome of each decision tree and choose the most popular outcome

After training the data, we can then pass test data through it, in order to output a prediction.

```
# Build the model
```

```
rf_mod <- randomForest(OutcomeType ~  
AnimalType+AgeinDays+Intact+HasName+hours+weekday+TimeofDay+SimpleColor  
+IsMix+Sex+month, data = train, ntree = 600, importance = TRUE)
```

```
# Predict using the test set
```

```
prediction <- predict(rf_mod, test, type = 'vote')
```

This code outputs:

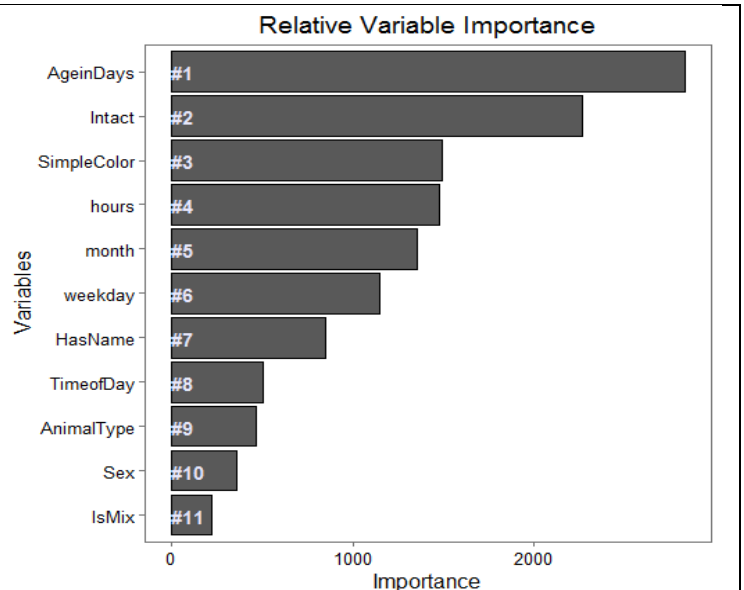
ID	Adoption	Died	Euthanasi	Return_to	Transfer
A675805	90%	0%	0%	1%	9%
A666170	97%	0%	0%	0%	3%
A669394	57%	0%	1%	18%	24%
A648948	26%	0%	2%	69%	4%
A683385	100%	0%	0%	0%	0%
A719068	2%	0%	1%	2%	95%
A684455	1%	0%	3%	73%	23%

We also calculated the **relative variable importance** (which variable has played a major role in predicting the outcome):

```
# Get importance
```

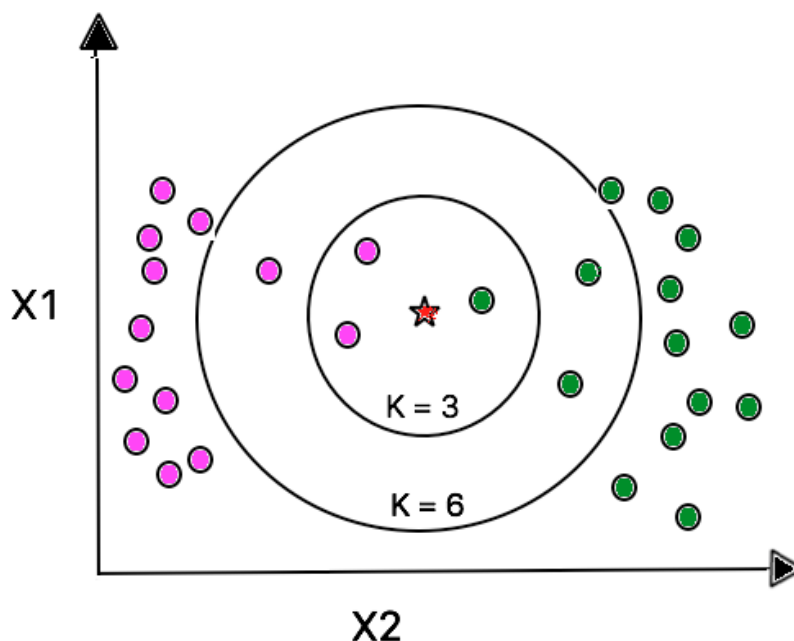
```
importance <- importance(rf_mod)
```

```
varImportance <- data.frame(Variables  
= row.names(importance), Importance  
= round(importance[,  
'MeanDecreaseGini'],2))
```



- Using K- Nearest Neighbor

Machine learning algorithms provide methods of classifying objects into one of several groups based on the values of several explanatory variables. Nearest neighbor methods are easily implemented and easy to understand. There is no model associated to them, so errors have to be estimated computationally, but it provides one simple solution to classifying a new object based on known results in a reference set. kNN is a generalization of “if it walks like a chicken, looks like a chicken, and talks like a chicken, it is probably a duck.” That is, objects that are close together with respect to the explanatory variables are likely to have the same classification.



In the simplest setting, like the example we will do here, objects can fall into one of two classes, (A) or (B) . We have a set of (n) measurements, (v_1, \dots, v_n) , of any object in question, and assume these are all numeric variables. We use a distance measure, namely Euclidean distance, to manifest when two objects are close with respect to these variables. So, given objects in the domain (s) , (r) , which have (v_i) measurements (x_1, \dots, x_n) , (y_1, \dots, y_n) , respectively, define $[dist(s, r) = \sqrt{\sum_i (x_i - y_i)^2}]$. Depending on characteristics of the variables, other distance measures may be more appropriate, but we'll stick with Euclidean distance.

The kNN algorithm begins with a training set of objects for which we know not only the values of the explanatory variables but also the classifications (A) and (B) . To predict the classification of a new object (q) , the $(k=1)$ version of kNN would proceed by finding the element of the training set with the minimum distance from (q) , suppose it is (p_1) , and predict the classification of (q) to be the same as (p_1) . If (p_1) is

rather isolated and there are lots of points in the other class almost as close to (q) this could be a misclassification. So, we generally pick some larger odd number (m) for (k) ; find (p_1, \dots, p_m) closest to (q) and vote on what the classification of (q) should be. A rule of thumb in machine learning is to pick (k) near the square root of the size of the training set. In practice this does a good job of telling signal from noise.

This outputs:

ID	Adoption	Died	Euthanasies	Return_to	Transfer
A675805	92%	0%	0%	1%	7%
A666170	98%	0%	1%	0%	1%
A669394	64%	0%	0%	15%	21%
A648948	24%	0%	1%	70%	5%
A683385	8%	0%	0%	5%	87%

6. Analysis:

We have found out that KNN is having good accuracy on the training dataset. We understood that it is because we have all the attributes as categorical and we have one attribute with 382 levels and one with 57 levels after preprocessing. In this scenario, as we are testing on large dataset containing 26729 rows and we found that KNN performs better with our dataset. Random Forest is having comparatively lesser performance because of more levels for the attributes.

7. Conclusion:

Adoption is the most likely outcome for the animals in the given dataset and Euthanasia is the least likely. Hence, we can see that the predictions are analogous to the real world scenario where Euthanasia is uncommon.

8. References:

1. <https://cran.r-project.org/web/packages/gridExtra/vignettes/arrangeGrob.html>
2. <https://cran.r-project.org/web/packages/gridExtra/gridExtra.pdf>
3. <http://www.sthda.com/english/wiki/ggplot2-easy-way-to-mix-multiple-graphs-on-the-same-page-r-software-and-data-visualization>
4. <https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>
5. <https://www.kaggle.com/c/shelter-animal-outcomes/data>