A PROJECT REPORT
ON

PREDICTING THE OUTCOME OF AN E-MAIL CLASSIFICATION

Submitted to
Osmania University in
partial fulfilment of the
requirements for the award of
Master of Science in Applied Statistics

DEPARTMENT OF STATISTICS
UNIVERSITY COLLEGE OF SCIENCE
OSMANIA UNIVERSITY
HYDERABAD – INDIA

By

KASHAPOGU Y SHRUTHI BHANDHAVI Roll No.: **1007 17 508 021**
MANGILIPALLY SRIVIDYA Roll No.: **1007 17 508 023**
VENKATESWARLU HIMABINDHU Roll No.: **1007 17 508 026**
PAINDLA PRASANNA Roll No.: **1007 17 508 024**
KATLA RAMYA Roll No.: **1007 17 508 018**
PENDRAM SUSHMA Roll No.: **1007 17 508 030**

Under the Supervision of
T. SANDHYA
DECEMBER 2018

A PROJECT REPORT
ON

PREDICTING THE OUTCOME OF AN E-MAIL CLASSIFICATION

Submitted to
Osmania University in
partial fulfilment of the
requirements for the award of
Master of Science in Applied Statistics

DEPARTMENT OF STATISTICS
UNIVERSITY COLLEGE OF SCIENCE
OSMANIA UNIVERSITY
HYDERABAD – INDIA

By

KASHAPOGU Y SHRUTHI BHANDHAVI Roll No.: **1007 17 508 021**
MANGILIPALLY SRIVIDYA Roll No.: **1007 17 508 023**
VENKATESWARLU HIMABINDHU Roll No.: **1007 17 508 026**
PAINDLA PRASANNA Roll No.: **1007 17 508 024**
KATLA RAMYA Roll No.: **1007 17 508 018**
PENDRAM SUSHMA Roll No.: **1007 17 508 030**

Under the Supervision of
T. SANDHYA

DECEMBER 2018

DECLARATION

The research presented in this project has been carried out in the **Department of Statistics, Osmania University, Hyderabad.** The work is original has not been submitted so far, in part or full, for any other degree of diploma of any university.

KASHAPOGU Y SHRUTHI BHANDHAVI
MANGILIPALLY SRIVIDYA
VENKATESWARLU HIMABINDHU
PAINDLA PRASANNA
KATLA RAMYA
PENDRAM SUSHMA

Department of Statistics Osmania
University
Hyderabad – 500 007, T. S.
INDIA

CERTIFICATE

This is to certify that

KASHAPOGU Y SHRUTHI BHANDHAVI	Roll No.: **1007 17 508 021**
MANGILIPALLY SRIVIDYA	Roll No.: **1007 17 508 023**
VENKATESWARLU HIMABINDHU	Roll No.: **1007 17 508 026**
PAINDLA PRASANNA	Roll No.: **1007 17 508 024**
KATLA RAMYA	Roll No.: **1007 17 508 018**
PENDRAM SUSHMA	Roll No.: **1007 17 508 030**

have submitted the project titled **"PREDICTING THE OUTCOME OF AN E-MAIL CLASSIFICATION"** in partial fulfilment for the degree of Master of Science in Applied Statistics.

Head

Department of Statistics

Internal Examiner External Examiner

ACKNOWLEDGEMENTS

We deem it a great pleasure to express our deep sense of gratitude and indebtedness to our research supervisor **T. SANDHYA,** Statistics Department, University College of Science, Osmania University for her valuable guidance, and enlightening discussions throughout the progress of our project work.

We also express our sincere and heartfelt thanks to **Prof.C.JAYALAKSHMI,** and all staff members of the Department of Statistics, Osmania University for providing the necessary support and facilities in the department for completion of this work successfully.

It is indeed with great pleasure we record our thanks to **Dr.G. JAYASREE**, Chairperson, Board of Studies, Osmania University for having provided with all the facilities to carry out our work.

We thank **Dr.N.Ch.BHATRACHARYULU, Dr.K.VANI, Dr.S.A.JYOTHI RANI, Dr.G.SIRISHA, Mrs.J.L.PADMA SHREE** for their encouragement and constant help during research.

We would like to express our deepest gratitude to **Dr.M. VENUGOPALA RAO, BALA KARTHEEK** for their advice, guidance and involvement at various stages of this work. We would also like to thank them for their understanding and constant encouragement throughout this project.

We thank all Non-Teaching members of the **Department of Statistics**, who helped us during our thesis work.

We are thankful to the **Osmania University** for permitting us to carry out this work.

CONTENTS

CHAPTER 1
INTRODUCTION AND SCOPE OF THE PROBLEM

INTRODUCTION AND SCOPE OF THE PROBLEM

1.1 Scope of the Problem

This problem is related to a project to create a predictive model, predicting the outcome of an E-Mail Classification. This data is extracted from **HacKart**, an **E Commerce Portal** for the **hackers**.

VARIABLE	DEFINITION
Email_Type	Type of E-Mail (1 / 2)
Subject_Hotness_Score	Hotness Score of Subject given by Email Marketing Tools (On the scale of 0 to 5, higher is better)
Email_Source_Type	Source of E-Mail (1 / 2)
Customer_Location	Location of Customer (A/ B/ C/ D/ E/ F/ G)
Email_Campaign_Type	Email Campaign Type (1/ 2/ 3)

Total_Past_Communications	Total past communications in three months
Time_Email_sent_Category	Email Sent Time Category (1/ 2/ 3)

Objective

- What are parameters to evaluate the **E-Mail Classification**
- An algorithm, by which we can classify whether the customers opened the EMail or not
- What are the key attributes that are efficient for the **Hackart**

1.2 Data Description

ATTRIBUTES

Word _Count	Total words in the E-Mail
Total_Links	Total available links in the E-Mail
Total_Images	Total images in the E-Mail
Email_Status	Dependent Variable (0 : Email not opened, 1 : Email opened)

1.3 Review of the Chapters

Chapter 2 gives the brief introduction about machine learning techniques like need of ML today, types of ML Algorithms and various models in each algorithm and what technique to use when and how to validate, tune the ML algorithms and how to measure the performance of the ML model

Chapter 3 describes the various results obtained for the problem. This section contains all the outputs generated through the ML algorithms applied on the data as well as validation and performance matrices.

Chapter 4 describes the summary and conclusions followed by **Bibliography**.

APPENDIX

It describes the **R Code** and **Data Set** used in modeling.

CHAPTER 2

REVIEW OF MACHINE LEARNING PROCESS

REVIEW OF MACHINE LEARNING PROCESS

2.0 Need of Machine Learning

In this age of modern technology, there is one resource that we have in abundance : a large amount of structured and unstructured data. In the second half of the twentieth century, machine learning evolved as a subfield of artificial intelligence that involved the development of self-learning algorithms to gain knowledge from that data in order to make predictions. Instead of requiring humans to manually derive rules and build models from analysing large amounts of data, machine learning offers a more efficient alternative for capturing the knowledge in data to gradually improve the performance of predictive models, and make data-driven decisions. Not only is machine learning becoming increasingly important in computer science research but it also plays an ever greater role in our everyday life.

2.1 Machine Learning Process

The **CRISP-DM** (**Cross-Industry Standard Process for Data Mining**) **Process** was designed specifically for the **Data Mining**. However, it is flexible and thorough enough that it can be applied to any analytical project whether it is Predictive analytics, Data science, or Machine learning. The Process has the following six phases

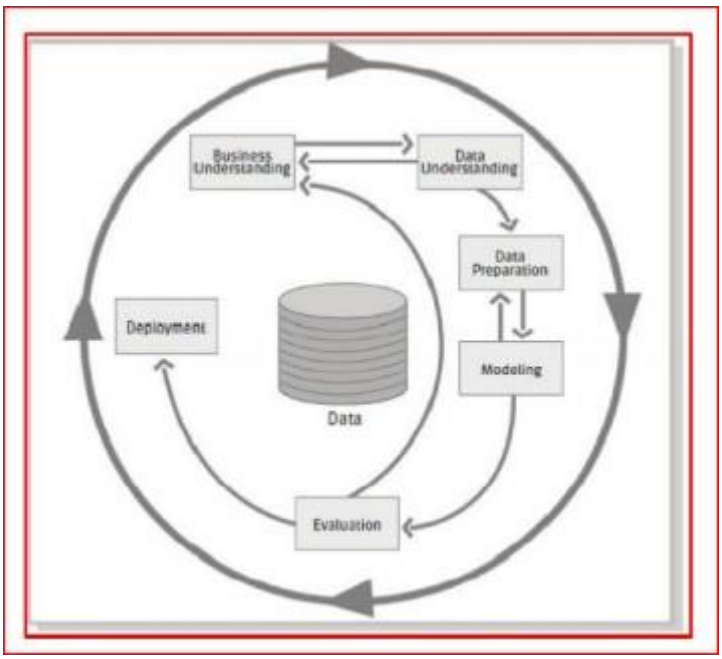


Fig. 2.1 CRISP-DM (Cross-Industry Standard Process for Data Mining) Process

- Business Understanding

- Data Understanding
- Data preparation
- Modelling
- Evaluation
- Deployment

And, each phase has different steps covering important tasks which are mentioned below:

2.1.1) Business Understanding

It is very important step of the process in achieving the success. The purpose of this step is to identify the requirements of the business so that you can translate them into analytical objectives. It has the following tasks:

1) Identify the Business objective

2) Assess the situation

3) Determine the Analytical goals

4) Produce a project plan

2.1.2) Data Understanding

After enduring the all-important pain of the first step, you can now get your hands on the data. The task in this process consist the following

1) Collect the data

2) Describe the data

3) Explore the data

4) Verify the data Quality

2.1.3) Data Preparation

This step is relatively self-explanatory and in this step the goal is to get the data ready to input in the algorithms. This includes merging, feature engineering, and transformations. If imputation for missing values / outliers is needed then, it happens in this step. The key five tasks under this step are as follows :

1) Select the data

2) Clean the data

3) Construct the data

4) Integrate the data

5) Format the data

2.1.4) Modelling

Oddly, this process step includes the consideration that you already thought of and prepared for. In this, one will need at least a modicum of an idea about how they will be modelling. Remember, that this is flexible, iterative process and some strict linear flow chart such as an aircrew checklist.

Below are the tasks in this step : 1)

Select a modelling technique

2) Generate a test design

3) Build a model

4) Assess a Model

Both cross validation of the model (**Using Train/Test or K Fold Validation**) and model assessment which involves comparing the models with the chosen criterion (**RMSE, Accuracy, ROC**) will be performed under this phase.

2.1.5) Evaluation

In the evaluation process, the main goal is to confirm that the work that has been done and the model selected at this point meets the business objective. Ask yourself and others, have we achieved the definition of success?

And, here are the tasks in this step :

1) Evaluate the results

2) Review the process

3) Determine the next steps

2.1.6) Deployment

If everything is done according to the plan up to this point, it might come down to flipping a switch and your model goes live. Here are the tasks in this step :

1) Deploying the plan

2) Monitoring and maintenance of the plan

3) Producing the final report

2.2 Types of Machine Learning

Broadly, the Machine Learning Algorithms are classified into 3 types.

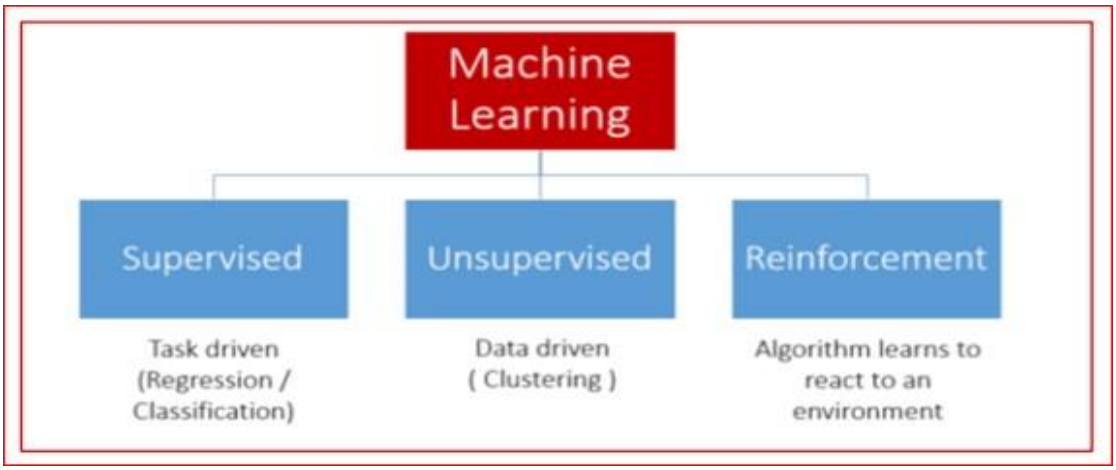


Fig. 2.2 Types of Machine Learning

2.2.1) Supervised Learning

This algorithm consists of a target / outcome / dependent variable which is to be predicted from a given set of predictors / independent variables. Using these set of variables, we generate a function that maps inputs to desired output. The training process continues until the model achieves a desired level of accuracy on the training data.

The process of Supervised Learning Model is illustrated in the below picture :

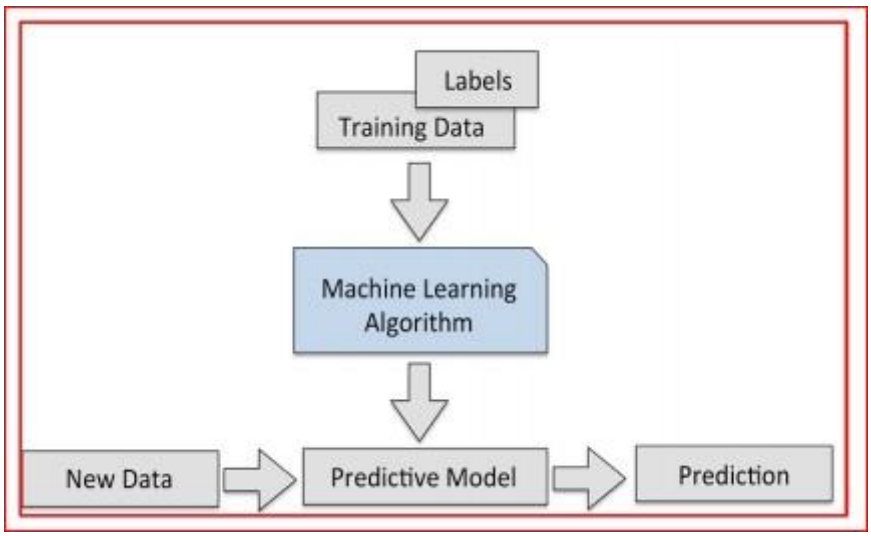


Fig. 2.2.1 Supervised Learning

Examples of Supervised Learning : Regression, Decision Tree, Random Forest, KNN, Logistic Regression...etc

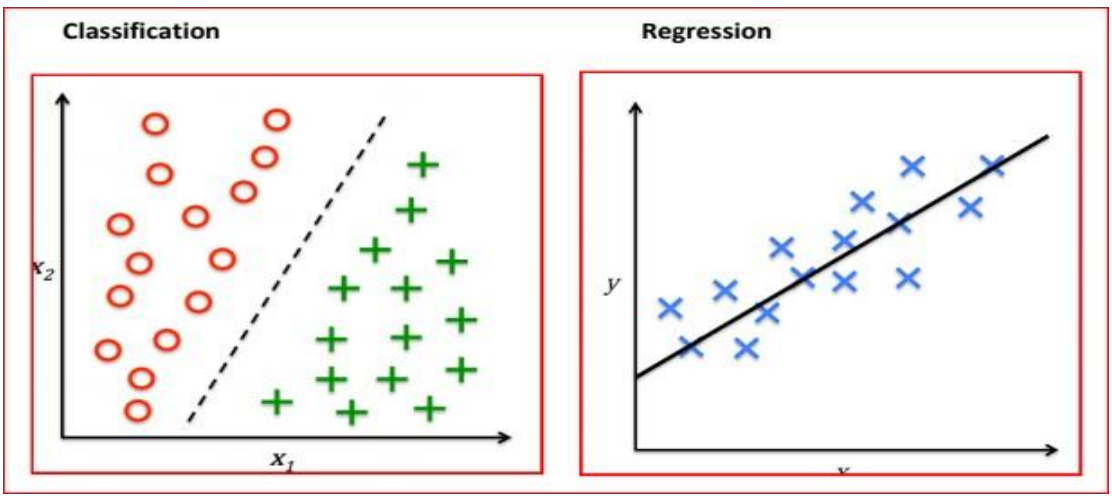


Fig. 2.2.1 Classification and Regression

2.2.2) Unsupervised Learning

In this algorithm, we will not have any target or outcome variable to predict / estimate. It is used for clustering population into different groups, which is widely used for segmenting customers in different groups for specific intervention. (**More of Exploratory Analysis**)

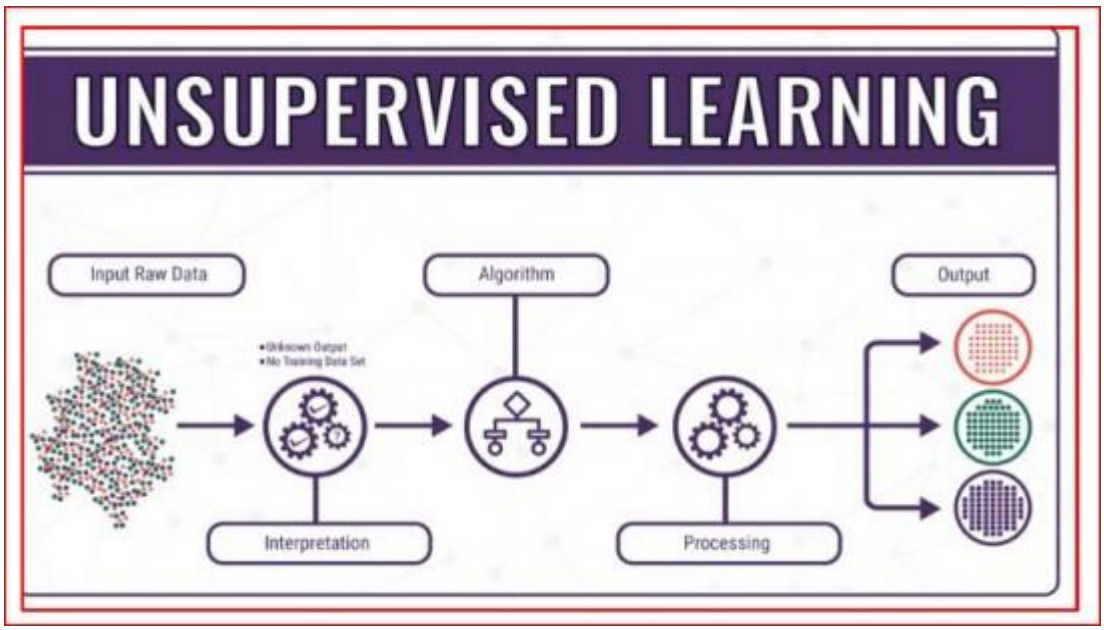


Fig. 2.2.2 Unsupervised Learning

Examples of Unsupervised Learning : Data Reduction Techniques, Cluster Analysis, Market Basket Analysis...etc

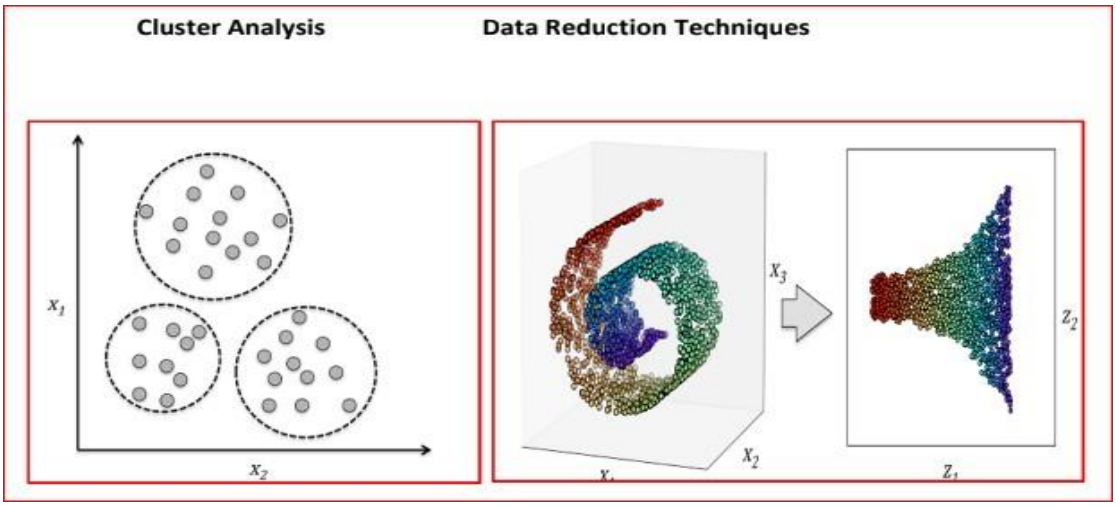


Fig. 2.2.2 Cluster Analysis and Data Reduction Techniques

2.2.3) Reinforcement Learning

Using this algorithm, the machine is trained to make specific decisions.

It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. The process of reinforcement learning is illustrated in the below picture :

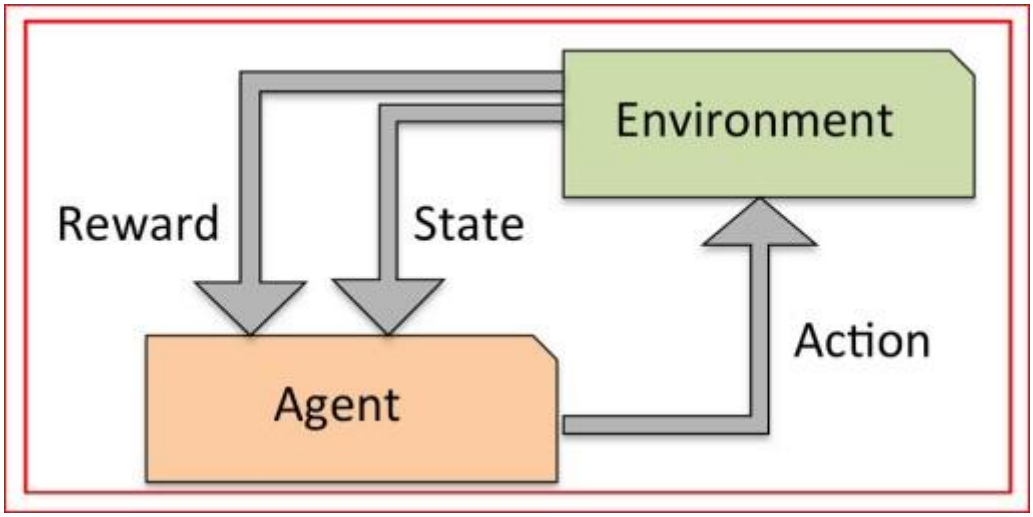


Fig. 2.2.3 Reinforcement Learning

Examples of Reinforcement Learning : Markov Decision Process, Self-Driving Cars...etc

2.3 Choosing the Algorithm

Choosing the right algorithm will depend on the type of the problem we are solving and also depends on the scale of the dependent variable. In case of continuous target variable, we will use regression algorithms and in case of categorical target, we will use classification algorithms and for the model which doesn't have target variable, we will use either Cluster Analysis / Data Reduction Techniques.

Below picture describes the process of choosing the right algorithm :

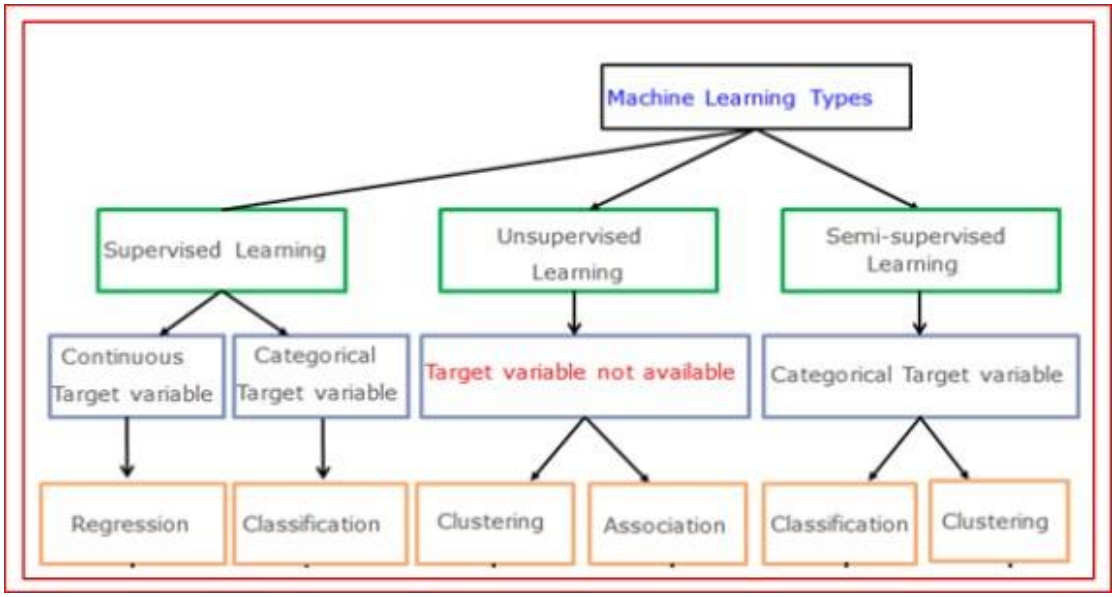


Fig. 2.3 Choosing the Algorithm

2.3.1) Types of Regression Algorithms

There are many Regression Algorithms in Machine Learning, which will be used in different regression applications. Some of the main regression algorithms are as follows :

a) Simple Linear Regression: - In Simple linear regression, we predict scores on one variable from the data of second variable. The variable we are forecasting is called the **Criterion Variable** and referred to as **Y.** The variable we are basing our predictions on is called the **Predictor Variable** and denoted as **X.**

b) Multiple Linear Regression: - Multiple linear regression is one of the algorithms of regression technique, and is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one dependent variable with two or more independent variables. The independent variables can be either continuous or categorical.

c) Polynomial Regression: - Polynomial regression is another form of regression in which the maximum power of the independent variable is more than 1. In this regression technique, the best fit line is not a straight line instead it is in the form of a curve.

d) Support Vector Machines:- Support Vector Machines can be applied to regression problems as well as Classification. It contains all the features that characterises maximum margin algorithm. Linear learning machine maps a non-linear function into high dimensional kernel-induced feature space. The system capacity will be controlled by parameters that do not depend on the dimensionality of feature space.

e) Decision Tree Regression: - Decision tree builds regression models in the form of a tree structure. It breaks down the data into smaller subsets and while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**.

f) Random Forest Regression:- Random Forest is also one of the algorithms used in regression technique. It is very a flexible, easy to use machine learning algorithm that produces, even without hyper -parameter tuning, a great result most of the time. It is also one of the most widely used algorithms because of its simplicity and the fact that it can used for both regression and classification tasks. The forest it builds is an ensemble of Decision Trees, most of the time trained with the **"Bagging"** Method.

Other than these we have regularized regression models like **Ridge, LASSO** and **Elastic Net regressions** which are used to select the key parameters and these is also **Bayesian regression** which works with the **Bayes theorem.**

2.3.2) Types of Classification Algorithms

There are many Classification Algorithms in Machine Learning, which can be used for different classification applications. Some of the main classification algorithms are as follows :

a) Logistic Regression/Classification:-Logistic regression falls under the category of supervised learning; it measures the relationship between the dependent variable which is categorical with one or more than one independent variables by estimating probabilities using a **Logistic/Sigmoid** function. Logistic regression can generally be used when the dependent variable is **Binary** or **Dichotomous**. It means that the dependent variable can take only two possible values like **"Yes or No"**, **"Living or Dead"**.

b) K -Nearest Neighbours:- k-NN algorithm is one of the most straightforward algorithms in classification, and it is one of the most used ML algorithms. An object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its k nearest neighbours. It can also use for regression — output is the value of the object (**predicts continuous values**). This value is the **average** (**or median**) of the values of its k nearest neighbours.

c) Naive Bayes: - Naive Bayes is a type of Classification technique based on Bayes' theorem, with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other function. Naive Bayes model is accessible to build and particularly useful for extensive datasets.

d) Decision Tree Classification:- Decision tree builds classification models in the form of a **tree structure**. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a **classification** or **decision**. The first decision node in a tree which corresponds to the best predictor called **Root Node**. Decision trees can handle both **categorical** and **numerical** data.

e) Support Vector Machines: - A Support Vector Machine is a type of Classifier, in which a discriminative classifier is formally defined by a separating hyperplane. The algorithm outputs an optimal hyperplane which categorises new examples. In two dimensional space, this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

f) Random Forest Classification:- Random Forest is a supervised learning algorithm.

It creates a forest and makes it somehow random. The forest it builds is an ensemble of Decision Trees, most of the times the decision tree algorithm trained with the **"Bagging"** Method. The general idea of the bagging method is that a combination of learning models increases the overall result. And Random Forest is also very powerful to find the variable importance in Classification/ Regression problems.

2.3.3) Types of Unsupervised Learning

Clustering is the type of unsupervised learning in which an unlabelled data is used to draw inferences. It is the process of grouping similar entities together. The goal of this unsupervised machine learning technique is to find similarities in the data points and group similar data points together and also to figure out which cluster should a new data point belong to.

Types of Clustering Algorithms:- There are many Clustering algorithms in machine learning, which can be used for different clustering applications. Some of the main clustering algorithms are as follows :

a) Hierarchical Clustering:- Hierarchical clustering is one of the algorithms of clustering technique, in which similar data is grouped in a cluster. It is an algorithm that builds the hierarchy of clusters. This algorithm starts with all the data points assigned to a bunch of their own. Then, two nearest groups are merged into the same cluster. In the end, this algorithm terminates when there is only a single cluster left.

It starts by assigning each data point to its bunch. Finds the closest pair using Euclidean distance and merges them into one cluster. This process is continued until all data points are clustered into a single cluster.

b) K -Means Clustering:- K-Means clustering is one of the algorithms of clustering technique, in which similar data is grouped into a cluster. K-means is an iterative algorithm that aims to find local maxima in each iteration. It starts with K as the input which is the desired number of clusters. Input k centroids in random locations in your space. Now, with the use of the **Euclidean Distance Method**, calculates the distance between data points and centroids, and assign data point to the cluster which is close to its centroid. Re calculate the cluster centroids as a mean of data points attached to it. Repeat until no further changes occur.

Types of Dimensionality Reduction Algorithms:- There are many dimensionality reduction algorithms in machine learning, which are applied for different dimensionality reduction applications. One of the main dimensionality reduction techniques is Principal Component Analysis (PCA) / Factor Analysis.

Principal Component Analysis (Factor Analysis):- Principal Component Analysis is one of the algorithms of Dimensionality reduction. In this technique, it transforms data into a new set of variables from input variables, which are the linear combination of real variables. These Specific new set of variables are known as principal components. As a result of the transformation, the first primary component will have the most significant possible variance, and each following component in has the highest possible variance under the constraint that it is orthogonal to the above components. Keeping only the best $m < n$ components, reduces the data dimensionality while retaining most of the data information.

2.4 Choosing and Comparing Models through Pipelines

When you work on machine learning project, you often end up with multiple good models to choose from. Each model will have different performance characteristics. Using resampling methods like k-fold cross validation; you can get an estimate of how accurate each model may be on unseen data. You need to be able to use these estimates to choose one or two best models from the suite of models that you have created.

2.4.1) Model Validation

When you are building a predictive model, you need to evaluate the capability or generalization power of the model on unseen data. This is typically done by estimating accuracy using data that was not used to train the model, often referred as cross validation.

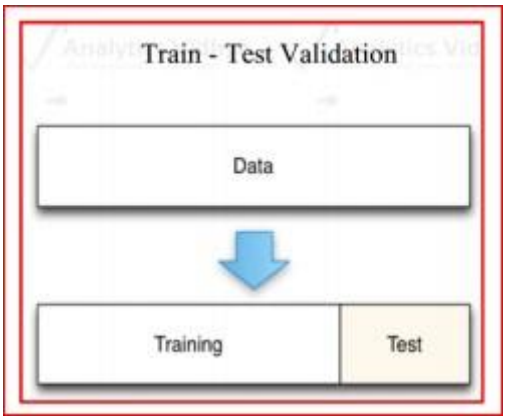


Fig. 2.4.1 Model Validation

A few common methods used for Cross Validation:

1) The Validation Set Approach (Holdout Cross Validation)

In this approach, we reserve large portion of dataset for **training** and rest remaining portion of the data for **model validation.** Ideally people will use 70-30 or 80-20 percentages for **training** and **validation** purpose respectively.

A major disadvantage of this approach is that, since we are training a model on a randomly chosen portion of the dataset, there is a huge possibility that we might missout on some interesting information about the data which, will lead to a **higher bias**.

2) K-Fold Cross Validation

As there is never enough data to train your model, removing a part of it for validation may lead to a problem of under fitting. By reducing the training data, we risk losing important patterns/ trends in data set, which in turn increases error induced by bias. So, what we require is a method that provides ample data for training the model and also leaves ample data for validation. K Fold cross validation does exactly that.

In K Fold Cross Validation, the data is divided into **k subsets**. Now the holdout method is repeated k times, such that each time, one of the k subsets is used as the test set/ validation set and the other k-1 subsets are put together to form a training set. The error

estimation is averaged over all k trials to get total effectiveness of our model. As can be seen, every data point gets to be in a validation set exactly once, and gets to be in a training set k-1 times. This significantly reduces the bias as we are using most of the data for fitting, and also significantly reduces variance as most of the data is also being used in validation set. Interchanging the training and test sets also adds to the effectiveness of this method. As a general rule and empirical evidence, K = 5 or 10 is preferred, but nothing's fixed and it can take any value.

Below are the steps for it:

- Randomly split your entire dataset into k"folds"
- For each k-fold in your dataset, build your model on k – 1 folds of the dataset. Then, test the model to check the effectiveness for kth fold.
- Record the error you see on each of the predictions.
- Repeat this until each of the k-folds has served as the test set.
- The average of your k recorded errors is called the **Cross-Validation Error** and will serve as your performance metric for the model. Below is the visualization of a k-fold validation when k = 5.

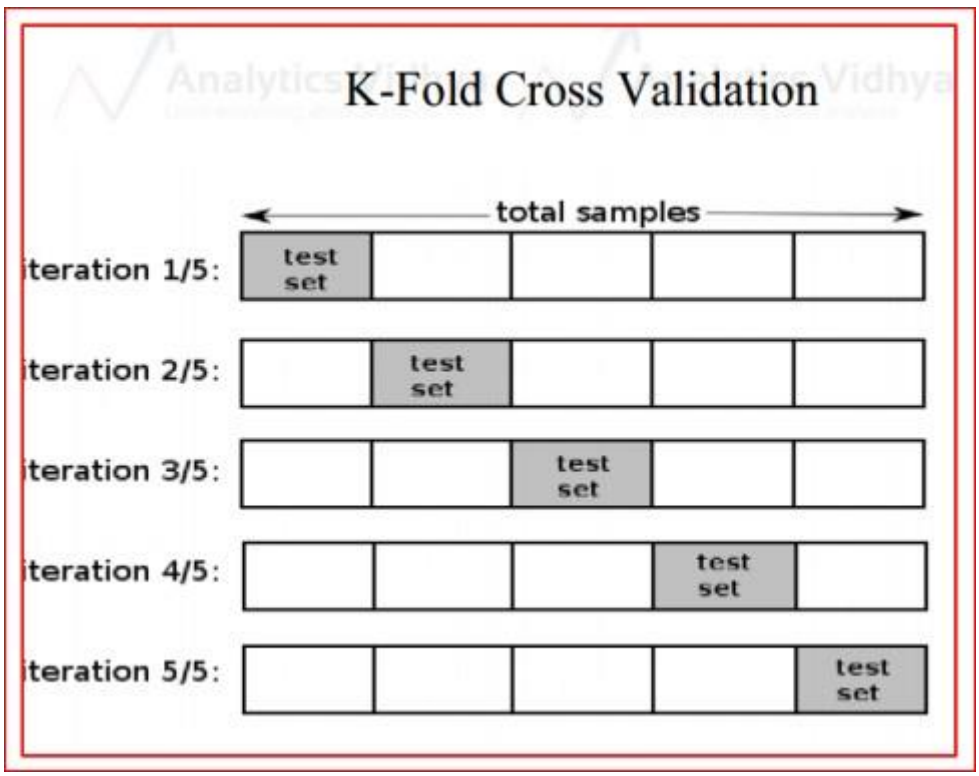


Fig. 2.4.1 K-Fold Cross Validation

How to choose K :

- **Smaller dataset :** 10-fold cross validation is better
- **Moderate dataset :** 5 or 6 fold cross validation works mostly
- **Big dataset :** Train – Val split for validation

Other than this, we have **Leave One Out Cross Validation (LOOCV),** in which each record will be left over from the training and then, the same will be used for testing purpose. This process will be repeated across all the respondents.

2.5 Model Diagnosis with Over Fitting and Under Fitting

2.5.1) Bias and Variance

A fundamental problem with supervised learning is the bias variance trade-off. Ideally, a model should have two key characteristics

1) Sensitive enough to accurately capture the key patterns in the training dataset.

2) Generalized enough to work well on any unseen dataset.

Unfortunately, while trying to achieve the above-mentioned first point, there is an ample chance of over-fitting to noisy or unrepresentative training data points leading to a failure of generalizing the model. On the other hand, trying to generalize a model may result in failing to capture important regularities.

If model accuracy is low on a training dataset as well as test dataset, the model is said to be **Under-Fitting** or that the model has **high bias**. The **Bias** refers to the simplifying assumptions made by the algorithm to make the problem easier to solve. To solve an under-fitting issue or to reduce bias, try including more meaningful features and try to increase the model complexity by trying higher- order interactions The **Variance** refers to sensitivity of a model changes to the training data. A model is giving high accuracy on a training dataset, however on a test dataset the accuracy drops drastically then, the model is said to be **Over-Fitting** or a model that has **high variance.**

To solve the over-fitting issue, try to reduce the number of features, that is, keep only the meaningful features or try regularization methods that will keep all the features. Ideal model will be the trade-off between Under Fitting and Over Fitting like mentioned in the below picture.

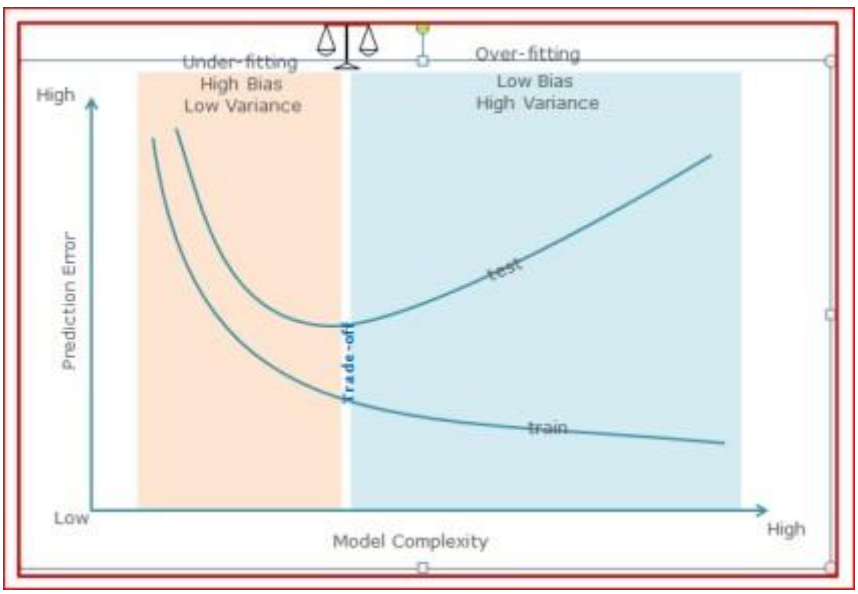


Fig. 2.5.1 Bias and Variance

And, the **Hyper parameters** will be tuned in the below mentioned ways to reach the optimal solution: 1) Grid Search

2) Random Search

3) Manual Tuning

2.5.2) Model Performance Matrix

Model evaluation is an integral part of the model development. Based on model evaluation and subsequent comparisons, we can take a call whether to continue our efforts in model enhancement or cease them and select the final model that should be used / deployed.

1. Evaluating Classification Models

Confusion Matrix

Confusion matrix is one of the most popular ways to evaluate a classification model. A **Confusion Matrix** can be created for a binary classification as well as a multi-class classification model.

A confusion matrix is created by comparing the **predicted class label** of a data point with its **actual class label.** This comparison is repeated for the whole dataset and the results of this comparison are compiled in a matrix or tabular format

Table 2.5.2 Confusion Matrix

		Predicted classed		
		Positive (C_0)	Negative (C_1)	
Actual class	Positive (C_0)	a = number of correctly Classified c0 cases	c = number of c_0 cases Incorrectly classified as c_1	Precision = a/(a + c)
	Negative (C_0)	b = number of c_1 cases Incorrectly classified as c_0	d = number of correctly classified c_1 cases	
		Sensitivity (Recall) = a/(a+b)	Specificity = d/c+d	Accuracy = (a+b)(a+b+c+d)

Specificity : The ratio of actual negative cases that are identified correctly.

shows an example confusion matrix.

. Example of classifications Accuracy measurement

		Predicted classed		
		Positive (C_0)	Negative (C_1)	
Actual class	Positive (C_0)	80	30	Precision = 70/110=0.63
	Negative (C_1)	40	90	
		Recall=80/120=0.67	Specificity = 90/240=0.75	Accuracy = 80+90/240=0.71

And, below are the various measures that will be used to assess the performance of the model based on the requirement of the problem and as well as data.

Table 2.5.2 Model Performance Matrix

Metric	Description	Formula
Accuracy	What% of predictions were Correct?	(TP + TN)/(TP + TN + EP + FN)
Misclassification rate	What % of prediction is wrong?	(FP + FN)/(TP + TN + FP + FN)
True positive rate OR Sensitivity or recall (completeness)	What % of positive cases did Model catch?	TP/(FN + TP)
False positive Rate	What % 'NO' were predicted as 'Yes'?	FP/FP+TN)
Specificity	What % 'NO' were predicted as 'NO'?	TN/(TN + FP)
Precision(exactness)	What % of positive predictions Were correct?	TP(TP + FP)
FI score	Weighted average of precision And recall	2*((precision*recall)/ (precision + recall))

2. Regression Model Evaluation

A regression line predicts the **y** values for a given **x** value. Note that the values are around the average. The prediction error (called as **Root-Mean-Square Error** or **RSME**) is given by the following formula :

$$RMSE = \sqrt{\frac{\sum_{k=0}^{n} \left(\hat{Y}_k - y_k\right)^2}{n}}$$

And, the regression will also assessed by **R Square** (**Coefficient of Determination**).

3. Evaluating Unsupervised Models

The Unsupervised algorithms will be assessed by the profile of the factors/ clusters which were derived through the models.

2.6 Overall Process of Machine Learning

To put overall process together, below is the picture that describes the road map for building ML Systems

CHAPTER 3

MACHINE LEARNING – AT WORK

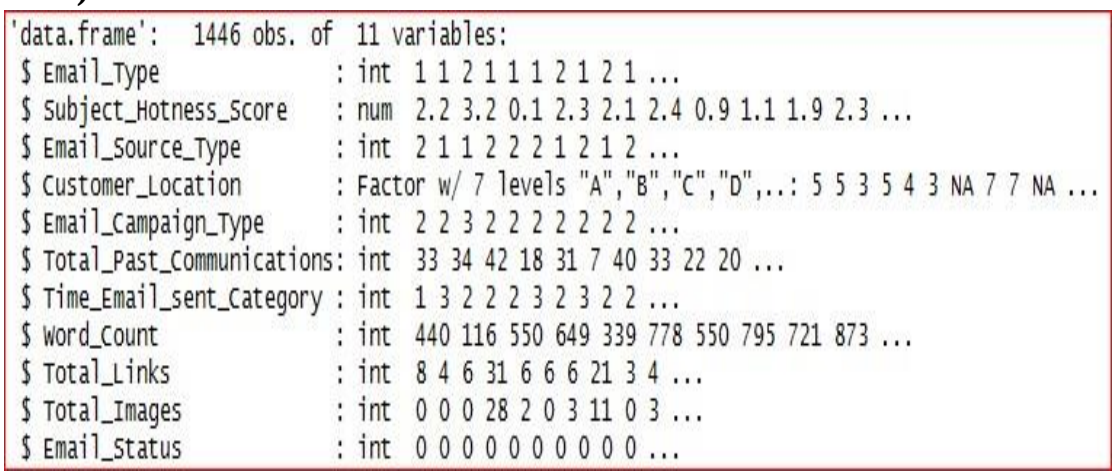
MACHINE LEARNING - AT WORK

3.1 An Approach to the Problem

In order to carry out the analysis, we have randomly extracted records from the computational data and the information of the same is mentioned in **Chapter 1.** In this **Chapter**, we are going to discuss about the results of different Machine Learning Methods used in order to obtain the solution for the problem mentioned in **Chapter 1**.

As mentioned in **Chapter 2**, the first step of a ML Algorithm is Data cleaning and preparing data for the modeling. As a first step, we have to check whether the data was read properly and all the scale types are as per the data.

3.1.1) Structure of the Data



Output : Fig. 3.1.1 Structure of the Data

The above output gives glance of a data set and identified some of the variables need the scale correction and changed them into respective factors.

3.1.2) Converting Data Types

```
'data.frame':   1446 obs. of  11 variables:
 $ Email_Type                : Factor w/ 2 levels "1","2": 1 1 2 1 1 1 2 1 2 1 ...
 $ Subject_Hotness_Score     : num  2.2 3.2 0.1 2.3 2.1 2.4 0.9 1.1 1.9 2.3 ...
 $ Email_Source_Type         : Factor w/ 2 levels "1","2": 2 1 1 2 2 2 1 2 1 2 ...
 $ Customer_Location         : Factor w/ 7 levels "A","B","C","D",..: 5 5 3 5 4 3 NA 7 7 NA ...
 $ Email_Campaign_Type       : Factor w/ 3 levels "1","2","3": 2 2 3 2 2 2 2 2 2 2 ...
 $ Total_Past_Communications: int  33 34 42 18 31 7 40 33 22 20 ...
 $ Time_Email_sent_Category : Factor w/ 3 levels "1","2","3": 1 3 2 2 2 3 2 3 2 2 ...
 $ Word_Count                : int  440 116 550 649 339 778 550 795 721 873 ...
 $ Total_Links               : int  8 4 6 31 6 6 6 21 3 4 ...
 $ Total_Images              : int  0 0 0 28 2 0 3 11 0 3 ...
 $ Email_Status              : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Output : Fig. 3.1.2 Converting Data Types

From the above output, we can observe that Email Type, Email Source Type, Email Campaign Type, Total Email sent Category and Email Status are converted into factors.

3.1.3) Summary of the Data

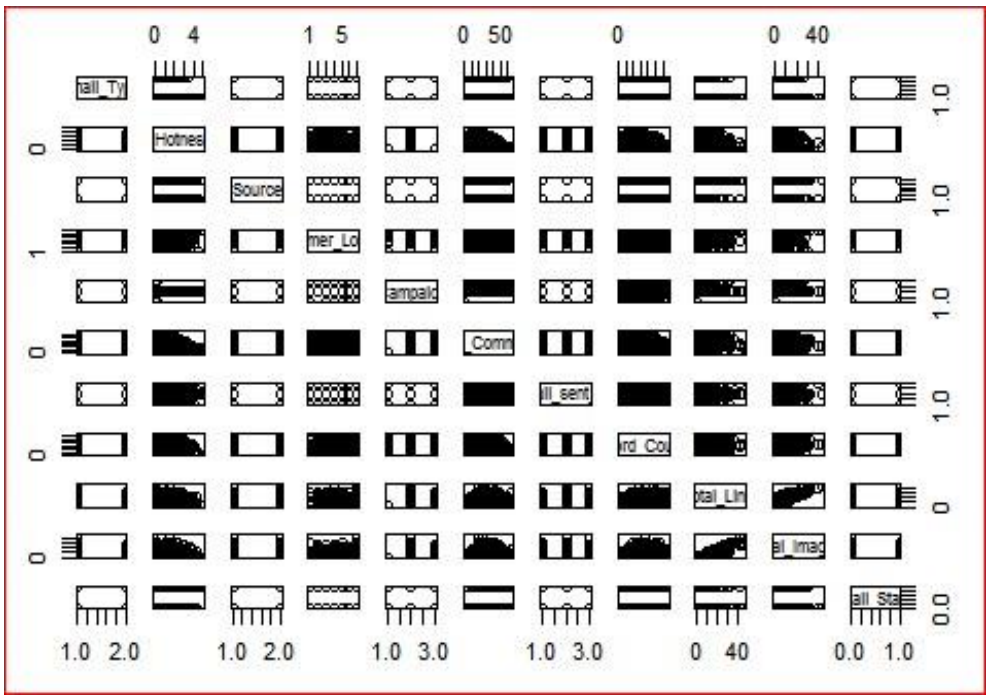
```
  Email_Type    Subject_Hotness_Score Email_Source_Type Customer_Location Email_Campaign_Type
Min.   :1.000   Min.   :0.0000       Min.   :1.000     G       :478       Min.   :1.000
1st Qu.:1.000   1st Qu.:0.2000       1st Qu.:1.000     E       :236       1st Qu.:2.000
Median :1.000   Median :0.3000       Median :1.000     D       :160       Median :2.000
Mean   :1.289   Mean   :0.8385       Mean   :1.438     C       :117       Mean   :2.407
3rd Qu.:2.000   3rd Qu.:1.3000       3rd Qu.:2.000     F       : 87       3rd Qu.:3.000
Max.   :2.000   Max.   :5.0000       Max.   :2.000     (Other):101       Max.   :3.000
                                                       NA's    :267
Total_Past_Communications Time_Email_sent_Category   Word_Count    Total_Links
Min.   : 0.0              Min.   :1.000            Min.   :  40   Min.   : 1.00
1st Qu.:23.0              1st Qu.:2.000            1st Qu.: 484   1st Qu.: 6.00
Median :34.0              Median :2.000            Median : 653   Median : 9.00
Mean   :33.1              Mean   :2.001            Mean   : 658   Mean   :10.34
3rd Qu.:43.0              3rd Qu.:2.000            3rd Qu.: 840   3rd Qu.:14.00
Max.   :67.0              Max.   :3.000            Max.   :1316   Max.   :46.00
NA's   :144                                                       NA's   :45
 Total_Images     Email_Status
Min.   : 0.000   Min.   :0.0000
1st Qu.: 0.000   1st Qu.:0.0000
Median : 0.000   Median :1.0000
Mean   : 3.629   Mean   :0.5726
3rd Qu.: 6.000   3rd Qu.:1.0000
Max.   :43.000   Max.   :1.0000
NA's   :42
```

Output : Fig. 3.1.3 Summary of the Data

The above output gives the clear understanding of data in terms of any central tendencies, proportions, minimum, maximum and any null values.

3.1.4) Understanding Data Visually

Also, look at the data visually to understand the relationships between and within the variables



Output : Fig. 3.1.4 Understanding Data Visually It gives the visual relationship between the variables.

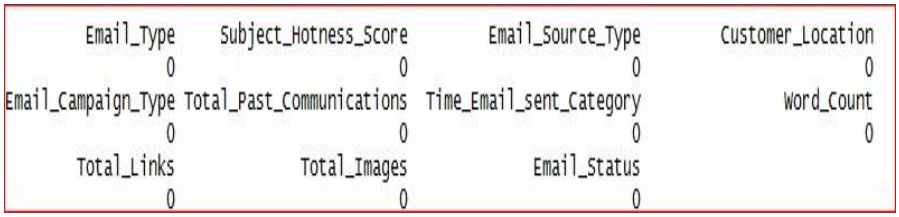
3.1.5) Checking for Missing Values

The missing values for the continuous variables will be imputed using Mean / Median value of the valid records and the categorical variables will be imputed using Mode value.

Then, check if there are any missing values in the data

```
          Email_Type     Subject_Hotness_Score         Email_Source_Type      Customer_Location
            0.000000                  0.000000                  0.000000              18.464730
Email_Campaign_Type Total_Past_Communications  Time_Email_sent_Category          Word_Count
            0.000000                  9.958506                  0.000000              0.000000
         Total_Links            Total_Images              Email_Status
            3.112033                  2.904564                  0.000000
```

Output : Fig. 3.1.5 Checking for Missing Values



Email_Type	Subject_Hotness_Score	Email_Source_Type	Customer_Location
0	0	0	0
Email_Campaign_Type	Total_Past_Communications	Time_Email_sent_Category	Word_Count
0	0	0	0
Total_Links	Total_Images	Email_Status	
0	0	0	

Output : Fig. 3.1 5 No Missing Values Now,

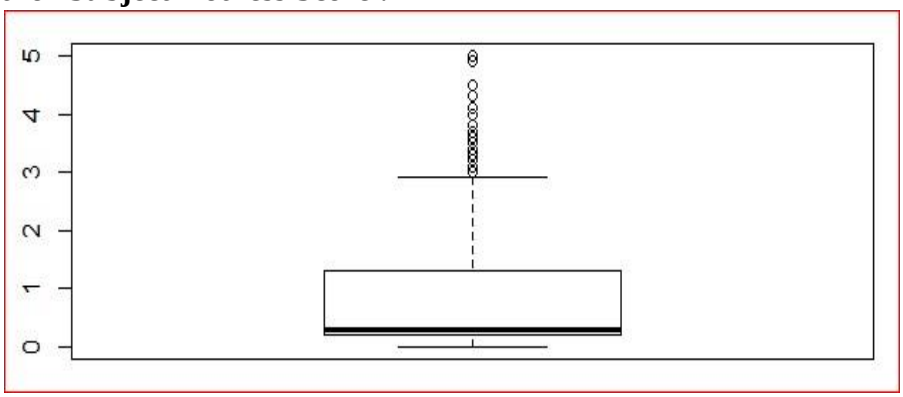
the above output has **no** missing values.

3.1.6) Checking for Outliers

We used Box-plots to check for outliers in each of the continuous variables.

Subject Hotness Score

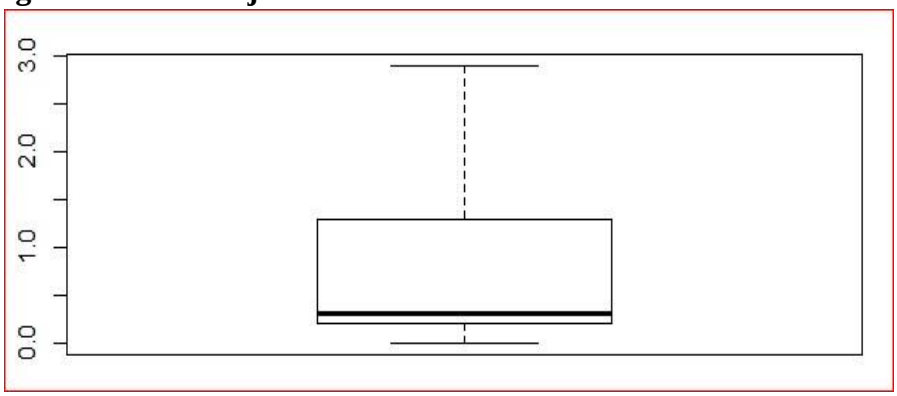
Box plot for Subject Hotness Score :



Output : Fig. 3.1.6 Box plot for Subject Hotness Score

Values more than 95th percentile will be imputed using the 95th percentile value and the values less than 5th percentile value will be imputed using 5th percentile value.

Removing Outliers in Subject Hotness Score :

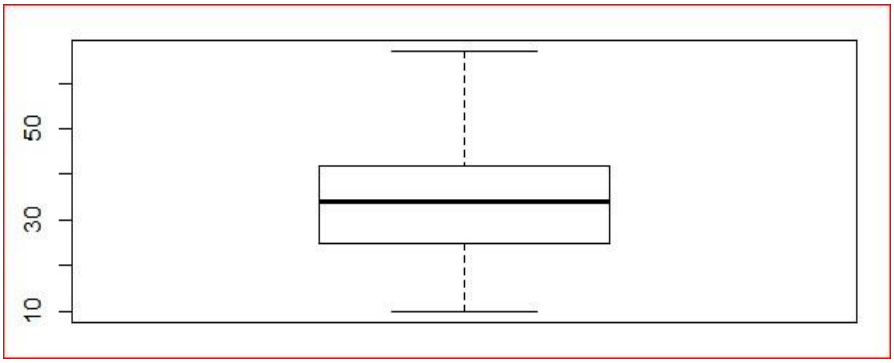


Output : Fig. 3.1.6 Removing Outliers in Subject Hotness Score

Above Box plot has **no** outliers in Subject Hotness Score.

Total Past Communications

Box plot for Total Past Communications :

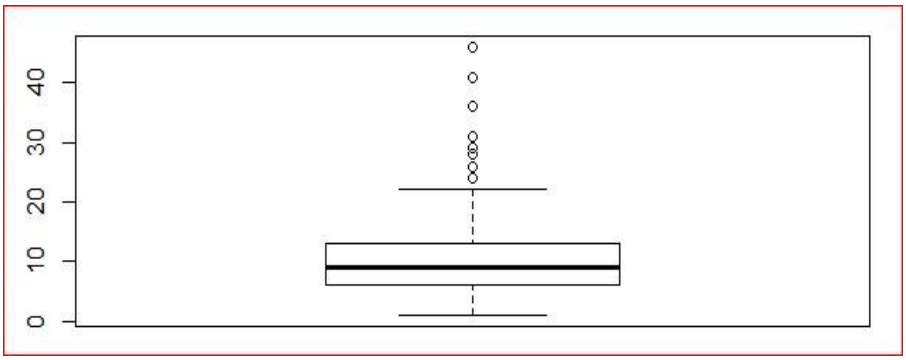


Output : Fig. 3.1.6 Box plot for Total Past Communications

Above Box plot has **no** outliers in Total Past Communications.

Total Links

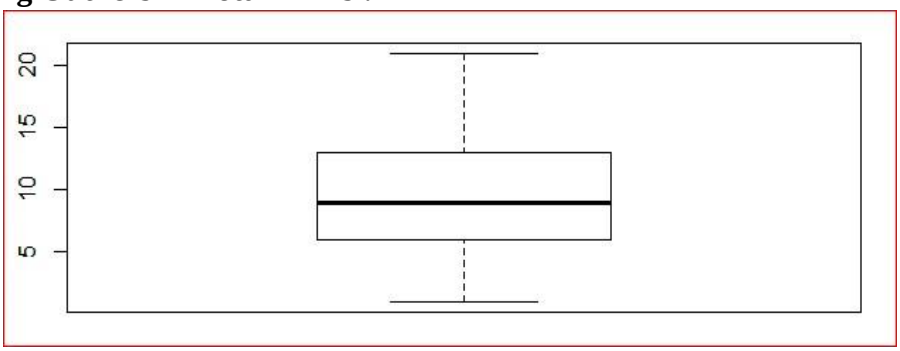
Box plot for Total Links :



Output : Fig. 3.1.6 Box plot for Total Links

Values more than 95th percentile will be imputed using the 95th percentile value and the values less than 5th percentile will be imputed using 5th percentile value.

Removing Outliers in Total Links :

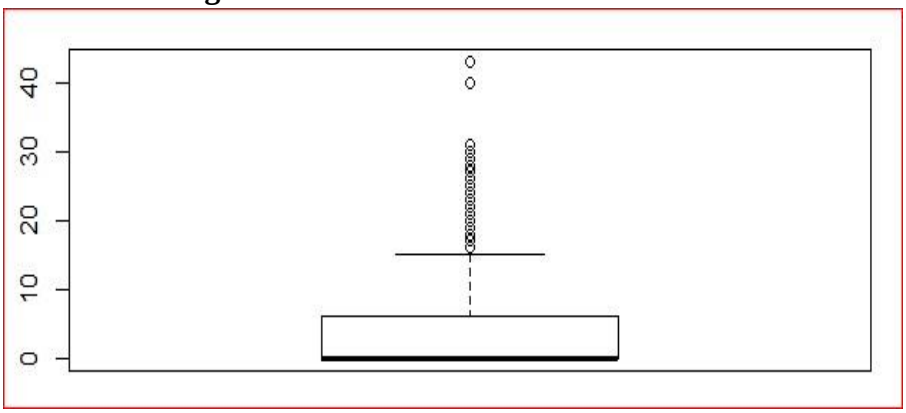


Output : Fig. 3.1.6 Removing Outliers in Total Links

Above Box plot has **no** outliers in Total Links.

Total Images

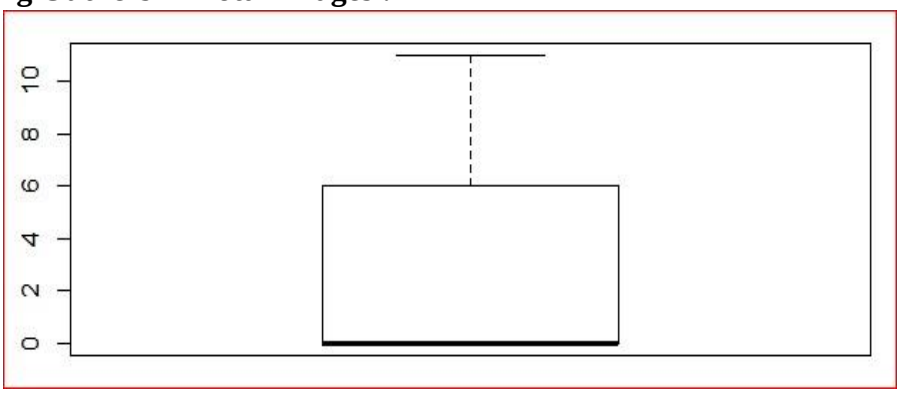
Box plot for Total Images :



Output : Fig. 3.1.6 Box plot for Total Images

Values more than 90th percentile will be imputed using the 90th percentile value and the values less than 10th percentile value will be imputed using 10th percentile value.

Removing Outliers in Total Images :



Output : Fig. 3.1.6 Removing Outliers in Total Images

Above Box plot has **no** outliers in Total Images.

3.1.7 Checking for the significance difference between variables

Categorical Vs Categorical

To test the significance difference between Categorical Vs Categorical variables, we will look at the Chi-Square test value.

Email Status Vs Email Type :

```
        Pearson's Chi-squared test

data:  data$Email_Status and data$Email_Type
X-squared = 0.096366, df = 1, p-value = 0.7562
```

Output : Fig. 3.1.7 Email Status Vs Email Type

From the above table, as the p value is >0.05, we can conclude that there is no significant relationship between Email Status and Email Type.

Email Status Vs Customer Location:

```
        Pearson's Chi-squared test

data:  data$Email_Status and data$Customer_Location
X-squared = 8.0641, df = 6, p-value = 0.2334
```

Output : Fig. 3.1.7 Email Status Vs Customer Location

From the above table, as the p value is >0.05, we can conclude that there is no significant relationship between Email Status and Customer Location.

Email Status Vs Email Campaign Type:

```
        Pearson's Chi-squared test

data:  data$Email_Status and data$Email_Campaign_Type
X-squared = 210.45, df = 2, p-value < 2.2e-16
```

Output : Fig. 3.1.7 Email Status Vs Email Campaign Type

From the above table, as the p value is <0.05, we can conclude that there is a significant relationship between Email Status and Email Campaign Type.

Continuous Vs Categorical

To test the significance difference between Continuous Vs Categorical variables, we will look at the T test/ANOVA.

Here, we applied ANOVA test.

Total Links Vs Email Status:

```
                    Df Sum Sq Mean Sq F value Pr(>F)
data$Email_Status    1     18   18.43   0.667  0.414
Residuals         1444  39886   27.62
```

Output : Fig. 3.1.7 Total Links Vs Email Status

From the above table, as the F value is >0.05, we can conclude that there is no significant relationship between Total Links and Email Status. **Total Images Vs Email Status:**

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
data$Email_Status	1	13	13.24	0.858	0.355
Residuals	1444	22297	15.44		

Output : Fig. 3.1.7 Total Images Vs Email Status

From the above table, as the F value is >0.05, we can conclude that there is no significant relationship between Total Images and Email Status.

3.1.8) Cross Validation Split Data into Train & Test

It order to carry out the analysis, we split the data **train** and **test** to algorithm model by 70% and 30%.

After splitting the data, the size for train and test are **1012** and **434** records.

```
> dim(data)
[1] 1446   11
> dim(training)
[1] 1012   11
> dim(test)
[1] 434  11
```

Output : Fig. 3.1.8 Split Data into Train & Test

The above output gives dimension of data, dimension of training and test data.

```
> names(training)
 [1] "Email_Type"              "Subject_Hotness_Score"   "Email_Source_Type"
 [4] "Customer_Location"       "Email_Campaign_Type"     "Total_Past_Communications"
 [7] "Time_Email_sent_Category" "Word_Count"             "Total_Links"
[10] "Total_Images"            "Email_Status"
```

Output : Fig. 3.1.8 Names of Training Data

The above output gives names of training data.

```
> names(test)
 [1] "Email_Type"              "Subject_Hotness_Score"   "Email_Source_Type"
 [4] "Customer_Location"       "Email_Campaign_Type"     "Total_Past_Communications"
 [7] "Time_Email_sent_Category" "Word_Count"             "Total_Links"
[10] "Total_Images"            "Email_Status"
```

Output : Fig. 3.1.8 Names of Test Data

The above output gives names of test data.

3.1.9) Running Pipeline using K-Fold Validation

We run the model by using K –Fold Validation.

Below is the output for Logistic Regression.

```
Generalized Linear Model

1012 samples
  10 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 810, 809, 809, 810, 810
Resampling results:

  Accuracy   Kappa
  0.7094474  0.4024414
```

Output : Fig. 3.1.9 Running Pipeline using K-Fold Validation

3.1.10) CART

We run the model by using Decision Tree.

Below is the output for Cart.

```
CART

1012 samples
  10 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 810, 809, 809, 810, 810
Resampling results across tuning parameters:

  cp          Accuracy   Kappa
  0.03908046  0.6995415  0.3720640
  0.04367816  0.6945910  0.3515204
  0.22528736  0.6077062  0.1653038

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.03908046.
```

Output : Fig. 3.1.10 CART

3.1.11) KNN

We run the model by using K -Nearest -Neighbours.

Below is the output for KNN.

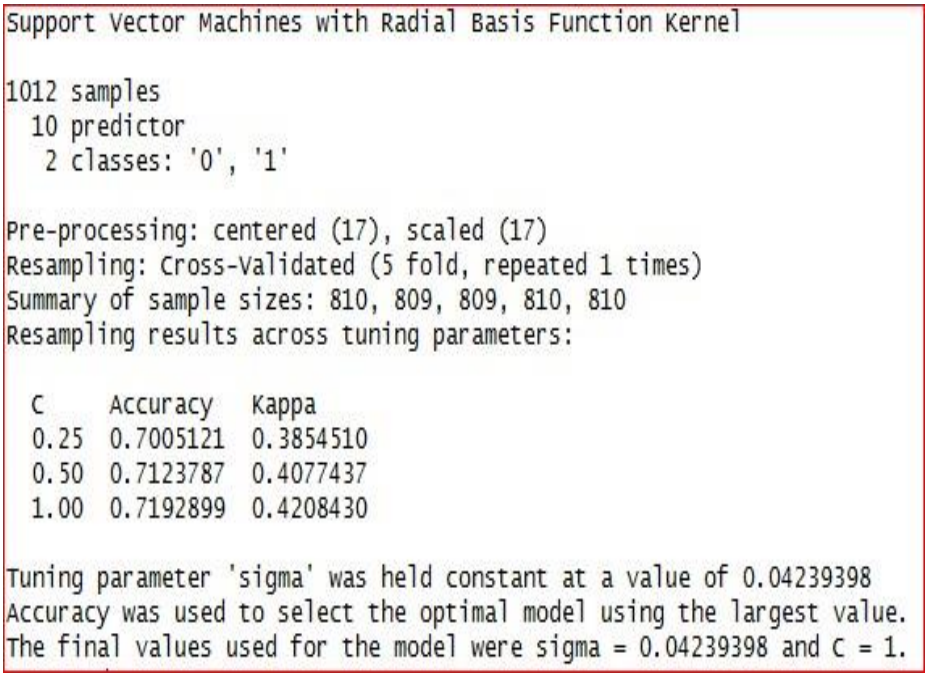
```
k-Nearest Neighbors

1012 samples
  10 predictor
   2 classes: '0', '1'

Pre-processing: centered (17), scaled (17)
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 810, 809, 809, 810, 810
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.6728723  0.3245265
  7  0.6649759  0.3068695
  9  0.6659221  0.3101396

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 5.
```

Output : Fig. 3.1.11 KNN

3.1.12) SVM

We run the model by using Support Vector Machine.

Below is the output for SVM.



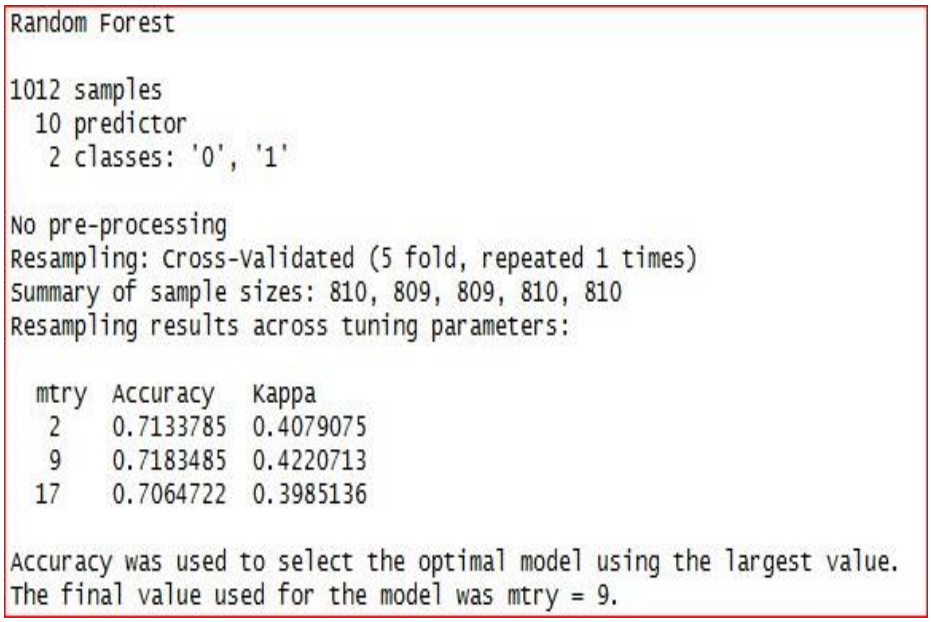
```
Support Vector Machines with Radial Basis Function Kernel

1012 samples
  10 predictor
   2 classes: '0', '1'

Pre-processing: centered (17), scaled (17)
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 810, 809, 809, 810, 810
Resampling results across tuning parameters:

  C     Accuracy   Kappa
  0.25  0.7005121  0.3854510
  0.50  0.7123787  0.4077437
  1.00  0.7192899  0.4208430

Tuning parameter 'sigma' was held constant at a value of 0.04239398
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were sigma = 0.04239398 and C = 1.
```

Output : Fig. 3.1.12 SVM

3.1.13) Random Forest

We run the model by using Random Forest.

Below is the output for Random Forest



```
Random Forest

1012 samples
  10 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (5 fold, repeated 1 times)
Summary of sample sizes: 810, 809, 809, 810, 810
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.7133785  0.4079075
   9    0.7183485  0.4220713
  17    0.7064722  0.3985136

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 9.
```

Output : Fig. 3.1.13 Random Forest

3.1.14) Comparing Algorithms

We can observe the best model by comparing accuracy of Logistic Regression, Support Vector Machine, K Nearest Neighbours, Decision Tree and Random Forest. Below is the output for choosing best model among all these algorithms.

```
call:
summary.resamples(object = results)

Models: logistic, svm, knn, DT, rf
Number of resamples: 5

Accuracy
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
logistic 0.6551724 0.6732673 0.6980198 0.7094474 0.7178218 0.8029557    0
svm      0.6584158 0.6732673 0.6945813 0.7192899 0.7475248 0.8226601    0
knn      0.6336634 0.6336634 0.6847291 0.6728723 0.6980198 0.7142857    0
DT       0.6534653 0.6584158 0.6945813 0.6995415 0.7227723 0.7684729    0
rf       0.6584158 0.6995074 0.7079208 0.7183485 0.7574257 0.7684729    0

Kappa
             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
logistic 0.2979943 0.3299829 0.3710056 0.4024414 0.4237814 0.5894428    0
svm      0.2905426 0.3280919 0.3710145 0.4208430 0.4829871 0.6315789    0
knn      0.2423720 0.2487687 0.3469388 0.3245265 0.3816120 0.4029412    0
DT       0.2536683 0.2781978 0.3522388 0.3720640 0.4455446 0.5306705    0
rf       0.2905426 0.3891273 0.4035632 0.4220713 0.5074151 0.5197080    0
```

Output : Fig. 3.1.14 Comparing Algorithms

From the above by **5** algorithms , we can observe that **Random Forest** is the best algorithm.

3.1.15) Finding the Key variables using Random Forest

In order to find the key variables we ran the Random Forest using grid search and obtained key hyperparameters and key variables.

Tuning the Random Forest

From the Random Forest, we observe that **Random Forest** is the best model to classify our data.

To obtain better accuracy, we tune the parameters using Random Forest Algorithm.

```
1012 samples
  10 predictor
   2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 3 times)
Summary of sample sizes: 910, 911, 910, 911, 911, 911, ...
Resampling results across tuning parameters:

  mtry  ntree  Accuracy   Kappa
  1     100    0.6927551  0.3570582
  1     200    0.6977094  0.3691796
  1     500    0.6990295  0.3712001
  2     100    0.7102092  0.4055241
  2     200    0.7112027  0.4072439
  2     500    0.7125097  0.4103722
  3     100    0.7177419  0.4223759
  3     200    0.7134674  0.4122571
  3     500    0.7184247  0.4237552
  4     100    0.7167389  0.4196104
  4     200    0.7144576  0.4151769
  4     500    0.7127879  0.4121998
  5     100    0.7118140  0.4098290
  5     200    0.7101899  0.4065748
  5     500    0.7098632  0.4057079

Accuracy was used to select the optimal model using the largest value.
The final values used for the model were mtry = 3 and ntree = 500.
```

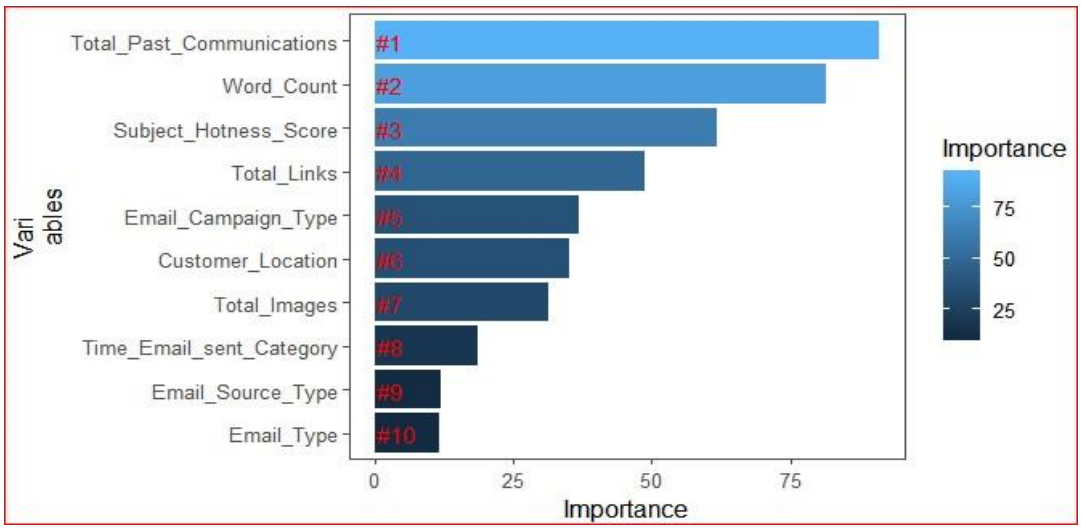
Output : Fig. 3.1.15 Tuning the Random Forest

From the above output, we choosed the best parameters from the gid search and applied Random Forest with those parameters and obtained the below **Variable Importance**.

3.1.16) Variable Importance

Variable Importance can be obtained using Random Forest.

Below is the output to know Variable Importance.



Output : Fig. 3.1.16 Variable Importance

Then, we applied Logistic Regression with the identified key variables and obtained the below results for both train and test data.

3.1.17) Confusion Matrix for Train/Test Data

Confusion Matrix for Train Data

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 281 125
         1 154 452

               Accuracy : 0.7243
                 95% CI : (0.6957, 0.7516)
    No Information Rate : 0.5702
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.4329
 Mcnemar's Test P-Value : 0.09368

            Sensitivity : 0.6460
            Specificity : 0.7834
         Pos Pred Value : 0.6921
         Neg Pred Value : 0.7459
             Prevalence : 0.4298
         Detection Rate : 0.2777
   Detection Prevalence : 0.4012
      Balanced Accuracy : 0.7147

       'Positive' Class : 0
```

Output : Fig. 3.1.17 Confusion Matrix for Train Data

From the above output, we obtained Confusion Matrix for train data.

In train data, we obtained the accuracy as **72.43**

Confusion Matrix for Test Data

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 118  53
         1  65 198

               Accuracy : 0.7281
                 95% CI : (0.6836, 0.7694)
    No Information Rate : 0.5783
    P-Value [Acc > NIR] : 6.631e-11

                  Kappa : 0.4375
 Mcnemar's Test P-Value : 0.3112

            Sensitivity : 0.6448
            Specificity : 0.7888
         Pos Pred Value : 0.6901
         Neg Pred Value : 0.7529
             Prevalence : 0.4217
         Detection Rate : 0.2719
   Detection Prevalence : 0.3940
      Balanced Accuracy : 0.7168

       'Positive' Class : 0
```

Output : Fig. 3.1.17 Confusion Matrix for Test Data

From the above output, we obtained Confusion Matrix for test data.

In test data, we obtained the accuracy as **72.81**

CHAPTER 4

SUMMARY

SUMMARY

CONCLUSION

In order to classify whether the E-Mail was opened by the customer or not, we developed an algorithm in which we applied Pipeline technique as well as Random Forest to choose the best model and key variables.

Once we identify the key variables, we run the model with only those key variables.

We have to test the train and test data.

The accuracy of **train** and **test** data are **72.43** and **72.81** respectively.

Since the accuracy of train and test data are almost same, we can say that our model is a **Generalized Model.**

Hence, we can apply our model for future predictions.

APPENDIX

R CODE

DATA SET

APPENDIX

R CODE

```
getwd( )
setwd("C:/DMMLT")
getwd( )

# READING DATA
data<-read.csv('email.csv',stringsAsFactors = T, na.strings = c("","  ","NA","?",NA))
View(data) head(data,10) tail(data)

# STRUCTURE OF A DATA SET
names(data)
pairs(data)
str(data)
summary(data)
dim(data)

# CORRECTING DATA TYPES
data$Email_Status<-as.factor(data$Email_Status)
data$Time_Email_sent_Category<-as.factor(data$Time_Email_sent_Category)
data$Email_Type<-as.factor(data$Email_Type) data$Email_Source_Type<-
as.factor(data$Email_Source_Type) data$Email_Campaign_Type<-
as.factor(data$Email_Campaign_Type) str(data)

# FIND MISSING VALUES IN DATA SET IF ANY
sum(is.na(data))
# MISSING VALUE PROPORTION FOR ALL THE VARIABLES
sapply(data, function(df) {
  (sum(is.na(df)==TRUE)/ length(df))*100;
})
```

 # IMPUTATION

```
install.packages("Hmisc") library(Hmisc)
data$Total_Past_Communications[is.na(data$Total_Past_Communications)]<median(
data$Total_Past_Communications,na.rm=T)
data$Total_Links[is.na(data$Total_Links)]<-median(data$Total_Links,na.rm=T)
data$Total_Images[is.na(data$Total_Images)]<-median(data$Total_Images,na.rm=T)
#Imputing with most frequent occuring level
data$Customer_Location[is.na(data$Customer_Location)]<-'G'
sum(is.na(data))

sapply(data, function(df) {
  (sum(is.na(df)==TRUE)/ length(df))*100;
})
```

 # REMOVING OUTLIERS

```
boxplot(data$Subject_Hotness_Score)
boxplot(data$Total_Past_Communications)
boxplot(data$Total_Links)
boxplot(data$Total_Images)

data$Subject_Hotness_Score[data$Subject_Hotness_Score>quantile(data$Subject_H
otness_Score, 0.95)] <- quantile(data$Subject_Hotness_Score, 0.95)
data$Total_Links[data$Total_Links>quantile(data$Total_Links,     0.95)]  <-
quantile(data$Total_Links, 0.95)
data$Total_Images[data$Total_Images>quantile(data$Total_Images,          0.90)]  <-
quantile(data$Total_Images, 0.90)

boxplot(data$Subject_Hotness_Score)
boxplot(data$Total_Links)
boxplot(data$Total_Images)

names(data)
```

 # HYPOTHESIS TESTING

```
chisq.test(data$Email_Status, data$Email_Type, correct=FALSE)
chisq.test(data$Email_Status, data$Customer_Location , correct=FALSE)
chisq.test(data$Email_Status, data$Email_Campaign_Type , correct=FALSE)
```

 # ANOVA

```
x1=aov(data$Total_Links  ~  data$Email_Status)
summary(x1)  x2=aov(data$Total_Images  ~
data$Email_Status) summary(x2) summary(data)
```

```
train_rows<- sample(1:nrow(data), size=0.7*nrow(data))
train_rows training <- data[train_rows, ] test <- data[-
train_rows, ] dim(data) dim(training) dim(test)
names(training) names(test) str(training)
```

 # SET-UP TEST OPTIONS

```
install.packages("dplyr")
library(dplyr)
install.packages("caret")
library(caret)
```

```
control <- trainControl(method="repeatedcv", number=5)
seed <- 7 metric <- "Accuracy"
```

 # MULTINOMINAL REGRESSION

```
set.seed(seed) fit.glm1 <- train(Email_Status~., data=training, method="glm",
metric=metric, trControl=control) print(fit.glm1)
```

 # CART

```
set.seed(seed) fit.cart <- train(Email_Status~., data=training, method="rpart",
metric=metric, trControl=control) print(fit.cart)
```

 # KNN

```
set.seed(seed) fit.knn <- train(Email_Status~., data=training, method="knn",
metric=metric, preProc=c("center", "scale"), trControl=control) print(fit.knn)
```

 # SVM

```
set.seed(seed) fit.svm <- train(Email_Status~., data=training, method="svmRadial", metric=metric, preProc=c("center", "scale"), trControl=control, fit=FALSE) print(fit.svm)
```

 # RANDOM FOREST

```
set.seed(seed) fit.rf <- train(Email_Status~., data=training, method="rf", metric=metric, trControl=control) print(fit.rf)
```

 # COMPARE ALGORITHMS

```
results <- resamples(list(logistic=fit.glm1,svm=fit.svm, knn=fit.knn, DT=fit.cart,rf=fit.rf ))
```

 # TABLE COMPARISON

```
summary(results)
```

 # BOX PLOT COMPARISON

```
bwplot(results)
```

 # DOT-PLOT COMPARISON

```
dotplot(results)
```

 # CHECKING RANDOM FOREST

```
customRF <- list(type = "Classification", library = "randomForest", loop = NULL)
customRF$parameters <- data.frame(parameter = c("mtry", "ntree"), class = rep("numeric", 2), label = c("mtry", "ntree")) customRF$grid <- function(x, y, len = NULL, search = "grid") {} customRF$fit <- function(x, y, wts, param, lev, last, weights, classProbs, ...) {   randomForest(x, y, mtry = param$mtry, ntree=param$ntree, ...)
}
customRF$predict <- function(modelFit, newdata, preProc = NULL, submodels = NULL)
  predict(modelFit, newdata)
customRF$prob <- function(modelFit, newdata, preProc = NULL, submodels = NULL)
```

```
  predict(modelFit, newdata, type = "prob")
customRF$sort <- function(x) x[order(x[,1]),]
customRF$levels <- function(x) x$classes

library(caret)
install.packages("randomForest")
library(randomForest)

control <- trainControl(method="repeatedcv", number=10, repeats=3)
#tunegrid <- expand.grid(.mtry=c(1:5), .ntree=c(100, 150, 200, 250))
tunegrid <- expand.grid(.mtry=c(1:5), .ntree=c(100,200,500))

set.seed(seed)
custom <- train(Email_Status~Email_Type+
                Subject_Hotness_Score+     Email_Source_Type+
                Customer_Location+         Email_Campaign_Type+
                Total_Past_Communications+ Time_Email_sent_Category+
                Word_Count+                Total_Links+
                Total_Images,data    =    training,method=customRF,    tuneGrid=tunegrid,
trControl=control)

print(custom)

library('randomForest')

rf_model <- randomForest(Email_Status~Email_Type+
                         Subject_Hotness_Score+     Email_Source_Type+
                         Customer_Location+         Email_Campaign_Type+
                         Total_Past_Communications+ Time_Email_sent_Category+
                         Word_Count+                Total_Links+
                         Total_Images,data = training,ntree=200,mtry=2,trControl=control)
```

 # VARIABLE IMPORTANCE

#We can have a look at the variable importance of our random forest model below :

```
install.packages("ggthemes")
library('ggthemes')
library('dplyr')

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

 # Create a rank variable based on importance

```
rankImportance <- varImportance %>%   mutate(Rank =
paste0('#',dense_rank(desc(Importance))))
```

 # Use ggplot2 to visualize the relative importance of variables

```
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
y = Importance, fill = Importance)) +   geom_bar(stat='identity')
+   geom_text(aes(x = Variables, y = 0.5, label = Rank),
hjust=0, vjust=0.55, size = 4, colour = 'red') +   labs(x =
'Variables') +   coord_flip() +   theme_few()
```

#FINAL MODEL

```
install.packages("nnet")
library(nnet)
mymodel <- glm(Email_Status~Subject_Hotness_Score+Customer_Location+
Total_Past_Communications+Word_Count+Total_Links+Total_Images+Email_Cam
paign_Type, data=training,family='binomial')
```

 # PREDICTION

#For Train Data p1 <- predict(mymodel,
newdata=training, type = 'response') pred1 <- ifelse(p1>0.5,
1, 0) library(caret)

```
confusionMatrix(as.factor(pred1),training$Email_Status)
```

#For Test Data p2=predict(mymodel, newdata=test,

type = 'response') pred2 <- ifelse(p2>0.5, 1, 0)

library(caret)

confusionMatrix(as.factor(pred2),test$Email_Status)

DATA SET

S.No	Email_ Type	Subject_ Hotness_ Score	Email_ Source_ Type	Customer_ Location	Email_ Campaign_ Type	Total_ Past_ Communications	Time_ Email_ sent_ Category	Word_ Count	Total_ Links	Total_ Images	Email_ Status
1	1	2.2	2	E	2	33	1	440	8	0	0
2	1	3.2	1	E	2	34	3	116	4	0	0
3	2	0.1	1	C	3	42	2	550	6	0	0
4	1	2.3	2	E	2	18	2	649	31	28	0
5	1	2.1	2	D	2	31	2	339	6	2	0
6	1	2.4	2	C	2	7	3	778	6	0	0
7	2	0.9	1		2	40	2	550	6	3	0
8	1	1.1	2	G	2	33	3	795	21	11	0
9	2	1.9	1	G	2	22	2	721	3	0	0
10	1	2.3	2		2	20	2	873	4	3	0
11	2	1.5	1		2	30	3	988	8	0	1
12	1	1	2	G	2		1	704	4	0	0
13	2	0.1	2	E	3	29	1	744	2	0	0
14	1	1.5	1	G	2	29	1	577	11	0	0
15	1	0.2	2	C	3	40	2	623		14	0
16	2	0.1	2		3	23	2	939	11	1	0
17	1	0.3	2	D	3		1	608	31	18	0
18	1	0.5	1	D	2	51	3	366	11	2	0
19	1	0.2	1	C	3	41	2	694	9	6	0
20	1	1.4	1		2	39	2	610	9	0	0
21	1	3.1	2	D	2	26	3	490	26	25	0
22	1	1.6	1	D	2	20	2	1140	26	20	1
23	1	0.9	2	E	2	11	1	939	19	14	0
24	1	0.6	2	B	2	44	1	416	14	7	0
25	2	1.5	1	B	2	18	3	684	5	1	0
26	1	0	2	G	2		2	827	16	5	0
27	1	0.1	1	G	3	26	3	904	4	0	0

28	1	1.2	1	F	2	36	2	392	6		0
29	1	2	2	G	2		2	470		4	0
30	1	2	2	G	2	11	3	673		18	0
31	1	2.3	1	G	2	35	2	314	14		0
32	2	0.6	1	F	2	40	2	187	8	0	0
33	2	1.1	1	G	2	37	2	873	9	0	0
34	1	2.9	1	G	2		2	678	8	0	0
35	2	0.1	2	G	3	51	2	392	6	4	1
36	2	0	1	D	3	50	2	490	11	0	1
37	1	2.3	1	C	2	22	2	392		2	0
38	1	2.5	2	D	2		2	419	4	0	0
39	1	0.4	2	E	2	35	1	778	6	3	0
40	1	0.3	2	D	3	35	3	694	24	13	1
41	2	0.4	1	B	2	22	2	733	6	0	0

42	1	1	1	G	2	29	1	827	14	2	0
43	2	1.1	1	C	2	21	2	694	11	1	0
44	1	0.3	2	D	2	39	3	608	11	8	0
45	1	0.3	2		2	20	1	733	6	2	0
46	1	2.6	1	G	2	24	2	518	11	0	0
47	1	1.5	1	G	2	40	2	518	6	4	0
48	1	0.9	2	C	2	30	2	1038	18	11	0
49	1	3.4	1		2	23	1	220		1	0
50	1	0	1	D	3	40	3	593	4	0	0
51	1	0.2	1	E	3	60	1	419	16	7	0
52	2	0.6	1	G	2	29	2	623	4	0	0
53	1	0	1	G	3	49	1	521	7	0	0
54	2	0.1	1	D	3	37	2	1061	8	0	0
55	1	2.9	2	G	2	8	2	623	6	0	0
56	1	0	1	C	2	33	2	868	14	2	0
57	1	0.6	2	C	2	26	2	855	6	5	0
58	1	1.3	1	B	2	38	2	771	6	0	0
59	1	2	2	G	2	23	2	605	2	0	0
60	1	2.8	2	G	2	23	1	806	11	6	0
61	1	0.5	2	E	2	40	2	630	11	9	0
62	1	2.3	2	B	2	23	1	550	8	0	0
63	1	1.6	2	C	2	15	2	922	9	0	0

64	1	0	1		2	41	2	608		0	0
65	1	1.2	2		2	18	2	760	8	6	0
66	1	0.2	2	D	2	24	1	1189	16	5	0
67	1	1.3	2	E	2	23	1	694	14	9	0
68	1	1.8	1	G	2	20	2	1038	11	4	0
69	1	2.3	1	D	2	15	3	934	9	0	0
70	2	0.2	1	G	3	21	2	1173	9	5	0
71	1	2.5	1	E	2	13	2	1061	11	6	0
72	1	0.7	1		2	23	1	1271	24	11	0
73	2	0.2	2		3	45	3	79	4	0	0
74	1	1.9	2		2	11	3	694	4	0	0
75	2	1.5	1	E	2	26	2	700	31	15	0
76	1	3	2	G	2	5	2	311	4	0	0
77	1	0.7	2		2	25	2	704	16	4	0
78	1	3.6	2	B	2	7	2	733	11	7	0
79	2	0	2	E	3	28	2	1038	3	0	0
80	1	0.5	2	C	2	26	2	704	21	4	0
81	1	1.7	2	G	2	20	3	623	31	27	0
82	2	1.6	1	G	2	16	3	880	8	0	0
83	1	1.2	2		2	29	2	827	31	12	0
84	1	0.4	1	G	2	46	2	440	6	0	0
85	1	1.3	2	B	2	21	2	841	18	7	0
86	1	2.4	1	C	2	30	2	282	11	4	0
87	1	0.8	1	G	2	40	2	490	9	1	1

88	1	0.1	2		3	38	2	1014	6	0	0
89	1	0.2	2	C	3	33	2	730	10	4	0
90	1	1.8	1	G	2	30	1	578	14		0
91	1	1.5	2	G	2	37	2	565	28	27	0
92	1	0	1	G	3	48	3	770	13	0	0
93	1	1.2	1		2	18	3	880	16	9	0
94	1	0.9	2	G	2	22	2	751	11	9	0
95	1	1.5	2		2	26	1	673	9	4	0
96	1	0.3	1	E	3	48	2	605	6	0	0
97	1	0.8	1	F	2	40	1	339	6	0	0
98	2	2.3	1	G	2	20	2	419	6	0	0
99	2	0.9	1	C	2	23	3	737	8	0	0

100	2	0.2	2		3	21	2	744	6	0	0
101	2	0.3	1	E	2	26	2	655	2	0	0
102	1	0.5	1	G	2	41	3	542	16	12	0
103	1	2.1	1	C	2		2	771	6	2	0
104	1	1.4	1	C	2	21	3	1189	8	0	0
105	1	1.3	2	F	2	54	1	99	8	0	0
106	1	3.1	2	G	2	23	3	366	8	0	0
107	2	0.1	2	C	3	36	1	440	2	0	0
108	1	0.7	2	G	2	20	2	730	46	43	0
109	1	1.2	2	D	2	23	2	608	8	0	0
110	1	2.1	2	E	2	23	2	733	6	0	0
111	2	0.1	1	G	3		3	1140	13	0	0
112	2	0.3	2	B	3	31	2	649	5	0	0
113	2	0.2	2	G	3	52	2	79	21	7	0
114	1	3.4	2	G	2	12	1	470	4	0	0
115	1	0.3	1	E	3	29	2	704	21	2	0
116	1	0	1	G	3	64	2	419		0	0
117	2	0.8	1	G	2	42	2	484	9	0	0
118	2	0.1	2	B	3	32	1	760	6	0	0
119	2	0.2	1		2	20	3	892	6	3	0
120	1	1.6	1	D	2	16	3	922	16	10	0
121	2	2.9	1	E	2	14	2	560	8	0	0
122	1	0.4	2	B	2	29	2	931	21	13	0
123	2	1.2	1	F	2	9	2	960	18	7	0
124	2	1.3	1	D	2	21	3	1203	9	5	0
125	2	0.1	2	E	3	26	2	827	4	0	0
126	1	2	2		2	27	2	1014	11		0
127	1	3.1	1		2		2	40	9	0	0
128	1	0.1	1	B	3	43	2	962	4	0	0
129	1	0.3	2		2	35	3	855	16	7	0
130	1	0.1	2	G	3	15	2	977	9	0	0
131	2	0.3	2	G	3	31	3	678	6	2	0
132	1	1.1	2	B	2	23	2	713	4	0	0
133	1	0.7	2		2	31	1	812	21	9	0
134	1	0	1	D	3	54	1	311	14	7	0
135	1	0.6	1	G	2	22	1	1216	11	2	0
136	1	3.4	1	E	2	29	2	233	6	0	0

137	2	0.6	1	F	2	50	1	424	14	12	0
138	1	1.8	1	E	2	33	2	504	4	0	0
139	2	0.1	1	D	3	36	3	827	6	0	0
140	2	1.7	1	G	2	19	3	649	8	0	0
141	1	1	1	G	2	50	2	152	8	0	0
142	2	0.1	2		3		3	524	8	0	0
143	1	0.1	1	G	3	21	2	962	11	10	0
144	1	0.7	1	D	2	34	2	988	9	8	0
145	1	0.4	1		2	50	1	565	21	15	0
146	1	0.1	2		3	43	2	593	11	0	0
147	2	0.2	2	C	3	24	3	987	9	0	0
148	2	1.1	1		2	8	2	1014	4	0	0
149	1	1.1	2	D	2	37	1	605	5	0	0
150	2	1.8	1		2	12	2	1061	2	0	0
151	1	0.3	2	G	2	25	2	931	11	10	0
152	2	0.1	1	G	3	30	2	40	4	2	0
153	2	1.2	1	E	2	23	2	623	6	0	0
154	2	1.1	1	G	2	20	2	1173	11	3	0
155	2	0.2	1		3	25	3	339	6		0
156	1	3.4	2	G	2	15	2	631	6	2	0
157	2	0.4	1	F	2	36	3	684	4	0	0
158	1	0.4	1	C	2	50	1	462	11	0	0
159	1	0.7	2	C	2	57	2	314	6	0	0
160	1	0.1	1	D	2	39	3	827	9	0	0
161	2	0.2	2	G	3	47	1	521	16	4	0
162	2	0.2	2	G	3	16	2	841	11	0	0
163	1	1.7	2		2	8	2	704	19	7	0
164	1	2.1	2		2	18	2	649	4	0	0
165	2	0.1	2	G	3	17	3	462	4	0	0
166	1	0.3	2	E	2	30	2	593	11	0	0
167	1	1.2	2	G	2		2	1061	6	0	0
168	2	1.3	1	G	2	41	2	578	9	0	0
169	1	1.6	1		2	34	2	424	31	17	0
170	1	0.4	2	D	2	25	2	577	3	0	0
171	2	0.2	2	G	3	45	2	490	11	0	0
172	1	0.9	1	F	2	29	3	684	11	4	0
173	2	0.3	2	D	3	31	2	524	16	0	0
174	1	0.4	2	G	2	35	3	577	11	0	0

175	1	1.9	2		2	35	2	440	6	4	0
176	1	0	1	B	3	17	1	733	4		0
177	2	1.3	1	G	2	8	2	730	5	0	0
178	1	2.9	2	E	2	22	1	694	14	1	0
179	1	1.1	1	D	2	35	2	524	11	6	0

180	1	2.3	1	G	2	27	2	416	8	0	0
181	1	1.9	1	G	2	16	2	912	8	0	0
182	2	0	1	C	3	41	1	608	11	0	0
183	1	0.9	1	G	2	44	2	610	5	0	0
184	1	0.8	1	F	2		1	721	11	1	0
185	1	2.8	2	C	2	12	3	741	7	0	0
186	1	0.8	2	G	2	30	2	694	16	2	0
187	1	2.4	1	C	2	27	1	424	9	7	0
188	1	0.2	2	G	3	30	2	1173	7	3	0
189	1	0.6	2	C	2	39	2	721	11	9	0
190	1	0.6	1	G	2	32	2	771	8	7	0
191	1	0	2	B	3	21	3	757	16	11	0
192	1	1.5	1	C	2	24	3	1038	6	0	0
193	1	0.3	2	G	3	40	3	933	4	0	0
194	2	1.2	1	E	2	21	2	661	6	0	0
195	1	1.3	2	G	2		2	577	11	1	0
196	1	0.1	1		3	33	1	892	24	16	0
197	2	0	1	G	3	43	2	678	6	0	0
198	2	0.3	1	F	3	5	2	751	4	0	0
199	1	0.3	2		2	41	2	721	31	30	0
200	1	0.1	1	D	3	54	2	490	21	6	0
201	1	0	1	G	3	30	2	931	21	3	0
202	1	0.5	2	E	2	27	2	1203	21	16	0
203	2	0.1	2		3	21	3	987	4	3	0
204	1	4	1	E	2	18	2	132	9	1	0
205	1	0.8	1		2	36	1	542	9	0	0
206	1	0.5	2	G	2	27	3	694	11	0	0
207	1	0.8	2	B	2	35	2	649	11	4	0
208	1	1.9	2	D	2	10	2	722	18	13	0
209	1	0.3	1	F	3	55	2	132	11	7	0
210	1	1.1	1	F	2	38	3	542	11	0	0
211	1	2.1	2	F	2	30	1	560	6	0	0
212	1	0.1	2	D	3	20	2	1309	14	12	0

213	2	0.3	1	G	3	20	2	939	11	2	0
214	1	0.8	2	F	2		3	1288	31	13	0
215	1	0	1		3	46	2	520	6	0	0
216	1	1.1	1	G	2	20	1	795	9	0	0
217	1	0.6	1	C	2	59	2	565	6		0
218	1	0.6	2	E	2	20	2	684	6	5	0
219	2	1.3	1	D	2		3	1262	4	0	0
220	1	2.8	1		2	17	3	933	6	0	0
221	1	0.3	1	G	3		1	855	16	0	0
222	2	0.4	1	G	2	20	3	1216	11	0	0
223	2	1.6	1	C	2	43	2	314	5	0	0
224	2	0.8	1	G	2	25	2	922	4	0	0
225	1	1.5	2	G	2	19	1	827		0	0

226	1	0.2	2	E	3	46	2	577	6	0	0
227	1	2.8	1	B	2	19	2	578		0	0
228	1	0.2	2		2	39	2	796	9	0	0
229	1	1.6	2	D	2	10	1	922	11	0	0
230	1	0.3	2	F	3	39	2	1014	11	8	0
231	1	2.2	1	B	2		2	812	14	7	0
232	2	0.2	2	G	3	50	2	841	9	8	0
233	1	2.3	2	D	2	24	2	733	11	1	0
234	1	1.1	2	B	2		2	79	6	0	0
235	1	2.3	2		2	12	3	623	6	0	0
236	1	1.6	1		2	19	3	662	4	0	0
237	1	2.1	2	D	2	26	2	630	8	0	0
238	1	2.2	1	F	2	33	2	653		0	0
239	1	0.4	1	G	2	27	2	673	21	3	0
240	1	2.4	1	F	2	15	2	1038	14	4	0
241	1	1.2	1	C	2	30	2	1061	6	0	0
242	2	0.7	1	G	2	26	2	623	7	4	0
243	2	0.1	1	G	1	0	2	795	1	0	0
244	1	1.1	1	A	2	36	1	678	31	30	0
245	2	0	1	G	3		2	873	6	0	0
246	1	1.3	1	D	2	50	2	339	11	0	0

247	2	0.8	1		2	50	2	132	4	0	0
248	1	1.3	2	G	2	45	1	419	14	7	0
249	1	2.8	2	G	2		2	593	4	0	0
250	1	1.1	1	G	2	45	2	389	14	9	0
251	1	1.1	1	G	2	38	2	806	16	6	0
252	1	3.2	2	G	2	12	1	655	13	0	0
253	1	0.4	2		2	44	3	655	11	1	0
254	1	1.9	1	G	2	24	2	610	5	1	0
255	1	3.2	2	G	2		2	484	9	8	0
256	1	0	2		2	36	1	1173	13		0
257	2	0	1	G	2	31	1	550	9	0	0
258	2	0.1	2	D	3	34	1	855	16	0	0
259	1	0.8	1	G	2	32	2	630	11	1	0
260	1	2.2	2	F	2	27	3	440	16	14	0
261	1	0.5	1	G	2		2	684	11		0
262	1	2.1	2	G	2	15	2	560	4	0	0
263	1	2.6	1		2	13	2	827	4	0	0
264	2	0	1		3	23	2	778	16	12	0
265	2	0.2	1	G	3		2	694	26	21	0
266	1	0.2	1	G	3	41	2	524	14	9	0
267	1	0.8	2		2	16	2	623	6	0	0
268	1	1.8	2	D	2	27	2	678	21	4	0
269	2	0.8	1	B	2	31	2	842	11	3	0
270	1	0.3	2		2	34	2	623	4	0	0
271	2	1.6	1	D	2	18	3	931	14	4	0

272	1	0.1	1	G	2	25	2	1014	9	0	0
273	1	1.3	2	D	2	42	3	520	8	5	0
274	1	3.8	2	D	2	21	2	694		7	0
275	2	1.8	1		2	30	3	521	14	7	0
276	1	2.6	2		2	23	1	458	21	16	0
277	1	0	2	G	3	37	2	1103	2	0	0
278	1	0.3	1		2	22	2	1310	9	4	0
279	1	0.9	2	E	2		1	678	4	0	0
280	1	0.2	1	G	2	46	2	392	11	7	0

281	1	0.5	1	G	2	28	2	868		10	0
282	2	0.1	2	E	3	37	2	440	11	9	0
283	1	2.5	1		2	29	2	519	9	7	0
284	1	1.2	1		2	48	3	366	11	1	0
285	1	1.2	2	E	2	22	2	1082	7	0	0
286	1	0.9	1	F	2	42	2	366	18	9	0
287	1	0.1	2	G	3	37	3	902	14	1	0
288	2	0	2		3	17	2	1173	2	0	0
289	1	0.6	2		2	49	2	220		2	0
290	2	0.3	1	G	3	30	2	1229	31	16	0
291	1	0.8	2	E	2	7	2	542	6	0	0
292	2	0.1	1	G	3	36	1	470	6	0	0
293	1	2.7	2	E	2	22	2	462	6	0	0
294	1	1.6	1	D	2	30	3	653	4	0	0
295	1	0	1		3	55	2	366	21	9	0
296	2	0	1	A	3	38	2	733	8	0	0
297	1	0.2	2	E	2	40	2	440	11	0	0
298	1	2.7	2	G	2	21	2	700	31	25	0
299	2	2.8	1		2	5	1	542	5	2	0
300	2	0.2	1	A	3	59	2	145	5	3	0
301	2	1.6	1	G	2		2	623	14	0	0
302	1	2.5	2		2		2	352	5	0	0
303	1	1.3	2	E	2	14	2	757	6	0	0
304	1	0.3	2	G	2	40	2	458	8	6	0
305	1	0.3	1	G	2	34	1	713	16	6	0
306	1	0.3	2		3	46	1	694	16	0	0
307	1	0.6	2	D	2	14	3	730	6	0	0
308	1	0.2	2		3	22	2	988	16	10	0
309	2	0.2	2	B	3	49	2	440	11	0	0
310	1	1.2	2	G	2	22	2	1229	14	3	0
311	2	0.3	1	B	2	41	2	904	4	0	0
312	1	0.2	1	D	3	8	1	934	6	0	0
313	1	1.8	2		2	22	2	812	16	12	0
314	1	1.4	1	F	2	29	3	577	6	0	0
315	1	0.6	1	G	2	41	2	1061	6	0	0
316	2	0.2	2		3	15	2	1310	10	0	0

317	1	1.9	1	C	2	23	2	542	3	0	0

318	1	0	1	E	3	44	2	962	21	12	0
319	1	1.1	1		2	20	3	827	6	0	0
320	1	1.4	2	D	2	18	2	1014	8	0	0
321	1	1	2	B	2	10	1	1252	4	0	0
322	1	3.8	1		2	11	2	416	8	0	0
323	2	1.6	1		2		2	519	6	0	0
324	1	0.2	1		3	39	2	673	16	0	0
325	2	0.4	1	G	2	17	2	768		2	0
326	2	0.2	1	G	3		1	722		0	0
327	1	0.4	1	G	2	34	2	636	9	8	0
328	2	0	1		3		2	902	7	0	0
329	1	0.7	2	E	2	38	1	524	4	2	0
330	1	0.9	1		2	34	2	812	6	0	0
331	1	2.5	2	E	2	28	2	389	6	1	0
332	1	2.3	2	E	2	30	1	366	6	3	0
333	1	0.3	2	B	2	28	3	1038	4	0	0
334	1	2.5	1	C	2	20	3	542	9	4	0
335	1	1.1	1	G	2	25	1	827	11	0	0
336	1	1.7	2	G	2	19	2	796	11	0	0
337	1	1.9	2	B	2	23	2	873	6	0	0
338	2	0.1	1		1		2	253	1	0	0
339	2	0.5	1	G	2	41	2	806	14	1	0
340	1	2.6	1	C	2	19	2	840	4	0	0
341	2	3	1	D	2		1	962	4	0	0
342	1	1.5	1	D	2	20	1	608	7	4	0
343	2	0.4	1		2	41	1	560	3	0	0
344	1	0	2	F	3		2	1252	19	12	0
345	1	0.9	2	G	2	41	2	504	21	5	0
346	1	0.3	2		3	37	2	873	6	0	0
347	1	1.2	2	G	2		1	880	12	9	0
348	2	0.1	1	B	2	50	2	518	11	1	0
349	1	0.7	1	C	2		3	873	11	5	0
350	1	3.7	2		2	14	3	550	9	0	0

351	2	1.3	1	B	2	18	2	1173	3	0	0
352	1	0	2	D	3	34	3	1157	21	8	0
353	2	2.4	1	G	2	20	2	630	4	1	0
354	1	1.7	1		2	36	2	366	31	28	0
355	2	0.3	1	G	2	14	2	694	2	0	0
356	1	1.4	2	A	2	22	3	662	14	0	0
357	1	3.2	1	E	2	18	3	655	4	0	0
358	1	1	2	C	2		1	623	11	6	0
359	2	0.2	1	E	2	41	2	700	7		0
360	2	0.4	1	G	2	34	2	842	4	0	0
361	1	0.1	1	E	3	38	2	424	11	0	0
362	1	1.6	1	G	2	24	3	741	41	40	0
363	1	2.2	1	E	2	13	2	251	6	4	0

364	1	2.7	1		2	26	3	610	14	5	0
365	1	2.2	1	D	2	21	2	1038	24	18	0
366	1	3.1	1	B	2	9	2	542	16	4	0
367	1	2.4	2	D	2	17	2	1014	6	0	0
368	1	0.3	1	C	2	38	2	771	19	3	0
369	1	2	2	G	2	12	2	934	6		0
370	2	0	2	G	3	31	3	542	14	9	0
371	1	4.3	2	G	2	16	2	352	8	4	0
372	1	0.6	2	E	2	24	1	1189	11	1	0
373	1	1.3	2	G	2	17	3	1262	11	0	0
374	1	1.1	2		2	30	2	806	14	13	0
375	1	0.8	2	F	2	27	2	812	14	4	0
376	1	3.6	1	G	2	24	3	610	11	4	0
377	2	1	1	G	2	21	3	542	2	0	0
378	2	0	2	D	3	32	1	934	4	0	0
379	2	0.4	1	G	2	23	2	842	8	0	0
380	1	1.3	2	G	2	38	2	653	11	8	0
381	1	1.2	1	E	2	12	2	757	9	1	0
382	1	2	1	C	2		2	841	11	1	0
383	2	0.7	1	G	2	20	2	1173	6	0	0
384	1	0.3	1	G	3	62	2	132	2	0	0
385	1	1.1	2		2	37	2	827	16	15	0
386	2	2	1	B	2	14	1	704	6	3	0
387	2	0.4	1		2	38	3	605	6	0	0

388	2	0.3	2	G	3	30	2	366	8	6	0
389	2	1.3	1	G	2		1	904	16	0	0
390	1	1.5	1	G	2	46	2	392	11	0	0
391	2	0.6	1	E	2	14	1	751	6	0	0
392	1	0.2	1	G	2	33	2	1189	14	0	0
393	1	0.4	2		2	37	2	934	16	6	0
394	2	0	2	G	3		1	1140	21	16	0
395	2	1.3	1	E	2	10	3	1229	6	2	0
396	1	0.9	2	G	2	33	2	771	5	0	0
397	1	1.4	2	G	2		2	392	9	0	0
398	1	2.3	1		2	12	3	868	9	5	0
399	1	0.2	2		2	43	2	934	6	0	0
400	1	2.1	2	E	2	18	2	673	11	9	0
401	1	3	2	A	2	30	1	311	11	2	0
402	2	0.2	1		3	34	2	649	18	16	0
403	1	1.8	2	D	2	24	2	1157	11	0	0
404	1	0.3	1	G	3	38	1	608	11	9	0
405	1	0.5	2		2	32	2	760	14	0	0
406	1	2.6	1	G	2	29	1	578	11	2	0
407	2	0.1	2		3	27	2	1061	4	0	0
408	1	0.8	2		2	18	2	751	6	0	0
409	2	0	2		3	37	1	962	8	5	0

410	1	2	2	D	2	24	2	636	8	3	0
411	1	1.9	1	G	2	29	1	934	18	13	0
412	1	1.2	2		2	23	1	593	14	0	0
413	1	2.4	2		2	12	1	741	7	0	0
414	1	1.6	1	D	2	28	1	771	3	0	0
415	2	2.4	1	F	2	16	2	741	4	0	0
416	1	1.3	2	E	2		2	934	11	6	0
417	1	0.7	1	C	2	24	1	902		4	0
418	1	0.4	1	G	2	46	2	655	8	0	0
419	1	3.7	2	G	2	16	2	352	21	19	0
420	2	0.2	2	G	3		2	855		0	0

421	1	1.5	2	D	2		2	880	19	0	0
422	1	0.5	2	B	2	34	2	649	9	8	0
423	1	0.3	2		2	30	1	1173	31	15	0
424	1	2	2		2	34	2	694	9	0	0
425	1	0.2	1	D	3	35	2	1203	11	1	0
426	1	0.1	2		3		1	1316	26	7	0
427	2	0.8	1	E	2	30	2	524	6	2	0
428	1	1.4	1	G	2	23	1	855	18	11	0
429	1	3	2	D	2	11	1	593	19	11	0
430	1	0.1	2	E	3	17	1	892	4	0	0
431	1	1.3	2	A	2	25	2	1038	11	9	0
432	2	0.2	2	B	3	21	3	966	14	4	0
433	1	0.3	1	F	3	45	2	840	14	6	0
434	1	0.3	2		2	25	2	922	19	17	0
435	1	2.9	2	G	2	10	3	99	11	4	0
436	2	0	1	G	3	27	1	741	9	0	0
437	1	0.3	2	E	3	54	2	462	9	0	0
438	1	1.6	1	G	2	25	2	578	21	8	0
439	1	0.9	1	B	2	30	3	778	16	0	0
440	1	0.5	1	E	2	18	2	1280	4	0	0
441	1	3.4	1	D	2	8	1	484	4	0	0
442	2	2.3	1	F	2	16	2	796	6	4	0
443	1	1.8	1	G	2		3	1014	4	3	0
444	2	0.8	1		2	12	2	972	14	6	0
445	1	0.3	1	G	3	42	2	988	21	20	0
446	2	0.2	1		3	38	3	741	9	0	0
447	1	0.8	1	F	2	43	3	610	9	0	0
448	1	1.5	2	D	2	20	2	855	11	8	0
449	1	0.6	1	B	2	26	2	704	9	0	0
450	1	3	1	E	2	23	1	311	8	0	0
451	2	0	2	C	3		2	605	6	0	0
452	2	0.2	1	E	3	43	3	79	9	0	0
453	2	0.1	1	G	3	12	2	855	4	0	0
454	1	0	1	G	3	36	1	796	2	0	0
455	2	0.4	1	F	2	34	2	700	4	0	0

456	1	2.1	2	G	2	15	1	855	11	0	0
457	2	0.1	1	C	3		2	954	13	9	0

458	2	0	1	D	3	25	1	251	9	0	0
459	1	1	1	G	2	42	2	470	11	0	0
460	1	3.1	2	G	2	26	2	519	6	0	0
461	1	1.9	2	G	2	16	2	1038	7	0	0
462	2	0.3	1	G	2	35	1	721	11	4	0
463	2	0.9	1	D	2	33	2	1140	9	0	0
464	2	0.1	1	G	2	42	2	988	9	0	0
465	1	1.4	1		2	41	2	366	14	2	0
466	1	1.5	1	D	2	12	2	922	6	0	0
467	2	0.3	1	G	2	31	3	1157	4	2	0
468	1	3.6	2	D	2	9	3	484	8	5	0
469	1	3.4	1	E	2		2	416	14	0	0
470	2	0.9	1		2	36	2	416	9	8	0
471	2	0	1	G	3		1	704	16	8	0
472	1	0.2	1	D	3		3	873		8	0
473	1	2.1	2	F	2	21	2	605	2	0	0
474	1	1.4	1	G	2	20	2	922	14	8	0
475	2	3.1	1	G	2		2	424	6	0	0
476	1	0.4	2	G	2	30	2	649	19	10	0
477	1	0.8	1	G	2	36	3	873	21	20	0
478	2	0	1	G	3	40	1	419	3	0	0
479	2	0	2	G	3	56	2	366	11	0	0
480	1	0.1	1	F	3	47	2	694	9		0
481	1	1.9	1		2		2	678	9	4	0
482	1	1.5	2	G	2	35	2	771	16	15	0
483	1	0.2	2	G	2	30	2	1203	9	6	0
484	1	0.9	1	E	2	35	2	392	2	0	0
485	1	0.1	2	B	3	45	1	578	8	0	0
486	1	2.9	2	G	2	10	1	593	16	7	0
487	1	0.5	2	E	2		2	424	8	0	0
488	1	1.2	2	E	2	31	3	721	4	0	0
489	1	2.3	2	G	2		2	649	31	22	0
490	1	2.4	1	E	2	25	2	440	4	0	0
491	1	0.4	1		2	28	3	1203	8	3	0
492	1	0.2	2	F	2	41	2	655	6	0	0
493	1	0.1	2		2	20	3	730	8	0	0

494	1	0.4	2	G	2	20	2	542	6		0
495	2	0.3	1		3	22	3	1038	6	1	0
496	1	0.6	2		2	32	2	704	21	11	0
497	1	1.1	2		2	37	2	733	5	3	0
498	1	0.4	2	G	2	9	2	768	9	0	0
499	1	1.4	1	E	2	25	3	636	8	0	0
500	1	1.2	2	G	2	26	2	661	4	0	0
501	2	0	2	G	3	11	2	1189	6	0	0

502	1	1.6	1	C	2	31	3	1038	11	2	0
503	2	1.9	1	G	2		1	1014	6	0	0
504	1	0.1	2	G	3	45	3	504	7	0	0
505	2	0.1	1	G	2	43	2	484	11	7	0
506	1	1.2	2	D	2	32	3	1061	26	10	0
507	2	0.2	1	G	2	27	3	608	4	0	0
508	2	1.3	1	G	2	20	2	868	4	0	0
509	1	0.6	1		2	34	2	1122	31	14	0
510	1	0.2	2	B	2	24	2	722	4	0	0
511	1	0	1	E	3	36	3	827	9	0	0
512	1	1.5	2	E	2	32	2	610	6	0	0
513	2	2.5	1	E	2	10	2	827	11	0	0
514	1	0.1	2	G	3	49	3	630	14	0	0
515	1	2	1	E	2	22	2	577	14	0	0
516	2	0.2	1	G	2	25	3	1140	6	0	0
517	1	1.7	2	G	2	18	3	662	16	0	0
518	1	0	1	G	2	30	2	673	16	0	0
519	1	0.3	1		3	38	2	778	6	1	0
520	1	0.1	2		3	6	3	987	6	0	0
521	1	2.4	2	G	2	19	2	623	11	9	0
522	1	3.4	2	E	2	7	3	462	9	6	0
523	1	1.1	1	D	2	36	1	655	2	0	0
524	1	3.6	1	C	2	11	1	251	11	0	0
525	1	1.7	2	G	2	41	3	470	31	22	0
526	2	1	1	G	2		2	806	6	0	0
527	1	0.4	1	G	2	51	2	565	14	12	0
528	2	0	2	E	3	30	2	855	21	19	0
529	1	2.1	1		2	22	3	741	6	0	0
530	1	3.1	1	G	2	24	2	655	11	0	0

531	1	1.4	2	G	2	20	2	1103	16	15	0
532	1	0.6	1	E	2	31	2	662	11	0	0
533	1	0	1		3	40	2	623	9	2	0
534	1	2	1	C	2		1	662	11	5	0
535	1	2.2	1	A	2	48	2	99	21	10	0
536	1	1.4	2	G	2	16	2	880	11	10	0
537	2	0.1	1	E	3	21	2	694	19	17	0
538	2	0.2	2	C	3	27	2	608	14		0
539	1	1.8	1	G	2		2	550	9	0	0
540	1	0	1	B	3		1	1241	6	1	0
541	1	0.9	2	G	2	19	2	730	46	27	0
542	2	1.1	1		2	27	2	1060	11	6	0
543	2	1.9	1	F	2	7	2	778	14	13	0
544	1	3.2	2	D	2	7	3	655	11	0	0
545	1	1.1	2	F	2	31	2	842	31	17	0
546	1	1.6	2	E	2	17	3	721	8	0	0
547	1	0.9	2		2	20	2	1157	21	7	0

548	1	2.2	1	E	2	22	1	484	4	0	0
549	2	0.2	2		3	31	2	812	3	0	0
550	2	0.8	1	C	2	20	2	623	4	0	0
551	2	0.1	1	G	3	32	2	730	6	0	0
552	1	3.1	2	G	2	17	2	504	4	0	0
553	2	0.3	2	C	3	39	2	678	24	6	0
554	2	1.6	1		2		3	827	5	0	0
555	1	1.5	1	A	2	12	2	339	6	4	0
556	1	1.6	2	G	2	14	2	880	18	6	0
557	1	1.1	2	D	2	38	2	440	8	0	0
558	1	3.2	2	B	2	8	2	605	9	0	0
559	2	0.2	1	G	3	53	2	694	2	0	0
560	1	2.4	2	G	2	16	1	1082	11	7	0
561	1	0.8	1	C	2	33	2	1038	9	5	0
562	1	0.2	1	G	3	36	2	855	6	0	0
563	2	0	2		3		2	1061	21	15	0
564	1	1.1	2		2	25	2	662		10	0
565	1	0.2	2	E	3	49	2	578	11	0	0
566	1	2.2	1	D	2	27	2	484	16	9	0

567	1	2.5	1	C	2	15	2	700	6	3	0
568	1	1.3	1	E	2	45	2	470	9	5	0
569	1	0.9	2	G	2	19	3	1296	21	18	0
570	1	0.1	2	D	3	27	2	704	10	3	0
571	1	1.4	2	D	2	31	1	623	21	6	0
572	1	0	1	E	2	50	2	440	26	10	0
573	1	3	1	G	2	9	2	678	4	0	0
574	1	2.7	2	F	2	7	2	1014	4	0	0
575	1	2.2	1	D	2	19	3	662	6	0	0
576	1	1.9	2		2	20	1	806	11	0	0
577	1	1.6	1		2	18	3	1122	18	6	0
578	1	0.2	1	D	3	51	2	578	6	0	0
579	1	0.2	1	G	2		2	757	14	6	0
580	2	0.9	1	D	2	40	2	542	26	17	0
581	2	0.1	1		2	25	2	1229	9	5	0
582	1	0.7	1		2	19	2	190	8	0	0
583	2	0.9	1	G	2	40	2	352	6	0	0
584	1	0.8	1		2		2	132	6	0	0
585	1	0.1	1	E	3	57	2	519	14	6	0
586	2	0.1	1	G	3		3	254	8	0	0
587	1	0.1	2		3	22	2	694	31	17	0
588	1	2.1	2	E	2	35	2	419	9	3	0
589	1	0.6	1	C	2	20	1	1173	14	1	0
590	1	0.6	2		2		1	922	14	0	0
591	1	2.3	1	E	2	15	3	1061	11	10	0
592	1	0.2	2	G	2	42	2	988	36	31	0
593	1	0.8	2	F	2	33	2	713	4	3	0

594	1	1.3	2	G	2	35	2	700	7	1	0
595	1	3.4	2	C	2	15	2	311	21	14	0
596	1	2.3	1	E	2	33	2	610	10	0	0
597	1	2.1	1		2	16	2	988	31	19	0
598	1	1	2		2	54	2	366	4	0	0
599	1	0.2	1	G	2	31	3	1216	4	0	0
600	1	2.1	1	F	2	14	2	1122	16	13	0
601	1	3	2	D	2	15	1	311	8	6	0
602	2	0.1	2	G	3	40	2	440	4	0	0

603	1	0.3	2	G	2	40	2	1103	4	0	0
604	1	0.7	1	G	2	42	2	524	11	0	0
605	2	0.2	1		3	27	2	608	4	0	0
606	2	0.3	2	F	3	33	2	722	3	0	0
607	2	0.3	1	G	3	19	2	1262	3	0	0
608	2	1.6	1		2	22	3	673	6	2	0
609	1	0.3	1	G	3	43	2	873	6	0	0
610	2	0.2	1	G	3	21	1	741	14	0	0
611	2	2	1	D	2	15	1	440	4		0
612	2	0	1		3	22	2	855	4	0	0
613	2	0.5	1		2	16	3	1271	11	0	0
614	1	1.7	1		2	36	2	440	11	0	1
615	1	0.3	1	D	3	56	2	251	16	0	1
616	2	0.2	1	G	3	33	3	868	24	11	1
617	2	0.3	1	E	3	57	3	518	14	12	1
618	1	0.1	2	G	3	41	3	577	26	7	1
619	1	0	1	G	3	58	2	653	8	0	1
620	1	1	2	G	2	35	3	608	14	8	1
621	1	0.2	1	C	3	20	1	796	16	0	1
622	1	0.3	2	G	3	46	3	550	2	0	1
623	2	0.1	1	E	3		2	760	14	13	1
624	1	0.2	2	G	3		2	605	11	7	1
625	1	1.9	2	E	2	32	1	610	11	6	1
626	1	0.3	1	E	3	36	2	684	14	11	1
627	2	0	1	G	3	32	3	812	6	0	1
628	2	0	2	E	3	52	2	389	5	0	1
629	2	0.2	1	G	3	44	3	275	6	2	1
630	1	0	2	G	3	41	2	827		15	1
631	1	1.2	1	G	2	32	2	462	11	0	1
632	1	0.1	1	G	3	37	1	565	8	3	1
633	1	0.3	1		3	60	1	311	14	10	1
634	1	1.6	1		2	30	2	152	21	7	1
635	1	2.3	2	G	2		2	462	6	0	1
636	1	0.1	1	D	3	42	3	988	14	7	1
637	1	0.2	1	C	2	56	3	275	11	0	1
638	2	0	2	F	3		1	649	7		1

639	2	0.1	2	G	3	34	2	1140			1

640	1	0	1		3	34	2	1102	14	0	1
641	2	0.3	1	E	3	15	3	560	6	0	1
642	2	0	1	G	3	41	2	577	11	6	1
643	1	0.3	1	C	3	45	2	314	4	0	1
644	1	0.3	1	E	3	33	2	704	11	0	1
645	1	0	1		3	48	1	605	11	0	1
646	1	1.5	1	G	2	40	2	339	7	0	1
647	1	0.3	1	B	3	54	1	339	11	4	1
648	2	0	1	C	3	46	2	392	6	0	1
649	1	0.2	1	G	2	55	3	610	16	2	1
650	1	0.2	1	E	3		1	649	11	0	1
651	1	0.2	1	D	3	25	2	842	6	0	1
652	1	0.3	2	E	3	47	1	630	6	2	1
653	1	0.4	1	B	2	40	2	841	21	17	1
654	1	1.1	2	C	2	48	2	366	11	10	1
655	2	0.3	1	G	3	45	3	187	8	0	1
656	1	0	1		3	42	2	962	21	16	1
657	1	2	2	E	2	15	2	1102	14	1	1
658	2	0.2	1	D	3	26	2	550	6	0	1
659	2	0.3	2	E	3	43	2	542	14	11	1
660	1	0.3	2	G	3	28	2	912	11	7	1
661	1	0.3	2	E	3	52	2	392	11	10	1
662	2	0.1	2	G	3	35	3	1157	11	0	1
663	1	0.3	1		3	42	1	392	6	1	1
664	2	0	2	A	3		2	1061	4	0	0
665	2	0.3	2		3		3	311	6	4	0
666	1	1.3	1	E	2	18	2	1216	4	0	0
667	2	0	2	E	3	48	1	806	8	7	0
668	2	0.1	1	F	3	50	2	419	8	2	0
669	2	0.3	1	D	3	25	2	1157	11	0	0
670	1	0.8	1	E	2	41	3	339	11		0
671	1	0	1	G	3	60	3	251	14	5	1
672	1	1.4	2	F	2	45	2	220	6	2	1
673	1	1	1	A	2	38	2	366	21	14	1
674	1	1.5	1	E	2	41	2	311	6	0	1
675	2	0.2	1	G	3		2	655	11	0	1

676	1	0.5	2		2		2	560	18	16	1
677	2	0.2	2	F	3		2	550	8	6	1
678	2	0.3	1	E	3	45	2	655	6	0	1
679	2	0	2	E	3	38	1	904	6	4	1
680	1	0.3	1	D	3	50	1	524	19	1	1
681	2	0.1	2	E	1	0	2	458	1	0	1
682	2	0	2	C	3	35	2	593	8	0	1
683	2	0.1	2		3	10	2	524	8	0	1
684	1	0.2	1		3	42	1	560	6		1
685	1	0.1	1	B	3	42	3	988	6	0	1

686	1	0.3	2	C	3	32	3	855	11	0	1
687	1	0	1	B	3	44	2	700	26	14	1
688	1	0.2	1	D	1	0	2	560	1	0	1
689	1	1	1		2	40	2	484	14	3	1
690	1	0	1	C	1	0	3	578	1	0	1
691	1	0.3	1	E	3	45	3	470	6	0	1
692	1	0.3	1	F	1	0	2	458	1	0	1
693	1	0.7	1	D	2	53	2	314	14	11	1
694	1	0.2	2		3	44	2	873	6	1	1
695	1	0.1	1		3	36	2	673	11	6	1
696	1	0.2	1		3	49	2	366	11	0	1
697	1	0.7	1	G	2	32	2	694	26	20	1
698	1	0.2	2	E	3	37	1	661	9	0	1
699	1	0.1	1	G	3	55	2	519	9	0	1
700	2	0	1	G	3	41	1	520	5	0	1
701	2	0.1	2	E	3	42	2	577	11	0	1
702	1	0.2	2	B	3	27	3	608	16	0	1
703	1	0.2	1	G	3		2	806	14	0	1
704	1	0.1	1		3	48	2	694	14	6	1
705	1	0	1	E	3	63	3	132	11	2	1
706	1	0.9	2	G	2	38	1	610	9	0	1
707	1	0.1	1		3		2	796	9	7	1
708	1	0	1	G	3	49	2	440	11	10	1
709	1	0.3	1	G	3	35	2	694	14	11	1
710	1	1.2	1	G	2	30	2	1122		28	1
711	2	0	2	G	3	36	3	636	11	9	1

712	1	0.2	2	B	3	36	1	684	6	0	1
713	1	0	1		3		3	339	31	19	1
714	2	0	1	E	3	23	1	1241	5	0	1
715	1	0.2	2	D	3	51	1	458	8	5	1
716	2	0.1	2		3	50	1	220	6	0	1
717	2	0.1	1	C	1	0	1	1038	1	0	1
718	2	0	1	D	1	0	2	649	1	0	1
719	1	0.2	2	G	3	54	2	653	21	12	1
720	2	0.2	2	G	3	41	2	934	11	7	1
721	1	1.7	2	E	2	51	2	253	21	20	1
722	1	0.3	1	C	3	47	2	416	5	0	1
723	1	0.3	2	G	3		3	623	11	5	1
724	1	0	1	D	3		2	760	11	1	1
725	1	0.5	1	F	2	44	2	841		14	1
726	1	0.1	1	A	3	57	2	251	11	10	1
727	1	0.3	2	B	3	52	1	490	16	14	1
728	1	0.2	1	E	3	40	1	778	8	0	1
729	1	0.3	1	E	3	50	1	194	5	0	1
730	1	0.2	1	G	3	25	2	275	3	0	1
731	1	0	2		3	29	2	730	11	8	1

732	2	0	2	G	3	45	3	655	9		1
733	1	0.1	2	E	3	40	2	700	8	7	1
734	2	0.1	1	G	3	36	2	1173	11	9	1
735	2	0.3	2	E	3	30	1	988	9	6	1
736	1	0.6	2	G	2	30	1	1122	21	6	1
737	2	0	2	D	3	42	1	873	14	2	1
738	2	0	2	D	3	43	1	721	9	0	1
739	1	1	1	G	2		2	1189	6	0	1
740	1	0.6	2	D	2	33	2	700	41	25	1
741	1	0	2	G	3	36	2	608	14	0	1
742	1	3.2	2	E	2	30	2	40	11	0	1
743	2	0.1	2	E	3	37	2	608	4	0	1
744	1	0.3	1	F	3	47	2	655	14	2	1
745	1	1.4	1		2	28	2	636	14	7	1
746	1	0.3	1		2	49	2	416	4	0	1
747	1	1.3	2	F	2	47	2	470		0	1

748	1	0.1	2	D	3	30	1	721	14	3	1
749	2	0.1	2	G	3	32	2	1189	9	5	1
750	1	2.1	2	E	2	23	3	542	21	10	1
751	2	0.1	1		3	48	3	542	7	0	1
752	1	0.3	1	G	3	40	2	962	14	0	1
753	1	0.2	1	D	3		2	988	9	0	1
754	2	0.2	2	E	3	36	1	440	11	3	1
755	2	0	1	C	3	51	2	470	6	0	1
756	1	0.2	2		3		2	419	11	0	1
757	2	0.2	2	G	3	30	2	933	4	0	1
758	1	1.6	2	E	2	19	3	661	8	0	1
759	1	1.3	1	G	2	46	2	251	6	0	1
760	2	0.2	2		1	0	1	631	1	0	1
761	2	0.3	1	G	3	38	2	806	9	2	1
762	1	3.6	2	G	2	14	2	458	6	3	1
763	1	1.4	2	G	2	40	2	282	21	5	1
764	1	0.3	1	G	3	31	2	1262	24		1
765	1	0	1	D	3	35	2	1103	6	2	1
766	1	0	1		3	57	3	187	6	0	1
767	1	0.2	2	G	3	25	3	954	11	0	1
768	2	0	1	D	1	0	2	458	1	0	1
769	2	0	1		3		2	649	11	2	1
770	1	1.5	1	G	2	33	2	440	19	6	0
771	1	2.8	1	E	2	35	2	40	4	0	0
772	1	0	1	C	3	53	1	353	6	0	0
773	1	3.1	1		2	11	1	904	11	0	0
774	1	2	2	E	2	36	2	282	6	0	1
775	1	0.2	1	C	3	50	3	578		5	1
776	1	0.2	1	G	3	40	2	649	9	0	1
777	1	0.1	2	G	3	40	2	519	3	0	1

778	1	1.2	2	G	2	40	2	416	11	9	1
779	1	4.1	2	C	2	15	2	610	14	0	1
780	1	2.1	2	G	2	20	3	760	11	0	1
781	1	0.5	2		2	50	2	440	8	0	1
782	1	2.1	2	G	2	18	3	577	11	0	1
783	1	4	2	E	2	13	3	311	4	0	1

784	1	0.4	2	E	2		1	458	9		1
785	2	0	1	D	3	55	1	186	2	0	1
786	1	0.2	1		3	53	3	605	14	1	1
787	2	0.1	2	B	3	18	1	392	6	5	1
788	1	0.7	2	D	2	32	3	1060	14	11	1
789	1	0.1	1	G	3	40	2	880	21	15	1
790	1	0.2	1	C	3	34	2	721	14	0	1
791	1	1.6	1	E	2	30	3	79	11		1
792	2	0	1	G	3		2	868	21	6	1
793	2	0.2	2	E	3	20	1	773	3	0	1
794	2	0.3	2	E	1	0	2	251	1	0	1
795	2	0.1	2	E	1	0	2	190	1	0	1
796	1	2.5	1	F	2	35	2	653	16		1
797	1	0.3	2		3	51	3	366	16	0	1
798	1	0.9	1	G	2	25	2	684	14	2	1
799	1	0	1	D	3	46	3	519	4	0	1
800	2	1.5	1		2	29	2	678	9	0	1
801	1	1.7	1	D	2	38	3	771	11	5	1
802	1	1.3	1	E	2	48	1	389	11	1	1
803	1	1.7	2	G	2	30	2	578	11	4	1
804	2	1.2	1	C	2	15	3	1203	6	0	1
805	1	0	2	E	3	40	2	933	11	7	1
806	1	2.2	2		2	34	2	314	9	3	1
807	1	0.2	1		3	34	3	577	4	2	1
808	2	1.8	1	C	2	30	2	519	3	0	1
809	1	0.1	2		3	55	3	470	7	1	1
810	1	0	2		3	40	1	1014	14	0	1
811	1	2.4	2	G	2	40	2	253	14	7	1
812	2	0	2	E	3	30	2	462	11	0	1
813	1	1.9	2	D	2	30	1	504	11		1
814	2	2.7	1	E	2	5	2	40	2	0	1
815	1	0	1		3		1	578	8		1
816	1	0.2	2		3		1	812	19	2	1
817	1	1.5	2	F	2	38	1	519	11	7	1
818	2	1	1	D	2	24	1	1103	14	0	1
819	2	0.3	1	G	3	34	3	673	11	0	1

820	1	0.3	2	D	3	44	1	741	6	0	1
821	1	1.6	2	E	2	20	2	733		0	1
822	1	0.3	1	F	2	30	1	713	14	0	1
823	2	0.2	1		3	37	1	988	8	0	1

824	1	0.7	1	D	2	42	2	542	11	6	1
825	1	1.3	2	F	2	43	1	462	6	0	1
826	1	0.2	2	C	3	42	2	760	11	2	1
827	1	0.5	2	F	2	29	2	704	14	12	1
828	1	0.3	2	D	3	52	2	484	9	1	1
829	2	0.4	1	G	2	45	2	419	2		1
830	1	0	2	G	1	0	2	733	1	0	1
831	1	0	2	G	3	25	1	1140	7	3	1
832	2	0.3	2	B	3	35	3	855	6	0	1
833	1	0.6	2	D	2	36	1	542	11	0	1
834	1	0.2	2	G	2	50	3	518	6	4	1
835	1	3	1	D	2		2	630	6	0	1
836	1	1.2	2	D	2		1	311	9	0	1
837	1	0.2	1	E	3	34	3	704	9	4	1
838	1	0	2	F	3	60	2	389	21	14	1
839	1	0.1	1	D	2	40	3	700	11	0	1
840	1	0	2	G	3	45	3	524	9	0	1
841	1	0.3	1		3	55	1	565	8	0	1
842	2	0.1	1	G	3	49	3	50	8	0	1
843	1	1.3	1	E	2	40	2	653	21	10	1
844	1	0.1	1	E	3	50	2	578	9	0	1
845	1	0.3	2	G	3	44	2	771	21	16	1
846	1	0.2	1	G	3	22	1	892	2	1	1
847	1	0.2	2	E	3	26	2	661	8	0	1
848	1	0.3	1	B	3	56	2	519	6	5	1
849	1	0	1	E	3	50	2	770	11	0	1
850	2	0.3	1	E	3	41	2	608	11	8	1
851	1	2.3	1	F	2		1	366	4	0	1
852	1	0.1	2		3	56	2	187	6	4	1
853	1	0.8	2	B	2	26	2	1014	16	6	1
854	1	3	2	D	2	26	3	470	11	0	1
855	1	0.5	1		2	46	2	605	6	0	1
856	2	0	1		1	0	3	550	1	0	1

857	1	1.3	2	G	2	17	2	1173	21	10	1
858	1	0.6	1	G	2	62	2	67	2	0	1
859	1	0.1	2	B	3		2	1082	4	0	1
860	2	0	1	C	1	0	2	610	1	0	1
861	1	0	1	G	3	43	3	593	21	3	1
862	1	0.4	1	B	2	45	2	988	8	0	1
863	1	1.3	1	E	2		1	741	16	11	1
864	1	2.5	2	B	2	25	2	721	9	0	1
865	1	2.8	2	B	2	20	2	653	8	0	1
866	1	0.7	1	E	2		3	314	6	5	1
867	1	0.1	1	E	3	28	2	704	9	2	1
868	2	0	2	D	3		2	484	8	0	1
869	2	0.1	1	A	3	31	2	440	6	0	1

870	1	3.1	1	C	2	20	1	733	11	7	1
871	1	1.2	1		2	35	3	524	16	7	1
872	1	0.1	2		3	35	2	796	14	6	1
873	1	0.3	1	E	3	32	2	713	21	2	1
874	1	0.1	1	G	3	24	2	892	9	8	1
875	1	0	1	E	3	40	2	1038	11	0	1
876	1	1.9	1		2	29	1	440	4	0	1
877	2	0.7	1	G	2	40	2	678	16	10	1
878	1	0	1	G	3	44	2	721	6	0	1
879	2	0.1	1	C	3	47	2	806		0	1
880	1	0.9	1	E	2	50	3	352	6	0	1
881	1	0.2	2		3	47	3	462	26	23	1
882	1	0	1	E	3	50	3	458	4	0	1
883	1	0.5	2	G	2		2	608	6	0	1
884	1	0.3	2	G	3	42	2	904	11	0	1
885	1	1	1	D	2	40	3	550	14	8	1
886	2	0.3	2	E	3	50	2	190	6	0	1
887	2	0	1	G	3	45	2	678	9		1
888	1	0.1	2	D	3	50	2	440	11	0	1
889	2	0.3	1	G	3	33	2	1082	11	0	1
890	1	2.6	1	G	2		2	99	8	0	1
891	1	0	2	G	3	33	1	812	8		1

892	2	0	2	G	3	38	2	623	9	1	1
893	1	0.2	2	G	3	46	2	655	8	0	1
894	2	0	2		1	0	3	462	1	0	1
895	2	0.3	2	G	3	47	3	578	8	0	1
896	1	0.3	1	C	3	39	2	636	16	15	1
897	2	0.1	1	E	1		2	490	1	0	1
898	1	0.2	1	F	3	34	3	873	21	18	1
899	1	1.5	1	F	2	42	2	311	7	5	1
900	1	0.3	2	F	3	41	2	577	11	10	1
901	2	0.2	2	G	1	0	2	40	1	0	1
902	1	0.6	1	G	2	52	1	314	5	0	1
903	1	2	2	C	2	30	1	254	19	0	1
904	1	0.2	1	E	3	34	1	812		0	1
905	1	0.2	1	B	3	41	1	988	11	0	1
906	2	0.4	1	E	2	25	2	730	11	0	1
907	1	0	2	G	3	34	2	623	11	0	1
908	2	0.2	2	F	3	39	3	593	9	0	1
909	2	0.1	2	E	1	0	2	1122	1	0	1
910	1	2.4	2	E	2	24	2	504	9	0	1
911	2	0	1	G	3	27	2	1241	11	7	1
912	1	0.3	2	D	3	34	2	704	10	0	1
913	2	0	1	D	1	0	2	99	1	0	1
914	1	0.1	2	C	3	40	2	741	6	0	1
915	1	0.2	1		3	43	1	962	9	0	1

916	1	0.2	1	G	3	45	2	542	11	4	1
917	1	0	1	F	3	36	2	721	16	10	1
918	1	0	1	F	3	67	1	253	8	0	1
919	2	0.1	2	G	3		3	1014	9	8	1
920	1	1.2	1	A	2	35	2	550	11	9	1
921	1	0.5	2		2	50	2	145	7	0	1
922	1	1.6	2	C	2	30	1	99	9	8	1
923	1	0.2	2	A	3		3	251	6	5	1
924	1	1.9	1	G	2		2	504	8	3	1
925	2	0.1	2	C	1	0	2	939	1	0	1
926	1	1.2	2	G	2	45	2	484	16	1	1
927	1	0.2	2		1	0	2	631	1	0	1
928	1	3.2	1	A	2	35	3	419	9	3	1

929	1	0.3	2	G	1		2	962	1	0	1
930	1	0.1	1	C	3	53	2	550	14	3	1
931	2	0.3	1	G	3	57	1	314	21	10	1
932	2	0.2	1	E	3	33	2	1102	6	2	1
933	1	0.1	1	G	3	42	3	806	6	0	1
934	1	0	2	D	3	59	2	190	8	0	1
935	2	0.3	1	G	3	49	2	490	8	4	1
936	1	2.4	2	D	2		1	152	16	2	1
937	2	0.2	1		3		1	912		0	1
938	1	1.2	1	E	2	30	3	524	8	0	1
939	2	0.1	2	D	3	30	3	713	14	0	1
940	1	0.1	1	E	3	53	1	524	8	0	1
941	2	0.2	2	E	3	28	1	662	6	0	1
942	2	0.3	2	E	3	35	3	560	9	0	1
943	1	0	1	F	1	0	3	1203	1	0	1
944	2	0.2	1	G	3	41	2	904	6	0	1
945	1	0.5	1	G	2	50	1	366	8	0	1
946	2	0.1	2		1	0	2	484	1	0	1
947	1	0.6	1		2	41	2	678	11	0	1
948	1	3.3	1	G	2	25	2	339	11	9	1
949	1	0.2	2	D	2	48	1	311	14	0	1
950	2	1.6	1	C	2	25	2	771	8	0	1
951	1	0.3	2		3	37	2	721	14	6	1
952	1	1.8	1	G	2	36	2	416	16	0	1
953	2	0	2	F	1	0	3	741	1	0	1
954	2	0.1	1	E	3	32	2	812		0	1
955	2	0.1	2	G	3	20	2	490	4	0	1
956	1	0.2	2	B	3	57	1	565	11	2	1
957	2	0.3	2	E	3	30	2	730	11	4	1
958	1	2.6	2	G	2	17	2	484	2	0	1
959	2	0	2		3	29	2	713	6	3	1
960	2	0.3	2		3	24	1	741	11	0	1
961	1	0.9	1	F	2	40	1	253	6	0	1

962	1	0.8	1	E	2	29	2	608	5	0	1
963	1	0.2	1	E	3	50	2	988	8	0	1
964	1	0.3	1	G	2	42	3	1014	9	0	1
965	1	0.2	1		3	36	2	760	6	0	1

966	2	0.7	1	B	2	46	1	653	8	0	1
967	2	0.2	1	G	3	42	3	988	8	3	1
968	1	0.2	2	G	3	25	1	678	18	3	1
969	2	0	2	G	3	35	3	416	4	0	1
970	1	1.8	2		2	32	3	653	7	0	1
971	1	0.2	1	D	3	35	2	694	6	0	1
972	1	1.5	1		2	29	2	578	41	30	1
973	1	2.3	1	G	2	29	2	694	6	0	1
974	1	0	1	G	3	51	3	366	6	0	1
975	2	0.1	2	C	3	28	2	902	4	0	1
976	1	1.6	1	G	2	22	2	962	8	0	1
977	1	0	1	G	3	47	2	721	11	1	1
978	2	0.3	1	C	3	48	2	440	8	0	1
979	1	5	2	G	2	16	3	99	4	0	1
980	1	0	2	E	3	35	2	694	6	0	1
981	1	0.7	1	G	2	45	1	251	11	0	1
982	1	0.1	2		3	40	2	873	9	0	1
983	2	0.3	1	D	3	56	2	190	9	3	1
984	2	0	2	G	1	0	2	713	1	0	1
985	1	3.3	2	D	2	32	2	40	21	10	1
986	2	1.2	1	G	2	32	2	873	14	13	1
987	1	1.1	1	E	2	50	2	99	4	0	1
988	1	0.2	2	G	1	0	2	806	1	0	1
989	1	0.9	1	G	2	45	2	251	21	18	1
990	1	0.2	2	B	3	30	1	730	11	10	1
991	1	0.3	1	C	3	43	2	1061	6	0	1
992	1	0.4	2	G	2	25	3	577	8	3	1
993	1	0.3	2	D	3	46	2	700	6	0	1
994	1	1.4	1	E	2	37	3	392	11	8	1
995	1	0.3	1	E	3	40	2	962	8	2	1
996	2	0.3	1	G	3	28	2	1216	21	3	1
997	2	0.2	1		1	0	1	470	1	0	1
998	1	0.4	2	G	2	40	2	733	14	3	1
999	1	0.5	1	G	2	50	2	653	6	0	1
1000	1	1.5	1	D	2	40	1	440	6	0	1
1001	1	0.2	2	B	3	50	2	610	6	0	1
1002	1	1.7	2	F	2	15	3	855		2	1
1003	2	0.1	2	C	3	35	2	855	5	4	1
1004	1	0.3	1	G	3	32	2	892	19	7	1
1005	1	0.2	1	C	3	44	1	934		0	1

1006	1	0.4	2	D	2		2	366	14	12	1
1007	1	1.2	2	E	2	40	2	282	8	4	1

1008	1	2.8	2	B	2	18	2	504	6	2	1
1009	1	1.6	2	G	2	20	2	988	16	2	1
1010	1	1.7	1	E	2	48	2	194	4	0	1
1011	1	2.2	2		2	27	1	519	6	0	1
1012	1	0.2	1	C	3	31	1	1173	11	1	1
1013	2	0.2	1	A	3	47	2	462	11	0	1
1014	1	0.3	1	G	3	45	1	392	11	4	1
1015	2	1.3	1	G	2	25	2	988	6	0	1
1016	1	0.3	2	E	2		3	577	13	3	1
1017	1	1.6	1	C	2	35	1	462	16	9	1
1018	1	1.4	1	A	2	48	2	116	6	0	1
1019	1	0.2	2	B	3	50	2	694	4	0	1
1020	1	0.1	2	G	3	28	1	630	6	0	1
1021	1	0.2	1	E	3	33	1	892	14	4	1
1022	1	2	1	G	2	40	2	694	21	19	1
1023	2	0.3	1	D	3	30	2	636	11	9	1
1024	1	0.3	1	G	3	40	3	840	4	1	1
1025	1	2.2	2	D	2	37	2	190	14	7	1
1026	1	0.6	1	G	2	50	3	311	21	18	1
1027	1	0.1	1	F	3	43	2	560	13	0	1
1028	2	0	1	G	3	39	2	653	4	1	1
1029	2	0	1		3	35	1	1189	4	0	1
1030	1	0.3	1	E	3	50	2	610	4	0	1
1031	2	0.9	1	E	2	40	2	220	11	0	1
1032	1	0.1	2	B	3	52	3	339	6	0	1
1033	1	0.1	2	D	3	47	3	542	26	14	1
1034	2	1.6	1		2	28	2	655	4	0	1
1035	1	0.8	2	E	2	24	2	1103	11	1	1
1036	1	1.5	1	E	2	47	2	132	4	0	1
1037	2	0	1	B	3	26	2	741	11	9	1
1038	1	0.8	1	E	2	40	2	560	8	0	1
1039	1	0	1	E	3	57	1	519	11	0	1
1040	1	1.9	1		2	30	2	490	21	11	1
1041	2	0.6	1	E	2	42	2	904	8	0	1
1042	1	1.2	2	G	2	50	3	275	11	10	1
1043	1	0.1	1	G	3	48	2	578	11	0	1
1044	1	0.3	1	C	3	48	2	655	11	0	1

1045	1	0.3	2	G	3	49	2	311	8	4	1
1046	2	0.2	1	G	3	54	1	550	9	5	1
1047	1	0	1	G	3		1	733	8	0	1
1048	2	1.8	1	G	2	16	2	1082	11	0	1
1049	1	1.1	1	B	2		1	542	8	0	1
1050	2	0.2	1	G	3	33	2	1203	6	0	1
1051	1	0.7	1	E	2	26	1	1203		9	1
1052	1	1.5	2	B	2	34	2	440	21	7	1
1053	1	0.7	2	G	2	50	1	490	6	0	1

1054	2	0.1	1	G	3	51	2	314	6	0	1
1055	1	1.4	1	G	2	40	1	419	8	1	1
1056	1	4.9	2		2	14	2	116	3	0	1
1057	1	0.3	1		3	56	2	220	6	0	1
1058	1	0.2	1	B	3	27	2	722	21	6	1
1059	1	2.3	1	G	2	14	2	1122	9	7	1
1060	1	0.7	1	G	2	37	2	560	21	17	1
1061	1	2	1		2	45	3	40	4	0	1
1062	1	0.2	1	D	3	47	2	934	11	0	1
1063	1	1	1	E	2	38	2	655	31	29	1
1064	2	0	2	F	3	30	2	1173	16	14	1
1065	1	1.3	1	E	2	45	1	694			1
1066	1	1.3	1	E	2	31	2	760	11	0	1
1067	1	0.3	1	G	3	47	2	934	11	3	1
1068	1	0.3	2	G	3	50	1	806	11	2	1
1069	1	1.6	2		2	34	2	565	11	3	1
1070	1	1.5	1	F	2	32	2	462	11	0	1
1071	2	0.3	2	G	3	35	3	902	4	0	1
1072	1	1.3	2	D	2	32	2	524	6	0	1
1073	1	0.2	1		3	60	1	194	6		1
1074	1	0.5	1	B	2	45	2	578	21	5	1
1075	1	3	1	G	2	30	2	424		4	1
1076	1	2.8	1	E	2	21	2	631	11	0	1
1077	1	0.1	2	B	3	32	1	524	8	4	1
1078	2	0.5	1	G	2	21	2	1262	6	0	1
1079	2	0.3	1		3	55	2	519	9	4	1
1080	1	0.4	1	D	2	50	2	565	11	0	1

1081	1	0	2	C	3	44	1	741	31	26	1
1082	1	0.3	2	E	3	32	1	694	14	0	1
1083	2	0.3	1		3	52	1	152	8	0	1
1084	1	3.2	2	D	2	16	2	694	14	6	1
1085	2	0.2	1	C	3	45	2	578	5	0	1
1086	1	1.5	2	B	2	40	1	146	4	0	1
1087	2	0.3	2	E	3		2	649	16	1	1
1088	2	0.1	1	D	1	0	3	605	1	0	1
1089	2	0.3	2		3		1	694	11	2	1
1090	1	0.3	1	G	3	42	2	988	14	0	1
1091	1	0.3	1	E	3	54	3	610	6	3	1
1092	1	0.8	1	E	2	32	2	812	11	0	1
1093	1	0.2	2	G	3	45	1	462	16		1
1094	1	0.1	2	E	3	40	1	636	11	0	1
1095	2	0.3	2		3		2	608		0	1
1096	1	0.3	2	E	3	49	2	484	29	14	1
1097	2	0.3	2		3	35	2	741	31	15	1
1098	2	0.1	2		1	0	2	806		0	1
1099	1	1.6	1	G	2	10	2	713	9	0	1

1100	1	0.3	2	G	3	38	2	1060	11	0	1
1101	1	0.7	2		2	35	2	934	9	0	1
1102	2	0.8	1	E	2	30	2	578	11	0	1
1103	1	1.1	2	G	2	44	2	694	5	1	1
1104	1	0.2	2	C	3	50	2	275		14	1
1105	1	4.5	1	A	2	16	2	99		0	1
1106	1	0	1	E	3	53	2	653	11	3	1
1107	2	0.2	1	C	3	38	2	988	21	7	1
1108	1	0.2	1	E	2	41	2	560	8	6	1
1109	1	0.9	1	D	2	47	2	194	9	4	1
1110	1	0	2	G	3	50	2	462	14	0	1
1111	1	2.4	2	C	2	20	2	560	11	0	1
1112	2	0.3	1	F	3	42	1	542	21	4	1
1113	1	1.6	2		2	28	2	440	6	3	1
1114	1	0.3	1	E	3	30	2	880	10	0	1
1115	1	1.2	2	G	2	41	2	490	11	6	1
1116	1	2.8	1	D	2	25	2	311	11	0	1

1117	2	3	1	G	2	34	2	187	11	4	1
1118	1	3.1	2	G	2	31	2	311	14	1	1
1119	1	0	1	G	3	32	3	730	8	3	1
1120	1	0.3	2	C	3	55	3	416	11	0	1
1121	2	0.3	2	G	3	30	1	314	4	3	1
1122	2	0.7	1	E	2	40	2	630	8	0	1
1123	1	0	2	G	3	10	2	892	6	0	1
1124	2	0.2	1	G	3	40	1	760	8	0	1
1125	1	0.1	1	B	3	60	2	251	9	3	1
1126	1	2.1	1	G	2	34	3	565	11	9	1
1127	1	0.2	1	F	3	58	3	470	11	2	1
1128	1	0.5	1		2	37	3	608	24	22	1
1129	1	0.2	2	G	3	34	2	812	31	24	1
1130	2	1.6	1	G	2	23	2	840	7	0	1
1131	1	0.3	1	D	3	54	2	424	6	0	1
1132	1	1.9	2	G	2	20	2	608	14	11	1
1133	2	0	1	G	3	38	2	678	11	9	1
1134	1	0.2	1	G	3	55	2	187	16	14	1
1135	1	0.1	2	G	3	36	1	636	11	0	1
1136	1	1.8	1	D	2		2	880	11	9	1
1137	1	0.2	1	G	3	55	2	282	4	0	1
1138	1	1.3	2	E	2	37	2	605	16	9	1
1139	1	1.5	2	G	2	45	1	470	21	14	1
1140	2	0	1	G	1	0	3	721	1	0	1
1141	1	0.2	1	G	3	41	2	988	11	0	1
1142	1	3.8	2	E	2	14	3	519	21	16	1
1143	2	0	2		1		1	1316	1	0	1
1144	2	0.3	2	C	3	27	3	722	8	0	1
1145	2	0.2	2		3	50	2	470	11	0	1

1146	2	0.2	1	G	3	30	2	490	4	0	1
1147	1	0	2	F	3	40	2	623	11	0	1
1148	2	0	2	E	3	20	1	623	9	8	1
1149	1	0	1	G	3	53	2	610	6	0	1
1150	1	0.2	2	C	3	40	1	934	19	4	1
1151	1	1	1	B	2	28	3	1061	14	7	1
1152	1	0	1	G	3	34	2	520	4	0	1
1153	1	0	2	G	3	42	3	962	11	0	1
1154	1	0.2	1	F	3	41	2	778	14	0	1
1155	1	2.4	2	G	2	27	2	419	6	0	1
1156	1	0	1	B	3	47	2	560	6	0	1

1157	2	0.1	1	E	3	40	2	577		6	1
1158	1	1.2	1	E	2	41	2	251	14	3	1
1159	1	0.2	2	F	3	47	1	840	6	1	1
1160	1	0.1	2		3		2	605	8	0	1
1161	1	0.1	2	G	3		2	1060	26	25	1
1162	1	0	1	E	3	40	1	653	21		1
1163	1	0.3	1	E	3	46	2	840	6	0	1
1164	1	0.2	2	C	3	50	3	806	8	6	1
1165	1	0.1	1	C	3	59	2	470	21	12	1
1166	1	0.2	2	F	3	46	2	806	21	9	1
1167	2	0.3	1	C	3	51	2	653	8	0	1
1168	1	1.6	1		2	23	1	1103		3	1
1169	2	0.2	2		3	50	1	416	6	1	1
1170	1	0	1	B	3	50	2	416	6	5	1
1171	1	4.1	2		2	16	1	275	6	0	1
1172	1	0.2	1	E	3	40	2	458	9	7	1
1173	1	3.1	1	D	2	27	3	352	8	5	1
1174	1	0.3	1		3	50	2	458	11	0	1
1175	1	0.1	1	G	3	38	1	796	6	0	1
1176	1	1.4	2	E	2	38	3	550	16	10	1
1177	2	0.2	1	F	3	50	2	311	8	6	1
1178	1	0.2	2		3	30	3	1061	13	3	1
1179	1	0.2	1	E	3	37	3	760	11	1	1
1180	1	2.5	1	E	2	36	2	152	11	7	1
1181	2	0.9	1	G	2	51	1	51	6	3	1
1182	1	0	1		3	48	2	733	6	0	1
1183	1	0.2	1	G	3	52	2	490	8	0	1
1184	2	0	1		3	50	2	339	5	0	1
1185	1	0.3	2		1		2	842	1		1
1186	2	3.1	1	G	2	20	2	253	9	5	1
1187	1	0.7	1	C	2	28	3	840	11	0	1
1188	2	0.1	1	E	3	41	2	1060	6	0	1
1189	1	2.6	2	D	2	21	2	771	4	0	1
1190	1	0.2	2	G	3	20	1	1189	4	0	1
1191	2	0	2	D	3	35	1	577	14	0	1

1192	1	3.4	1	F	2	22	1	470	8	1	1

1193	2	0.1	1	F	1	0	2	605	1	0	1
1194	1	4.3	1		2	18	2	50	21	12	1
1195	1	0.9	1	E	2	44	2	424	8	0	1
1196	1	1.4	2		2	25	2	577	8	5	1
1197	1	0.1	1	F	3	50	2	524	8	0	1
1198	1	1.1	1	B	2	15	1	1252	11	0	1
1199	1	0.5	2		2		2	684	14	7	1
1200	1	1.9	2		2	11	2	40	14	9	1
1201	1	1.3	1	C	2	22	2	1102	6	1	1
1202	1	0.1	1	G	3	52	2	282	11	0	1
1203	2	0.2	2		3	27	1	605	11	7	1
1204	1	0.2	1		3	33	2	704	6	0	1
1205	1	1.1	2	D	2	39	3	565	14	8	1
1206	1	0.4	2	F	2	29	2	988	14	0	1
1207	1	0.1	2	G	3		2	795	6	0	1
1208	1	0	1	G	3	50	1	962	16	9	1
1209	2	0.2	2	G	3		1	339	8	0	1
1210	1	0.8	2	G	2	40	3	251	11	0	1
1211	2	0	2	E	3	35	2	462	5	0	1
1212	2	0.2	2	E	3	42	2	962	6	0	1
1213	2	0.1	1		3	60	3	99	14	0	1
1214	1	2.3	2	F	2	14	3	520	8	7	1
1215	1	0.1	1		3	41	2	988	11		1
1216	2	1.5	1	G	2	27	1	934	3	0	1
1217	1	0	1	G	3		1	700	9	7	1
1218	2	0.1	1	G	3	39	2	608	9	7	1
1219	2	0.3	2	F	3	42	2	988	21	20	1
1220	1	2.4	1		2	28	2	220	5	4	1
1221	1	0.6	1	F	2	39	2	649	9	7	1
1222	2	0.3	1	E	3	50	2	550	6	0	1
1223	1	0	1		3	45	2	605	9	3	1
1224	1	2.2	1		2	18	2	577	11	0	1
1225	2	0.1	2	G	3	30	2	812	9	0	1
1226	1	3.3	1	E	2	28	2	152	4	0	1
1227	1	1.2	2	E	2	30	1	721	11	4	1
1228	1	0.1	2	G	3	35	2	662	11	9	1

1229	1	0	1		3	44	3	934	14	0	1
1230	1	0.1	1	G	3	64	3	132	6	2	1
1231	1	0.1	1		3		2	577	14	10	1
1232	1	2.6	2	D	2	41	3	116	11	5	1
1233	1	1.6	1	B	2	41	2	352	6	5	1
1234	2	0.3	1	G	3	50	1	458	4	0	1
1235	1	0.1	1	C	3	37	3	636	14	0	1
1236	2	0.1	1	D	3	50	1	504	4	0	1
1237	1	0.1	1	F	3		1	610	6	4	1

1238	1	0.6	1	D	2	50	1	186	11	2	1
1239	1	0.1	2	G	3	35	2	608	9	5	1
1240	1	2.2	1	G	2	30	2	99	11	0	1
1241	1	0	1	E	3	45	2	873	11	0	1
1242	1	0.7	1		2	50	2	220	6	5	1
1243	1	0	1	G	3	34	2	741		9	1
1244	2	0.3	1	E	3	51	2	578	6	0	1
1245	2	0	1	C	3	46	2	458	11	0	1
1246	1	1.3	1	G	2	40	3	187	11	5	1
1247	1	0.4	2	G	2	41	3	694	10	0	1
1248	1	0.3	1	B	3	48	2	806	6	0	1
1249	2	0	2	G	3	55	1	366	9	0	1
1250	1	0.2	1		3	56	2	314	9	7	1
1251	1	1.8	2	D	2	30	3	424	21	5	1
1252	2	0.1	2	D	1	0	2	1082	1	0	1
1253	2	0.1	2	E	3		2	366	11	8	1
1254	1	3.1	1	G	2	15	2	988	21	18	1
1255	2	0.1	1	F	3	45	2	542	11	0	1
1256	1	0.2	1		3	44	3	873	21	16	1
1257	1	0.7	1	E	2		1	892	6	0	1
1258	2	0.3	2		3	14	1	841	6	0	1
1259	1	0.1	1	D	3	32	2	868	6	5	1
1260	2	0.4	1	B	2		3	220	9	0	1
1261	2	1.6	1	D	2	30	3	366	5	1	1
1262	1	0.3	1	A	1	0	2	1216	1	0	1
1263	1	0.2	2		3	51	1	440	6	0	1
1264	1	0.3	1	C	3	30	2	713	11	0	1

1265	2	0.2	2	E	3		3	673	11	5	1
1266	1	1.8	2	C	2	31	2	694	16	3	1
1267	2	0	1	E	3	37	2	694	11	2	1
1268	1	0.3	2	E	3	30	1	842	6	1	1
1269	1	2.4	1	G	2	35	3	152	6	5	1
1270	1	2.4	1		2	27	2	519	14	1	1
1271	2	0.3	1	G	3	48	2	462	14	0	1
1272	1	0.3	1	G	3	50	1	565	4	0	1
1273	1	0	1	G	3	37	2	988	11	5	1
1274	1	3.6	2		2	16	3	610	6	0	1
1275	1	0.3	2	G	3		1	389	6	5	1
1276	1	0.5	2	C	2		1	912	26	24	1
1277	1	1.3	2	D	2	32	2	577	16	7	1
1278	1	0.3	1	C	3	45	2	700	16		1
1279	1	1.2	1	E	2	50	2	233	11	1	1
1280	1	0.1	1	G	3	36	2	1103	11	6	1
1281	2	0.3	2	G	3	20	2	700	6	0	1
1282	1	1.5	2		2	32	2	655	6	1	1
1283	1	2.5	2		2	31	2	339	3		1

1284	2	0.3	1	G	3	29	2	841	14	7	1
1285	1	2.2	2	G	2	36	2	220	11	0	1
1286	1	0.2	1		3	65	3	40	4	0	1
1287	1	0.3	2	F	3	30	2	722	16	12	1
1288	1	2.1	1	C	2	20	3	1122	11	4	1
1289	1	0.2	1	G	3	60	2	40	11	5	1
1290	2	0	1	E	3	36	3	1102	3	1	1
1291	1	0.3	1	B	3	57	2	233	4	1	1
1292	1	0	2		3	54	2	366	6	2	1
1293	1	0.3	2	D	3	60	2	275	8	2	1
1294	1	0.3	1	G	3	50	1	490	4	0	1
1295	2	0.2	2	E	3	36	2	684	8	0	1
1296	1	2.3	2	G	2	20	2	484	5	0	1
1297	2	0.1	2		3	45	1	608	26	22	1
1298	1	0.4	1	C	2	54	1	314	14	0	1
1299	1	0.7	2	G	2	48	1	190	5	0	1
1300	1	0	1	E	3	49	2	462	8	0	1

1301	1	0	2		3	31	3	1203	14	3	1
1302	1	0	1	C	3	54	3	458	16	6	1
1303	2	0.3	2	E	3	23	2	1122	16	11	1
1304	1	2.7	1	E	2		3	610	3	2	1
1305	1	0.3	1	E	3	41	3	636	6	4	1
1306	2	0.3	1	D	3	40	2	1038	8		1
1307	1	0.6	1	E	2	40	3	806	21	13	1
1308	1	1.7	2	G	2	26	1	1038	16	13	1
1309	1	0	1	G	3	52	3	392	11	7	1
1310	1	0.2	2	G	3	50	2	220	11	8	1
1311	2	0.2	1	G	3	30	2	841	6	2	1
1312	1	2.1	2		2	30	3	550	11	6	1
1313	1	0.9	2	G	2	22	2	1173	11	2	1
1314	2	0.2	1	E	3	18	2	636	14	3	1
1315	1	0	2	E	3	40	2	1189	11	6	1
1316	2	0.1	2		3	31	3	722	26	12	1
1317	1	0.3	2	D	3	60	3	352	14	0	1
1318	2	0	1	G	3	54	2	458	3	0	1
1319	2	0.3	1		3	45	3	630	8		1
1320	1	3.1	1	G	2	26	3	314	4	3	1
1321	2	0.2	1	D	2	39	3	366	11	7	1
1322	1	2.3	2		2	23	3	550	21	3	1
1323	2	0.2	1	E	1		2	768	1	0	1
1324	1	0.3	1	E	3		3	389	11	5	1
1325	1	0	2	D	3	40	2	636	11	0	1
1326	1	0.3	1	E	3	46	3	550	21	17	1
1327	2	0.2	1	G	3	48	2	251	6	0	1
1328	2	0.2	2	C	3	30	2	722	4	0	1
1329	1	1.4	1		2	40	3	282	6	0	1

1330	1	2.5	2	F	2	40	2	233	4	0	1
1331	1	0.2	1		3	40	1	721	7	1	1
1332	1	0.3	1	E	3	16	2	841	9	1	1
1333	1	0.1	2	F	3	39	2	636	16	10	1
1334	1	0.3	1		3	42	2	560	6	0	1
1335	1	0	2	A	3	50	2	524	11	0	1
1336	1	0.8	1	C	2	34	2	1061	14	4	1

1337	1	0.3	2	G	3	50	2	610	11	9	1
1338	2	0	2		3		2	275	4	2	1
1339	1	0.3	1	G	3		2	593	21	12	1
1340	2	0.3	1	G	3	37	2	662	6	0	1
1341	1	1.9	1	G	2	32	2	392	11	0	1
1342	1	0.3	1	G	3		2	524	8	3	1
1343	1	1.1	2	G	2	50	2	251	6	0	1
1344	1	1.2	1	G	2	25	3	904	6	0	1
1345	2	0.1	1	E	3	50	3	389		0	1
1346	1	0	1	F	3	50	3	605	11	0	1
1347	1	0.5	2		2	32	1	795	9	0	1
1348	1	0	1	C	3	54	1	352	11	9	1
1349	1	0.1	1	G	3	42	2	904	16	12	1
1350	1	0.1	1	E	3		3	630	8	0	1
1351	2	0.7	1	G	2	32	2	730	21		1
1352	1	0.4	1	C	2	42	2	678	8	0	1
1353	1	0	2	G	3	48	1	524	8	0	1
1354	1	1.3	1	C	2		2	631	11	10	1
1355	1	0.5	2	C	2	35	2	721	16	0	1
1356	1	0	1	G	3	50	2	904	8	0	1
1357	1	0.5	2	G	2	28	1	636	14	9	1
1358	1	0.5	2	G	2	28	1	678	11	1	1
1359	2	0	2	E	1		1	868	1	0	1
1360	1	0.1	1		3	50	2	550	9	0	1
1361	1	0.9	2	E	2	44	2	565	21	12	1
1362	2	0.1	2	G	3	40	2	806	11	9	1
1363	2	0.3	2	G	3	47	2	504	11	4	1
1364	1	1.5	2	D	2	28	2	988	13	0	1
1365	1	3	1		2		3	116	4	0	1
1366	2	0.1	1	B	3	26	2	922	31	27	1
1367	2	0.2	2	F	3	28	2	684	9	0	1
1368	1	0.3	2	G	3	32	2	855	3	0	1
1369	1	0.3	1	G	3	55	2	352	6	0	1
1370	2	0.3	2	D	3	48	2	840	9	0	1
1371	1	0.2	1	B	3	40	2	1061	6	0	1
1372	1	0.3	1	E	3	46	2	806	4	0	1
1373	2	0.1	1	G	1	0	2	605	1	0	1

1374	2	0.2	1	D	3	48	2	806	6	0	1
1375	1	0	1	G	3	61	2	311	11	5	1

1376	2	0.1	2	G	3	40	1	770	9	0	1
1377	1	0	1		3	46	3	524	16	3	1
1378	1	0.3	1	E	3	42	2	655	4	3	1
1379	2	0	1		2		2	630	13	8	1
1380	2	0.3	2	D	1	0	3	939	1	0	1
1381	2	0.1	2	D	1	0	2	880	1	0	1
1382	2	0.9	1	C	2	28	3	868	13	11	1
1383	1	0.2	2	G	3	41	2	812	14	0	1
1384	1	0.1	2	C	3	50	2	490	4	0	1
1385	1	0.3	2	D	3	53	2	694	11	8	1
1386	2	0.6	1	E	2	40	3	524	9	6	1
1387	1	1.3	1		2		2	684	11	0	1
1388	2	0.3	2	G	3	30	2	233	8	1	1
1389	1	0.3	1	C	3	40	3	653	6	0	1
1390	2	0.1	1	G	3		3	868	16	0	1
1391	2	0.3	2	A	1	0	2	1014	1	0	1
1392	2	0.2	2	G	3		1	392	6	0	1
1393	1	1	1	G	2	41	1	694	9	0	1
1394	1	1.5	2	G	2	28	1	771	6	0	1
1395	1	0.7	2	A	2	41	2	416	22	3	1
1396	2	0.2	2	G	1	0	2	339	1	0	1
1397	1	0	2	G	3	40	2	578	4	0	1
1398	2	0.3	1	D	2	28	2	721	5	0	1
1399	2	0.1	2	G	3	46	2	524	9	0	1
1400	1	0	2	G	3	40	2	778	24	17	1
1401	1	0	2	E	1	0	1	187	1	0	1
1402	1	0.1	1	E	3	40	3	904	11	0	1
1403	1	1.7	1	G	2	20	2	1061	14	13	1
1404	1	0	1	G	3	40	2	623	11	0	1
1405	1	0.3	1	D	3		2	608	6	5	1
1406	1	0.2	1	D	3	42	2	593	8	5	1
1407	1	0.2	1	G	3	60	3	419		0	1
1408	1	1.2	2	E	2	40	1	392	11	9	1
1409	1	0	1	G	3	43	1	560	16	3	1
1410	2	3.5	1		2	14	1	440	4	0	1

1411	1	0.1	2		3	34	3	694	7	0	1
1412	1	0.1	2		3	39	2	760	6	0	1
1413	2	0.2	1	G	1	0	2	842	1	0	1
1414	2	0.3	2	G	3	43	3	339	8		1
1415	1	0.3	1		3	47	2	721	11	2	1
1416	1	0.3	1	F	3	65	2	419	31	17	1
1417	2	0.3	2	G	3	40	2	1082	9	2	1
1418	2	0	2	G	1	0	2	934	1	0	1
1419	1	0	1		3	30	2	1122	18	7	1
1420	1	0.6	1	B	2	34	2	649	8	7	1
1421	1	1	1	C	2	38	2	521	11	3	1

1422	1	0.2	2	E	3	36	2	713	16	9	1
1423	1	0.1	2		2	52	2	518	14	10	1
1424	1	1.9	2	D	2	12	2	1014	6	0	1
1425	2	0.2	1		1	0	2	542	1	0	1
1426	1	0.6	1	G	2	48	2	840		12	1
1427	2	0.1	1	G	3	53	2	152	11	6	1
1428	1	0.2	2	G	3	41	2	593	9	0	1
1429	1	0.4	1	G	2	45	3	962	16	8	1
1430	1	0.2	2	D	3	30	2	892	14	5	1
1431	1	0.7	2	E	2	45	2	694	7	0	1
1432	1	1.9	2	C	2	10	3	892	9	7	1
1433	2	0.3	1	G	3	42	2	560	11	10	1
1434	1	1.3	2	G	2	40	2	190	9	0	1
1435	1	0.1	1	E	3	43	2	484	11	6	1
1436	1	0.3	1	F	3	30	2	1203	6	1	
1437	1	0.6	1		2	51	2	311	11	2	1
1438	2	2.1	1	G	2	31	2	490	9	0	1
1439	2	0	2	G	3		2	934	6	0	1
1440	2	0	1	G	3	16	2	778	11	4	1
1441	2	0	2		3	45	3	630	6	0	1
1442	1	0.3	1	D	3	41	1	721	14	6	1
1443	1	0	1	G	3		2	842	4	3	1
1444	1	0.3	1		3	40	2	827	14	4	1

1445	2	0.3	1	G	3	36	2	812	11		1
1446	1	0.3	2	B	3	46	3	655	31	21	1

BIBILOGRAPHY

BIBLIOGRAPHY

- Multivariate data analysis (Fifth Edition) --- Joseph F.Hair, Rolph E.Anderson, Ronald l Tatham and William C.Black
- Data Mining- Theories, Algorithms, and Examples – NoNG YE
- A Practical Guide to Data Mining for Business and Industry -- Andrea Ahlemeyer-Stubbe, Shirley Coleman
- Data Mining and Predictive Analytics – Daniel T. Larose, Chantal D.Lorse
- machine_learning_mastery_with_r. – Jason Brownlee
- master_machine_learning_algorithms -- Jason Brownlee
- statistical_methods_for_machine_learning - Jason Brownlee
- Machine Learning Using R -- Karthik Ramasubramanian, Abhishek Singh
- Data Science for Business - Forster Provost & Tom Fawcett
- Deep learning with Deep learning R by François Chollet