A Mini Project Report on

Cardiac Arrhythmias Classification

Submitted in partial fulfillment of requirements for the award of VI semester of

Bachelor of Engineering

in

Computer Science and Engineering

by

Yashwanth G

1601-16-733-119

Dharanish P

1601-16-733-182

Under the guidance of

Dr K. Sagar	**G. Vivek**
Professor	Asst. Professor
Department of CSE	Department of CSE
CBIT, Hyderabad	CBIT, Hyderabad

Department of Computer Science and Engineering

Chaitanya Bharathi Institute of Technology

Gandipet , Hyderabad-500075.

CERTIFICATE

This project is to certify that the project titled "Cardiac Arrhythmias Classification" is a bonafide work carried out by Yashwanth G and Dharanish P In completion of a mini project under our guidance and supervision of Dr. K. Sagar and Mr. G. Vivek The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Mentors

Dr. K. Sagar, Assoc Prof

Mr. G. Vivek, Asst Prof

Head of Department

Dr. M Swamy Das

Professor and Head

Department of CSE

Batch Incharge

G. Mamatha, Asst Prof

ABSTRACT

Cardiac Arrhythmia, also known as irregular heartbeat or cardiacDysthymia, is a group of conditions where the heartbeat is irregular,too slow, or too fast. An arrhythmia is caused by a disruption of yourheart's normal electrical system, which regulates your heart rate
and heart rhythm.

In this project, we aim to classify heart arrhythmias patients among16 different classes based on ECG (Electrocardiography) data. Themethodology and the outcomes of developing a Machine Learningsystem that is capable of classifying a patient into 16 differen cardiac arrhythmic categories. Various Machine Learning algorithms likeSVM, Logistic Regression, KNN, Random Forest and Decision Treescan be used to classify the patients.

This work has great potential to serve the medicine industry. Thiswork can be of immense importance to researchers who areexploring various techniques to capture the key pre-informers of apotential cardiac disease, well before it is too late.

List of Figures

TABLE OF CONTENTS

1. Introduction

1.1 Objective

In this project, w aim to classify heart arrhythmias patients among 16 different classes based on ECG (Electrocardiography) data. This work has great potential to serve the medicine industry. Algorithms such as these can be have ground breaking impact regarding helping researchers target the keys features that cause cardiac arrhythmia and assist them in classifying patients in right categories, so as to be able to take measures in the right direction.

1.2 Problem Definition

Classifying patients into 16 classes aaccording to the ECG data. Patient's fata contains their age, gender, height and weight. Most of the patient's data contains the values derived from their ECG test.

16 classes are:

01	Normal
02	Ischemic changes
03	Old Anterior Myocardial Infarction
04	Old Inferior Myocardial Infarction
05	Sinus tachycardia
06	Sinus bradycardia
07	Ventricular Premature Contraction
08	Supraventricular Premature Contraction
09	Left bundle branch block
10	Right bundle branch block
11	1. Degree Atrioventricular block
12	2. Degree AV block
13	3. Degree AV block
14	Left ventricular hypertrophy
15	Arial Fibrillation or Flutter
16	Others

1.3 Existed System

In the existed system doctors use to examine the ECG test reports and have to find out what is the type of cardiac problem the patient is facing. It consumes more time and effort to examine the reports.

1.4 Proposed System

With the help of this project we try to automate the effort of the physician by predicting the type of cardiac Arrhythmias the patient is facing. It will decrease the time and we can address the patient with the medicine required.

1.5 Organization of Report

The existing methodologies are used.

2. Literature Survey

Cardiac Arrhythmias is a main term of heart. This work started mainly due to more heart failures and many are dying due to heart diseases.

What is Cardiac Arrhythmia?Heart arrhythmia, also known as irregular heartbeat or cardiac dysrhythmia is a group of conditions where the heartbeat is irregular, too slow, or too fast. An arrhythmia is caused by a disruption of your heart's normal electrical system, which regulates your heart rate and heart rhythm. The severity of cardiac arrhythmias can vary tremendously. Most arrhythmias are completely benign and inconsequential, while others are extremely dangerous and life-threatening. And many of them, while not particularly dangerous, produce symptoms that can be quite disruptive to your life.A healthy person will hardly ever suffer from long-term arrhythmia unless they have an external trigger, such as drug abuse or an electric shock. If there is an underlying problem, however, the electrical impulses may not be able to travel through the heart correctly, increasing the likelihood of arrhythmia. Some patients have no symptoms, but a doctor might detect an arrhythmia during a routine examination or on an EKG. Even if a patient notices symptoms, it does not necessarily mean there is a serious problem; for instance, some patients with life-threatening arrhythmias may have no symptoms while others with symptoms may not have a serious problem. The noticeable symptoms caused by arrhythmias generally fall into four major categories, including:

- Palpitations
- Dizziness
- Syncope (fainting)
- Cardiac arrest

Depending on the type and severity of your arrhythmia, you may notice symptoms such as sweating, feeling like your heart is racing or fluttering, noticing extra heartbeats, feeling your heartbeat has slowed, chest pain, or shortness of breath.

The following are the risk factors for Arrhythmia:

- Old age
- Inherited gene defects
- Heart problems
- Hypothyroidism
- Hypertension
- Obesity
- Illegal drugs
- Too much caffeine
- Heavy and regular alcohol

Making the correct diagnosis of a cardiac arrhythmia generally requires capturing it on an electrocardiogram (ECG) or other heart-monitoring tests, along with a physical exam and complete medical history. If your doctor doesn't find an arrhythmia with a heart-monitoring test, he or she might use a stress test, a tilt table test, or do an electrophysiology study.

Just as there are many types of heart rhythm problems, many different treatment options are available. Deciding which treatment to use for which arrhythmia can be challenging even for cardiologists. The most common options for treating cardiac arrhythmias include:

- Anti Arrhythmiac drug therapy
- Pacemakers
- Implantable defibrillators
- Ablation procedures

If making the right diagnosis or deciding on the best therapy turns out to be difficult, you may be referred to a cardiac electro physiologist—a cardiologist who specializes in heart rhythm disorders.Most heart arrhythmias aren't any cause for concern, even if they cause symptoms. If you do have symptoms of an arrhythmia, see your doctor but don't panic. You may need treatment to help control your symptoms, but the good news is that most people with arrhythmias have no trouble going about their daily activities and living normally. Treatment and implementing lifestyle changes like exercising, eating a heart-healthy diet, and watching your weight can help keep many symptomatic arrhythmias controlled and stop them from becoming dangerous.

3. Methodology

3.1 System Design

System Design of the project mainly focus on the data and training of the algorithm. We directly train the data and test the fit model with the test data. Output will be given directly on the console.

3.1.1 Proposed Algorithm

To implement the classification we used Logstic Regression algorithm. is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables. Sometimes logistic regressions are difficult to interpret; the Intellects Statistics tool easily allows you to conduct the analysis, then in plain English interprets the output. In statistics, the logistic model (or legit model) is a widely used statistical model that, in its basic form, uses a logistic function to model a binary dependent variable; many more complex extensions exist. In regression analysis, logistic regression is estimating the parameters of a logistic model; it is a form of binomial regression.

3.1.2 Diagrammatic Representation

DFD

Level-0

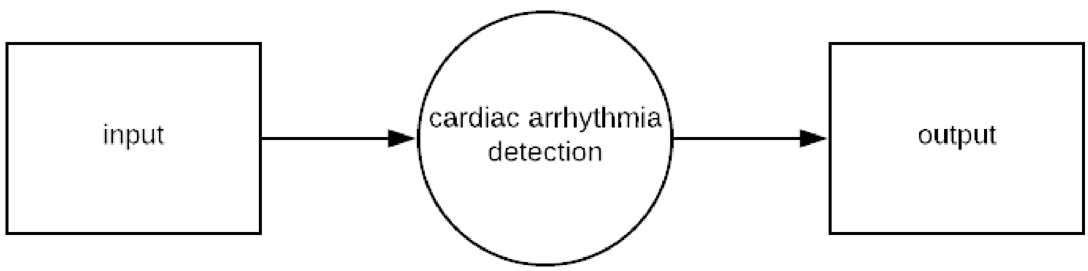


Fig.3.1 DFD Level-0

Level-1

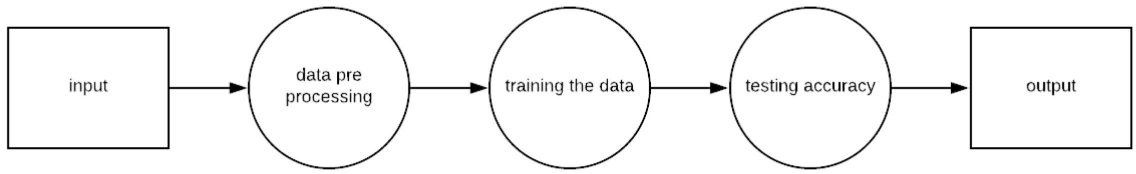


Fig.3.2 DFD Level-1

Level-2

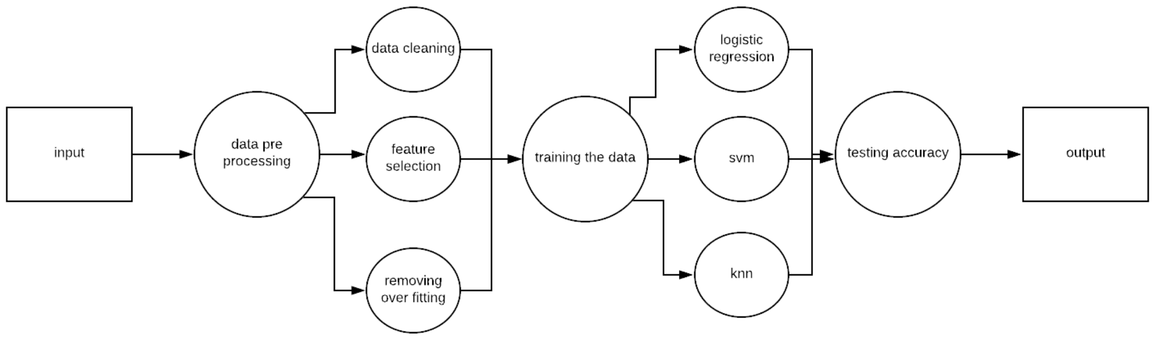


Fig.3.3 DFD Level-2

3.2 Implementation of Proposed solution

Implementation contains mainly two parts:

3.2.1 Data Analysis and Preprocessing

Data analysis and preprocessing includes

Data cleaning

- In data cleaning, we remove the duplicate rows or istances.
- Drop the irrelevent attributes which are not useful.
- We remove the null values with the mean of that attribute.

Feature selection

- Select the features which effects the data the most.
- Choose some of the attributes with which

	0	1	2	3	4	5	6	7	8	9	...	270	271	272	273
0	64	1	156	82	87	171	380	152	102	-36	...	0.0	3.9	-1.0	0.0
1	53	1	160	60	86	133	338	159	82	74	...	-0.4	11.6	-5.3	0.0
2	26	1	160	65	71	150	350	165	81	66	...	-0.4	12.1	0.0	0.0
3	43	1	157	71	80	162	383	141	84	53	...	0.0	10.8	-1.5	0.0
4	53	1	155	63	74	165	386	150	97	29	...	0.0	5.3	-0.5	0.0
5	27	1	162	53	82	168	374	167	78	66	...	0.0	12.5	-1.5	0.0
6	36	1	158	58	76	136	373	152	80	74	...	0.0	6.6	-1.0	0.0
7	67	1	165	53	78	180	348	126	112	74	...	0.0	13.6	0.0	0.0
8	30	0	167	77	87	164	362	168	89	71	...	-1.1	13.4	-1.6	0.0
9	32	1	160	58	83	122	386	174	54	81	...	-0.5	10.0	0.0	0.0
10	27	1	167	48	80	152	343	164	96	77	...	-0.5	12.9	-1.4	0.0
11	48	1	155	55	81	0	382	209	63	75	...	0.0	14.0	0.0	0.0
12	64	0	160	63	83	0	364	120	90	29	...	0.0	6.7	-0.4	0.0
13	62	1	163	60	80	185	354	166	107	-2	...	0.0	11.2	-1.0	0.0
14	50	0	172	80	103	142	366	161	94	54	...	0.0	11.4	-4.7	0.0
15	47	1	150	48	75	132	350	169	65	36	...	0.0	7.7	-0.8	0.0
16	44	1	160	88	77	158	399	163	94	46	...	-0.6	12.4	0.0	0.0
17	37	0	176	72	88	153	389	172	89	67	...	-0.9	16.6	-3.4	0.0
18	37	1	160	50	74	143	374	146	75	68	...	0.0	11.4	-0.9	0.0
19	45	1	165	86	77	143	373	150	65	12	...	0.0	4.4	-2.2	0.0

	274	275	276	277	278	class
0	0.0	0.7	1.6	6.6	19.0	1
1	0.0	1.0	2.1	0.5	17.3	1
2	0.0	-0.3	2.5	26.4	43.4	1
3	0.0	0.4	1.9	23.5	36.4	1
4	0.0	0.5	1.0	12.0	19.6	1
5	0.0	0.6	3.0	30.1	52.3	1
6	0.0	0.8	2.5	12.7	32.7	1
7	0.0	0.7	1.6	40.8	50.7	1
8	0.0	0.5	0.2	22.5	24.2	1
9	0.0	0.4	1.6	21.6	35.0	1
10	0.0	0.9	0.9	25.9	32.9	1
11	0.0	-0.1	2.9	33.6	59.7	1
12	0.0	0.3	0.4	23.7	26.4	1
13	0.0	0.9	3.8	28.3	60.9	1
14	0.0	0.5	1.6	14.2	27.6	1
15	0.0	0.6	1.7	17.2	31.1	1
16	0.0	0.3	1.7	39.2	54.1	1
17	0.0	0.7	1.8	24.9	41.4	1
18	0.0	0.7	1.8	40.1	55.5	1
19	0.0	0.5	1.5	4.9	17.2	1

[20 rows x 280 columns]

Fig.3.4 Training Data

3.2.2Training and fitting a model

- Training the choosen model with the preprocessed data.
- More training data increases the accuracy.
- Training depends on data and model seected.

```
for name, model in models:
    kfold = model_selection.KFold(n_splits=10, random_state = seed)
    cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring=scoring)
    results.append(cv_results)
    names.append(name)
    msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)
```

```
LR: 0.629129 (0.089640)
KNN: 0.601051 (0.075987)
SVM: 0.540015 (0.052591)
```

Fig.3.5 Training Output

3.2.3 Flow of implementation

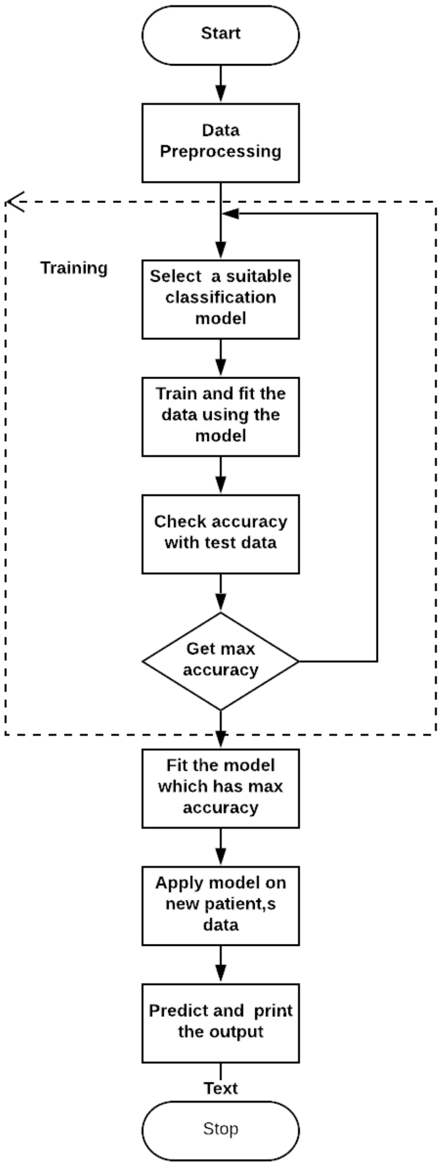


Fig.3.6 Flow Diagram of Algorithm

3.3 System Requirements

3.3.1 Software requirements

- Pandas, Numpy, Keras, Google colab

3.3.2 Hardware requirements

- Desktop with a minimum of 12gb ram.
- GPU or TPU

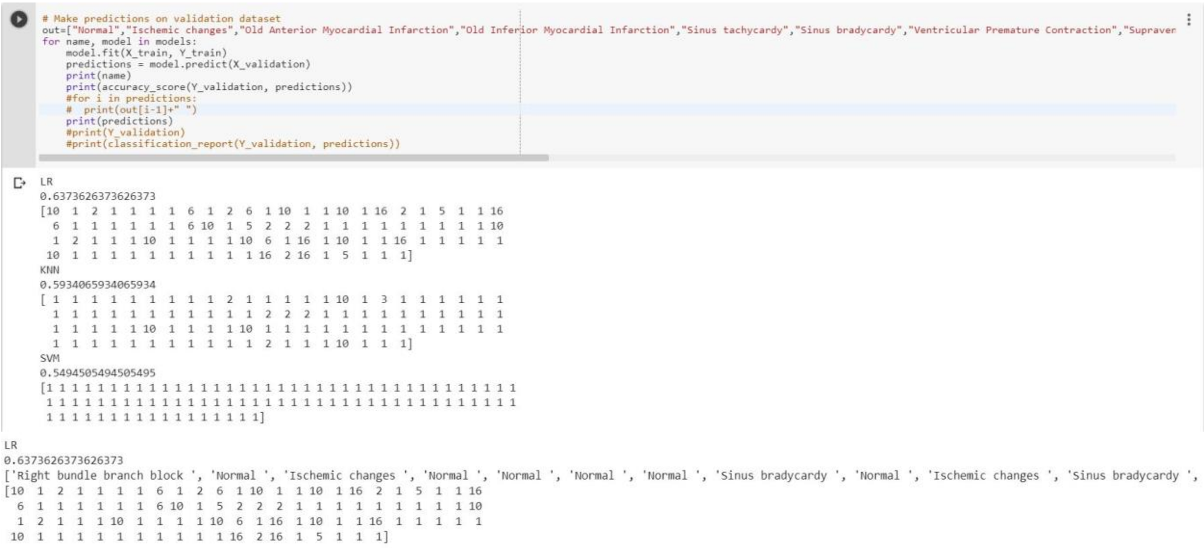
4. Results and discussions

Here as in the table(1), Logistic regression gives the moste accuracy during the training and testing.

As can be seen SVM with linear kernel gave the best performance. Trees on the other hand did not perform so well. One of the possible reasons may be the presence of 16 classes. Additionally, we used Cross Validation to obtain error estimate on the test set. This helped us to be sure that the error was the mean of all the various test sets that could be obtained from the given data and so results are less sensitive to the choice of test/training set. We used 10 k-folds for the purpose. Figure 5 shows the comparison between errors that we got for different models.

Model	Training Accuracy	Testing Accuracy
Logistic Regression(LR)	62.91%	63.72%
K-Nearest Neighbour(KNN)	60.90%	59.34%
Support Vector Machine(SVM)	54.00%	54.94%

Accuracy Table



```
# Make predictions on validation dataset
out=["Normal","Ischemic changes","Old Anterior Myocardial Infarction","Old Inferior Myocardial Infarction","Sinus tachycardy","Sinus bradycardy","Ventricular Premature Contraction","Supraven
for name, model in models:
    model.fit(X_train, Y_train)
    predictions = model.predict(X_validation)
    print(name)
    print(accuracy_score(Y_validation, predictions))
    #for i in predictions:
    #  print(out[i-1]+" ")
    print(predictions)
    #print(Y_validation)
    #print(classification_report(Y_validation, predictions))
```

```
LR
0.6373626373626373
[10  1  2  1  1  1  1  6  1  2  6  1 10  1  1 10  1 16  2  1  5  1  1 16
  6  1  1  1  1  1  1  6 10  1  5  2  2  2  1  1  1  1  1  1  1  1  1 10
  1  2  1  1  1 10  1  1  1  1 10  6  1 16  1 10  1  1 16  1  1  1  1  1
 10  1  1  1  1  1  1  1  1  1  1 16  2 16  1  5  1  1  1]
KNN
0.5934065934065934
[ 1  1  1  1  1  1  1  1  1  2  1  1  1  1  1 10  1  3  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  2  2  2  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1 10  1  1  1  1 10  1  1  1  1  1  1  1  1  1  1  1  1  1
  1  1  1  1  1  1  1  1  1  1  1  2  1  1  1 10  1  1  1]
SVM
0.5494505494505495
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
LR
0.6373626373626373
['Right bundle branch block ', 'Normal ', 'Ischemic changes ', 'Normal ', 'Normal ', 'Normal ', 'Normal ', 'Sinus bradycardy ', 'Normal ', 'Ischemic changes ', 'Sinus bradycardy ',
[10  1  2  1  1  1  1  6  1  2  6  1 10  1  1 10  1 16  2  1  5  1  1 16
  6  1  1  1  1  1  1  6 10  1  5  2  2  2  1  1  1  1  1  1  1  1  1 10
  1  2  1  1  1 10  1  1  1  1 10  6  1 16  1 10  1  1 16  1  1  1  1  1
 10  1  1  1  1  1  1  1  1  1  1 16  2 16  1  5  1  1  1]
```

Fig4.1 Test Output

5. Conclusion and future work

Since about 50% of the data were clustered in class 1, it is believed that with more data on the patients of the other classes, it is possible to learn more to get more accurate classification. Some of the classes had only 2 to 3 instances in the data; which makes it difficult to learn about these classes and hence their misclassification's probability is high when using various algorithms. It is clear that class 1 has the dominant effect on the predicting models so collecting more instances of patients in the other classes is a goal for better predictions in future. In future we can increase the classification of arrhythmias patients with ease and less data sets required and get more accuracy out of all graphs. In future we can build user interface to the project where we can give the patients data input easily on a graphical interface. In future, using image processing and deep learning, we can extract the data from the ECG report of the patient directly.

References

- https://www.verywellhealth.com/overview-of-cardiac-arrhythmias-1746267
- https://www.medicalnewstoday.com/articles/8887.php
- http://cs229.stanford.edu/proj2014/AlGharbi%20Fatema,%20Fazel%20Azar,%20Haider%20Batool,%20Cardiac%20Arrhythmias%20Patients.pdf
- https://pandas.pydata.org/pandas-docs/stable/
- https://keras.io/
- https://scikit-learn.org/stable/documentation.html
- https://arxiv.org/abs/1801.10033

Source code

```
import sys
import scipy
import numpy
import matplotlib
import pandas
import sklearn

print('Python: {}'.format(sys.version))
print('scipy: {}'.format(scipy.__version__))
print('numpy: {}'.format(numpy.__version__))
print('matplotlib: {}'.format(matplotlib.__version__))
print('pandas: {}'.format(pandas.__version__))
print('sklearn: {}'.format(sklearn.__version__))
import warnings
warnings.filterwarnings("ignore")



from pandas.plotting import scatter_matrix
from sklearn.impute import SimpleImputer
import matplotlib.pyplot as plt
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC


# Load Dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/arrhythmia/arrhythmia.data"
df = pandas.read_csv(url,header=None)
df.rename(columns={279:'class'}, inplace=True)

#dataset.drop()
df=df.sort_values(["class"],ascending=True)
df.reset_index(drop=True, inplace=True)
df.drop(list(range(0,50)),axis=0,inplace=True)
df.reset_index(drop=True, inplace=True)
#print(df.head(20))


#for i in range(0,280):
  #print(i)
#tempm = numpy.mean(dataset[i])
df = df.replace('?',0)
#print(dataset.mean())
```

```
# Shape
print(df.shape)


# Head
print(df.head(20))


# descriptions
print(df.describe())


# class distribution
print(df.groupby("class").size())


# Split-out validation dataset
array = df.values
Y = array[:,279]
Y=Y.astype('int')
X = array[:,0:279]
#print(X[0])
from sklearn.feature_selection import RFE
model = LogisticRegression()
rfe = RFE(model, 20)
fit = rfe.fit(X, Y)
print("Num Features: %d" % fit.n_features_)
print("Selected Features: %s" % fit.support_)
print("Feature Ranking: %s" % fit.ranking_)



print("Selected Features: %s" % fit.support_)
suited=[]
for i in range(0,len(fit.support_)):
  if fit.support_[i]==False:
suited.append(i)
print(suited)
print(df.shape)
df=df.drop(suited,axis=1)
print(df.shape)

#print(Y)
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y,
test_size = validation_size, random_state = seed)
```

```
print(df.groupby("class").size())


# Test options and evaluation metric
seed = 7
scoring = 'accuracy'


models = []

models.append(('LR', LogisticRegression()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('SVM', SVC()))

# evaluate each model in turn
results = []
names = []
#print()



print(df.shape)
for name, model in models:
kfold = model_selection.KFold(n_splits=10, random_state = seed)
cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold,
scoring=scoring)
results.append(cv_results)
names.append(name)
msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
    print(msg)



# Make predictions on validation dataset

for name, model in models:
model.fit(X_train, Y_train)
    predictions = model.predict(X_validation)
    print(name)
    print(accuracy_score(Y_validation, predictions))
    print(predictions)
    print(Y_validation)
    #print(classification_report(Y_validation, predictions))
```

Dataset

75,0,190,80,91,193,371,174,121,-16,13,64,-
2,?,63,0,52,44,0,0,32,0,0,0,0,0,0,0,44,20,36,0,28,0,0,0,0,0,0,52,40,0,0
,0,60,0,0,0,0,0,0,52,0,0,0,0,0,0,0,0,0,0,0,56,36,0,0,32,0,0,0,0,0,0,4
8,32,0,0,0,56,0,0,0,0,0,0,80,0,0,0,0,0,0,0,0,0,0,0,0,40,52,0,0,28,0,0,0
,0,0,0,0,48,48,0,0,32,0,0,0,0,0,0,52,52,0,0,36,0,0,0,0,0,0,0,52,48,0,
0,32,0,0,0,0,0,0,0,56,44,0,0,32,0,0,0,0,0,0,-0.2,0.0,6.1,-
1.0,0.0,0.0,0.6,2.1,13.6,30.8,0.0,0.0,1.7,-
1.0,0.6,0.0,1.3,1.5,3.7,14.5,0.1,-5.2,1.4,0.0,0.0,0.0,0.8,-0.6,-10.7,-
15.6,0.4,-3.9,0.0,0.0,0.0,0.0,-0.8,-1.7,-10.1,-22.0,0.0,0.0,5.7,-
1.0,0.0,0.0,-0.1,1.2,14.1,22.5,0.0,-2.5,0.8,0.0,0.0,0.0,1.0,0.4,-4.8,-
2.7,0.1,-6.0,0.0,0.0,0.0,0.0,-0.8,-0.6,-24.0,-29.7,0.0,0.0,2.0,-
6.4,0.0,0.0,0.2,2.9,-12.6,15.2,-0.1,0.0,8.4,-10.0,0.0,0.0,0.6,5.9,-
3.9,52.7,-0.3,0.0,15.2,-8.4,0.0,0.0,0.9,5.1,17.7,70.7,-0.4,0.0,13.5,-
4.0,0.0,0.0,0.9,3.9,25.5,62.9,-0.3,0.0,9.0,-
0.9,0.0,0.0,0.9,2.9,23.3,49.4,8
56,1,165,64,81,174,401,149,39,25,37,-
17,31,?,53,0,48,0,0,0,24,0,0,0,0,0,0,64,0,0,0,24,0,0,0,0,0,0,32,24,0,
0,0,40,0,0,0,0,0,0,48,0,0,0,0,0,0,0,0,0,0,0,44,20,0,0,24,0,0,0,0,0,0,
0,60,0,0,0,20,0,0,0,0,0,0,0,24,52,0,0,16,0,0,0,0,0,0,0,32,52,0,0,20,0,0
,0,0,0,0,0,44,48,0,0,32,0,0,0,0,0,0,0,48,44,0,0,32,0,0,0,0,0,0,0,48,40,
0,0,28,0,0,0,0,0,0,0,48,0,0,0,28,0,0,0,0,0,0,0,-
0.6,0.0,7.2,0.0,0.0,0.0,0.4,1.5,17.2,26.5,0.0,0.0,5.5,0.0,0.0,0.0,0.1,1
.7,17.6,29.5,0.3,-1.6,0.9,0.0,0.0,0.0,-0.3,0.4,-1.5,1.3,0.1,-
6.4,0.0,0.0,0.0,0.0,-0.3,-1.6,-15.3,-25.5,-0.3,0.0,4.2,-
0.9,0.0,0.0,0.4,0.7,8.3,12.3,0.2,0.0,2.2,0.0,0.0,0.0,-
0.2,0.8,6.6,11.7,0.4,0.0,1.0,-8.8,0.0,0.0,0.5,-0.6,-21.6,-
26.8,0.4,0.0,2.6,-7.9,0.0,0.0,0.8,2.0,-16.4,1.2,0.0,0.0,5.8,-
7.7,0.0,0.0,0.9,3.8,-5.7,27.7,-0.2,0.0,9.5,-
5.0,0.0,0.0,0.5,2.6,11.8,34.6,-0.4,0.0,11.0,-
2.4,0.0,0.0,0.4,2.6,21.6,43.4,-
0.5,0.0,8.5,0.0,0.0,0.0,0.2,2.1,20.4,38.8,6
54,0,172,95,138,163,386,185,102,96,34,70,66,23,75,0,40,80,0,0,24,0,0,0,
0,0,0,20,56,52,0,0,40,0,0,0,0,0,0,28,116,0,0,0,52,0,0,0,0,0,0,52,64,0,0
,0,88,0,0,0,0,0,0,0,36,92,0,0,24,0,0,0,0,0,0,0,128,0,0,0,24,0,1,0,0,0,0
,0,24,36,76,0,100,0,0,0,0,0,0,0,40,28,60,0,96,0,0,0,0,0,0,0,48,20,56,24
,32,0,0,0,0,0,0,0,44,88,0,0,28,0,0,0,0,0,0,0,44,76,0,0,28,0,0,0,0,0,0,0
,44,72,0,0,24,0,0,0,0,0,0,1.0,0.0,4.5,-2.8,0.0,0.0,0.3,2.5,-
2.2,19.8,0.8,-0.4,6.4,-1.3,0.0,0.0,0.7,2.7,14.2,37.9,-0.2,-
0.6,4.4,0.0,0.0,0.0,0.5,0.2,24.7,26.2,-1.0,-5.3,1.8,0.0,0.0,0.0,-0.5,-
2.5,-8.0,-28.5,0.5,0.0,1.7,-2.7,0.0,0.0,-0.2,1.0,-9.4,-
1.2,0.4,0.0,4.9,0.0,0.0,0.0,0.6,1.4,31.3,42.7,-0.8,0.0,0.7,-
3.8,6.5,0.0,0.3,-3.3,18.7,-13.6,-0.9,0.0,2.2,-4.1,7.4,0.0,0.5,-
2.4,20.9,-2.6,0.0,0.0,5.8,-4.1,4.0,-
0.5,0.4,0.3,20.4,23.3,0.7,0.0,10.0,-5.7,0.0,0.0,0.5,2.2,-
3.0,20.7,1.3,0.0,11.1,-3.4,0.0,0.0,0.4,3.4,11.5,48.2,0.9,0.0,9.5,-
2.4,0.0,0.0,0.3,3.4,12.3,49.0,10
55,0,175,94,100,202,380,179,143,28,11,-
5,20,?,71,0,72,20,0,0,48,0,0,0,0,0,0,64,36,0,0,36,0,0,0,0,0,0,20,52,4
8,0,0,56,0,0,0,0,0,0,64,32,0,0,0,72,0,0,0,0,0,0,0,60,12,0,0,44,0,0,0,0,
0,0,0,60,44,0,0,32,0,0,0,0,0,0,56,0,0,0,0,0,0,0,0,0,0,0,0,40,44,0,0,20,
0,0,0,0,0,0,0,52,40,0,0,32,0,0,0,0,0,0,0,56,48,0,0,36,0,0,0,0,0,0,0,60,
48,0,0,36,0,0,0,0,0,0,0,64,40,0,0,40,0,0,0,0,0,0,0.9,0.0,7.8,-
0.7,0.0,0.0,1.1,1.9,27.3,45.1,0.1,0.0,9.1,-
2.6,0.0,0.0,0.4,1.5,24.5,36.8,-0.4,-0.4,1.6,-2.2,0.0,0.0,-1.0,-0.9,-
1.5,-9.2,-0.4,-8.2,1.8,0.0,0.0,0.0,-0.7,-1.7,-23.4,-35.6,0.9,0.0,3.2,-
0.4,0.0,0.0,0.7,1.2,9.4,18.0,-0.1,0.0,5.1,-
2.5,0.0,0.0,0.3,0.6,9.8,12.6,1.6,-6.5,0.0,0.0,0.0,0.0,-0.4,-0.4,-18.2,-
22.4,2.1,0.0,1.2,-6.9,0.0,0.0,-0.5,2.9,-12.7,18.0,0.7,0.0,9.0,-

```
7.9,0.0,0.0,0.1,4.1,7.6,51.0,0.4,0.0,15.0,-
5.5,0.0,0.0,0.1,3.3,28.8,63.1,0.1,0.0,15.2,-
3.7,0.0,0.0,0.6,3.0,36.8,68.0,0.1,0.0,12.2,-
2.2,0.0,0.0,0.4,2.6,34.6,61.6,1
75,0,190,80,88,181,360,177,103,-
16,13,61,3,?,?,0,48,40,0,0,28,0,0,0,0,0,0,40,24,0,0,24,0,0,0,0,0,0,52
,36,0,0,0,60,0,0,0,0,0,0,48,28,0,0,0,56,0,0,0,0,0,0,48,36,0,0,28,0,0,
0,0,0,0,44,0,0,0,0,0,0,0,0,0,0,88,0,0,0,0,0,0,0,0,0,0,0,40,52,0,0,2
8,0,0,0,0,0,0,0,48,48,0,0,32,0,0,0,0,0,0,48,52,0,0,32,0,0,0,0,0,0,0,5
2,44,0,0,28,0,0,0,0,0,0,52,48,0,0,32,0,0,0,0,0,0,0.0,0.0,5.2,-
1.4,0.0,0.0,0.9,2.3,9.6,31.6,0.1,0.0,1.6,-
0.5,0.0,0.0,1.9,1.7,2.6,18.9,0.2,-3.8,1.2,0.0,0.0,0.0,1.0,-0.6,-7.7,-
13.4,-0.1,-3.4,0.8,0.0,0.0,0.0,-1.4,-1.5,-7.0,-17.8,-0.1,0.0,4.4,-
1.3,0.0,0.0,-0.1,1.1,8.2,16.5,0.6,-1.6,0.0,0.0,0.0,0.0,1.4,0.3,-3.5,-
1.9,0.0,-5.7,0.0,0.0,0.0,0.0,-0.4,-0.5,-25.0,-30.0,-0.2,0.0,1.6,-
6.0,0.0,0.0,-0.7,2.1,-12.4,8.6,-0.5,0.0,8.5,-10.2,0.0,0.0,-1.0,4.7,-
4.0,43.0,-0.2,0.0,15.2,-7.8,0.0,0.0,-0.1,4.9,16.2,63.2,-0.2,0.0,9.1,-
0.9,0.0,0.0,-0.2,2.9,21.7,48.9,-0.4,0.0,13.1,-3.6,0.0,0.0,-
0.1,3.9,25.4,62.8,7
13,0,169,51,100,167,321,174,91,107,66,52,88,?,84,0,36,48,0,0,20,0,0,0,0,0
,0,0,20,44,36,0,0,44,0,0,0,0,0,0,24,64,0,0,0,48,0,0,0,0,0,0,44,36,0,0,0
,52,0,0,0,0,0,0,0,28,64,0,0,16,0,0,0,0,0,0,24,44,40,0,0,44,0,0,0,0,0,0,
0,36,60,0,0,24,0,0,0,0,0,0,20,32,60,0,0,40,0,0,0,0,0,0,24,32,60,0,0,44,
0,0,0,0,0,0,0,52,40,0,0,36,0,0,0,0,0,0,0,44,40,0,0,32,0,0,0,0,0,0,20,36
,56,0,0,40,0,0,0,0,0,0,0.5,0.0,2.7,-6.4,0.0,0.0,0.9,1.7,-10.5,7.1,0.1,-
1.2,19.1,-2.3,0.0,0.0,1.4,4.3,36.7,84.8,-0.4,-
2.3,21.7,0.0,0.0,0.0,0.7,2.6,66.7,95.8,-0.2,-9.0,3.2,0.0,0.0,0.0,-1.1,-
2.9,-14.1,-39.0,0.5,0.0,1.8,-12.9,0.0,0.0,0.4,-0.4,-38.7,-42.1,-0.1,-
1.6,19.9,-0.7,0.0,0.0,1.0,3.3,40.4,65.4,0.4,0.0,6.7,-24.4,0.0,0.0,-
1.2,0.4,-61.2,-59.9,0.9,-0.5,11.9,-43.3,0.0,0.0,0.8,3.4,-111.4,-
95.1,2.0,-0.8,19.8,-48.4,0.0,0.0,1.6,8.7,-114.5,-72.8,2.0,0.0,31.0,-
25.7,0.0,0.0,0.8,5.9,29.2,85.8,0.6,0.0,19.5,-
11.4,0.0,0.0,0.8,3.3,20.1,49.1,0.0,-0.6,12.2,-
2.8,0.0,0.0,0.9,2.2,13.5,31.1,14
40,1,160,52,77,129,377,133,77,77,49,75,65,?,70,0,44,0,0,0,24,0,0,0,0,0,
0,0,40,32,0,0,24,0,0,0,0,0,0,0,44,28,0,0,24,0,0,0,0,0,0,44,16,0,0,0,48,
0,0,0,0,0,0,36,0,0,0,0,0,0,0,0,0,0,0,0,44,16,0,0,24,0,0,0,0,0,0,0,20,56
,0,0,12,0,0,0,0,0,0,0,24,56,0,0,16,0,0,0,0,0,0,0,36,48,0,0,24,0,0,0,0,0
,0,0,40,44,0,0,28,0,0,0,0,0,0,0,40,44,0,0,24,0,0,0,0,0,0,0,44,0,0,0,24,
0,0,0,0,0,0,-0.5,0.0,1.8,0.0,0.0,0.0,0.2,1.0,3.9,10.5,-0.1,0.0,7.6,-
1.1,0.0,0.0,0.5,1.4,13.5,22.7,0.0,0.0,5.9,-
0.5,0.0,0.0,0.3,0.6,12.2,15.0,0.1,-4.6,0.6,0.0,0.0,0.0,-0.4,-0.9,-9.7,-
14.7,0.2,-2.1,0.0,0.0,0.0,0.0,-0.3,0.4,-3.7,-1.4,-0.2,0.0,6.8,-
0.9,0.0,0.0,0.7,0.7,14.2,17.1,1.3,0.0,1.3,-11.5,0.0,0.0,-0.3,1.7,-
30.9,-13.9,1.7,0.0,2.3,-17.5,0.0,0.0,-0.6,4.5,-46.3,-1.3,1.1,0.0,3.7,-
11.0,0.0,0.0,-0.5,4.1,-19.8,21.2,0.1,0.0,7.7,-
6.4,0.0,0.0,0.4,1.9,1.4,15.4,0.0,0.0,7.4,-
2.5,0.0,0.0,0.4,1.3,9.3,18.9,-
0.4,0.0,6.5,0.0,0.0,0.0,0.4,1.0,14.3,20.5,1
49,1,162,54,78,0,376,157,70,67,7,8,51,?,67,0,44,36,0,0,24,0,0,0,0,0,0,0
,52,32,0,0,28,0,0,0,0,0,0,0,56,28,0,0,24,0,0,0,0,0,0,48,32,0,0,0,56,0,0
,0,0,0,0,52,0,0,0,0,0,0,0,0,0,0,0,0,52,28,0,0,28,0,0,0,0,0,0,0,20,44,0,
0,8,0,0,0,0,0,0,0,0,24,48,0,0,16,0,0,0,0,0,0,0,36,44,0,0,24,0,0,0,0,0,0,0
,44,48,0,0,28,0,0,0,0,0,0,0,48,44,0,0,28,0,0,0,0,0,0,0,48,40,0,0,24,0,0
,0,0,0,0,-0.3,0.0,4.1,-1.1,0.0,0.0,0.8,1.0,7.1,13.7,-0.3,0.0,8.4,-
1.5,0.0,0.0,0.6,0.7,19.4,22.9,0.0,0.0,4.4,-0.8,0.0,0.0,-0.3,-
0.6,11.2,6.9,0.1,-6.3,1.3,0.0,0.0,0.0,-0.6,-0.8,-13.1,-17.9,0.1,-
0.8,0.0,0.0,0.0,0.0,0.6,0.7,-2.0,2.9,-0.2,0.0,6.3,-
```

1.2,0.0,0.0,0.2,0.3,14.7,16.8,0.7,0.0,0.5,-7.3,0.0,0.0,0.2,-0.1,-15.5,-
16.4,0.9,0.0,0.7,-8.9,0.0,0.0,0.6,2.5,-20.5,4.0,0.8,0.0,2.1,-
9.0,0.0,0.0,0.6,3.8,-16.1,21.1,0.1,0.0,6.6,-
4.1,0.0,0.0,0.3,1.4,4.7,14.2,-0.2,0.0,8.5,-
2.7,0.0,0.0,0.1,0.8,14.5,20.9,-0.3,0.0,8.2,-
1.9,0.0,0.0,0.1,0.5,15.8,19.8,1
44,0,168,56,84,118,354,160,63,61,69,78,66,84,64,0,40,0,0,0,20,0,0,0,0,0
,0,0,44,12,0,0,28,0,0,0,0,0,0,0,36,8,0,0,20,0,0,0,0,0,0,40,12,0,0,0,44,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,36,12,0,0,20,0,0,0,0,0,0,0,20,56,
0,0,12,0,0,0,0,0,0,24,48,0,0,12,0,0,0,0,0,0,28,44,0,0,16,0,0,0,0,0,
0,0,44,32,0,0,32,0,0,0,0,0,0,44,28,0,0,28,0,0,0,0,0,0,0,40,24,0,0,24,
0,0,0,0,0,0,0.1,0.0,2.3,0.0,0.0,0.0,0.4,1.0,4.6,11.6,1.2,0.0,5.4,-
0.7,0.0,0.0,1.8,2.8,11.4,31.0,1.1,0.0,3.0,-
0.4,0.0,0.0,1.4,1.8,5.3,17.9,-0.7,-3.9,0.5,0.0,0.0,0.0,-1.1,-1.9,-7.5,-
20.4,-0.5,0.0,0.0,0.0,0.0,0.0,-0.6,-0.5,0.0,-3.4,1.1,0.0,4.2,-
0.5,0.0,0.0,1.6,2.3,7.2,22.8,0.5,0.0,0.9,-5.5,0.0,0.0,-0.7,1.0,-14.5,-
5.3,0.7,0.0,1.2,-6.4,0.0,0.0,-0.5,2.6,-13.9,10.0,1.5,0.0,2.4,-
10.3,0.0,0.0,0.3,6.8,-19.3,43.2,0.8,0.0,7.9,-
7.3,0.0,0.0,0.9,6.5,5.7,62.9,0.1,0.0,9.3,-
3.8,0.0,0.0,0.8,3.8,15.1,48.5,0.1,0.0,7.0,-
1.3,0.0,0.0,0.6,2.1,12.5,30.9,1
50,1,167,67,89,130,383,156,73,85,34,70,71,?,63,0,44,40,0,0,28,0,0,0,0,0
,0,0,56,24,0,0,32,0,0,0,0,0,0,0,72,0,0,0,28,0,0,0,0,0,0,56,28,0,0,0,60,
0,0,0,0,0,0,0,28,56,0,0,16,0,0,0,0,0,0,60,0,0,0,32,0,0,0,0,0,0,0,24,3
6,32,0,68,0,0,0,0,0,0,0,36,44,0,0,20,0,0,0,0,0,0,0,40,48,0,0,24,0,0,0,0
,0,0,0,56,40,0,0,40,0,0,0,0,0,0,0,52,36,0,0,32,0,0,0,0,0,0,20,44,36,0,0
,44,0,0,0,0,0,0,-0.1,0.0,3.5,-
2.0,0.0,0.0,0.4,1.3,3.7,13.5,0.0,0.0,9.9,-
0.8,0.0,0.0,1.2,1.2,26.8,35.2,0.0,0.0,8.3,0.0,0.0,0.0,0.8,0.3,29.8,32.0
,0.1,-6.1,1.1,0.0,0.0,0.0,-0.6,-1.2,-15.5,-24.1,0.0,0.0,0.6,-
4.1,0.0,0.0,-0.1,0.8,-10.6,-4.9,-
0.2,0.0,8.9,0.0,0.0,0.0,0.8,0.7,26.7,30.2,0.1,0.0,1.3,-
5.4,1.9,0.0,0.2,0.8,-5.2,2.1,0.8,0.0,4.4,-8.5,0.0,0.0,0.8,3.9,-
10.8,25.0,0.4,0.0,4.3,-7.3,0.0,0.0,1.1,4.0,-8.9,27.9,-0.5,0.0,7.0,-
3.2,0.0,0.0,1.1,1.3,13.2,22.3,-0.5,0.0,10.9,-
2.5,0.0,0.0,1.0,1.0,23.8,29.6,-0.5,-0.6,10.8,-
1.7,0.0,0.0,0.8,0.9,20.1,25.1,10
62,0,170,72,102,135,401,156,83,72,71,68,72,?,70,20,36,48,0,0,36,0,0,0,0,0
,0,0,0,52,0,0,0,28,0,0,0,0,0,0,0,104,0,0,0,36,0,0,0,0,0,0,40,36,0,0,0,4
8,0,0,0,0,0,0,28,24,40,0,0,40,0,0,0,0,0,0,0,84,0,0,0,28,0,0,0,0,0,0,0,2
4,64,0,0,12,0,0,0,0,0,0,20,16,56,0,0,28,0,0,0,0,0,0,24,28,52,0,0,40,0,0
,0,0,0,0,24,32,56,0,0,40,0,0,0,0,0,0,20,36,56,0,0,40,0,0,0,0,0,0,20,40,
52,0,0,40,0,0,0,0,0,0,0,-0.4,-0.5,5.8,-1.9,0.0,0.0,0.8,0.4,5.4,8.4,-
0.8,0.0,7.4,0.0,0.0,0.0,1.7,1.4,19.2,29.2,0.0,0.0,1.6,0.0,0.0,0.0,1.0,1
.1,8.3,16.6,0.1,-5.7,1.3,0.0,0.0,0.0,-1.0,-1.0,-9.1,-15.5,-0.5,-
0.6,1.5,-1.9,0.0,0.0,-0.4,-0.2,-2.8,-
4.1,0.4,0.0,4.4,0.0,0.0,0.0,1.3,1.0,18.4,25.2,0.7,0.0,0.6,-
8.5,0.0,0.0,-0.1,-1.4,-26.5,-39.9,0.1,-0.8,2.0,-12.9,0.0,0.0,-0.2,-
2.7,-35.3,-54.7,-0.9,-2.5,11.9,-18.6,0.0,0.0,-0.2,-5.2,-34.7,-76.3,-
1.4,-1.7,16.7,-13.3,0.0,0.0,0.9,-3.1,-12.5,-37.3,-0.9,-0.8,12.7,-
5.7,0.0,0.0,0.8,-0.3,6.1,3.7,-0.4,-0.5,9.0,-
2.0,0.0,0.0,0.8,0.9,12.3,19.3,3
45,1,165,86,77,143,373,150,65,12,37,49,26,?,72,0,40,28,0,0,20,0,0,0,0,0
,0,0,40,20,0,0,20,0,0,0,0,0,0,32,44,0,0,0,36,0,0,0,0,0,0,40,28,0,0,0,48
,0,0,0,0,0,0,0,40,28,0,0,20,0,0,0,0,0,0,12,24,16,0,0,20,0,0,0,0,0,0,0,2
0,44,0,0,12,0,0,0,0,0,0,0,36,48,0,0,24,0,0,0,0,0,0,0,40,48,0,0,24,0,0,0
,0,0,0,0,36,56,0,0,24,0,0,0,0,0,0,0,40,52,0,0,28,0,0,0,0,0,0,0,40,36,0,
0,24,0,0,0,0,0,0,0.4,0.0,6.1,-

1.7,0.0,0.0,0.8,1.6,9.9,22.7,0.1,0.0,4.5,-
1.3,0.0,0.0,1.2,1.8,7.7,21.7,0.0,-2.6,1.1,0.0,0.0,0.0,0.5,0.2,-1.7,-
0.6,-0.1,-5.3,1.6,0.0,0.0,0.0,-1.1,-1.8,-8.4,-23.1,0.1,0.0,4.6,-
0.9,0.0,0.0,0.3,0.7,8.0,13.7,0.1,-0.5,1.3,-
0.7,0.0,0.0,0.8,1.2,0.7,10.5,0.1,0.0,0.6,-4.4,0.0,0.0,-0.4,-1.8,-9.0,-
27.3,0.1,0.0,3.3,-6.7,0.0,0.0,0.2,-0.9,-10.1,-17.4,0.0,0.0,6.5,-
9.1,0.0,0.0,0.5,0.4,-8.8,-5.5,0.0,0.0,3.8,-5.6,0.0,0.0,0.5,0.3,-8.8,-
6.4,0.0,0.0,4.8,-4.3,0.0,0.0,0.6,0.9,-1.5,5.7,0.1,0.0,4.4,-
2.2,0.0,0.0,0.5,1.5,4.9,17.2,1
54,1,172,58,78,155,382,163,81,-24,42,41,-
13,?,73,0,72,0,0,0,24,0,0,0,0,0,0,0,44,44,0,0,28,0,0,0,0,0,0,80,0,0,0,0
,0,0,0,0,0,0,0,44,36,0,0,0,48,0,0,0,0,0,0,0,84,0,0,0,28,0,0,0,0,0,0,72,
0,0,0,0,0,0,0,0,0,0,0,0,24,36,28,0,72,0,0,0,0,0,0,0,36,32,0,0,24,0,0,0,
0,0,0,0,44,44,0,0,28,0,0,0,0,0,0,0,48,52,0,0,32,0,0,0,0,0,0,0,44,52,0,0
,32,0,0,0,0,0,0,0,44,48,0,0,28,0,0,0,0,0,0,-
0.2,0.0,6.3,0.0,0.0,0.0,0.8,0.7,22.6,26.9,-0.1,0.0,3.7,-
2.6,0.0,0.0,0.9,0.8,2.4,7.2,0.0,-3.5,0.0,0.0,0.0,0.0,0.4,0.1,-14.0,-
13.3,0.2,-5.2,0.6,0.0,0.0,0.0,-1.1,-0.6,-10.4,-
14.6,0.0,0.0,5.1,0.0,0.0,0.0,0.6,0.1,21.4,22.3,-0.2,-
2.9,0.0,0.0,0.0,0.0,0.7,0.4,-10.4,-7.5,0.6,0.0,0.5,-5.4,1.3,0.0,-0.8,-
0.4,-7.3,-11.4,1.0,0.0,2.3,-3.7,0.0,0.0,0.1,0.6,-1.8,4.4,-0.1,0.0,5.1,-
5.8,0.0,0.0,0.5,1.0,-1.5,7.1,-0.2,0.0,8.2,-
5.8,0.0,0.0,0.8,1.1,4.6,13.4,-0.3,0.0,7.1,-
4.2,0.0,0.0,0.8,0.5,4.7,8.2,-0.2,0.0,6.3,-
2.1,0.0,0.0,0.8,0.5,8.8,12.1,10
30,0,170,73,91,180,355,157,104,68,51,60,63,?,56,0,92,0,0,0,32,0,0,0,0,0
,0,28,48,20,0,0,52,0,0,0,0,0,0,36,40,0,0,0,52,0,0,0,0,0,0,56,0,0,0,0,0,
0,0,0,0,0,0,0,40,36,0,0,28,0,0,0,0,0,0,32,44,0,0,0,52,0,0,0,0,0,0,0,44,
44,0,0,28,0,0,0,0,0,0,0,48,24,0,0,32,0,0,0,0,0,0,0,52,36,0,0,36,0,0,0,0
,0,0,0,60,12,20,0,36,0,0,0,0,0,0,24,52,12,0,0,52,0,0,0,0,0,0,24,48,0,0,
0,48,0,0,0,0,0,0,0.2,0.0,3.2,0.0,0.0,0.0,0.4,1.1,14.7,21.9,0.0,-
1.3,14.6,-0.4,0.0,0.0,0.8,1.7,32.8,44.0,-0.1,-
2.6,13.1,0.0,0.0,0.0,0.4,0.6,21.6,25.5,-0.1,-8.0,0.0,0.0,0.0,0.0,-0.6,-
1.4,-22.4,-31.0,0.1,0.0,2.3,-5.7,0.0,0.0,-0.1,0.2,-5.6,-4.5,-0.1,-
1.9,13.9,0.0,0.0,0.0,0.6,1.1,27.5,34.3,0.8,0.0,3.3,-7.4,0.0,0.0,0.2,-
0.4,-9.0,-12.3,2.8,0.0,8.9,-
8.0,0.0,0.0,0.3,1.3,11.7,22.6,1.5,0.0,11.4,-
3.7,0.0,0.0,0.5,4.8,23.0,63.3,0.0,0.0,20.6,-
2.6,0.8,0.0,0.6,5.4,61.1,103.2,-0.1,-1.1,19.0,-
1.4,0.0,0.0,0.5,3.4,47.3,79.9,-0.6,-
0.9,12.3,0.0,0.0,0.0,0.4,2.1,28.5,48.6,6
44,1,160,88,77,158,399,163,94,46,20,45,40,?,72,0,80,0,0,0,28,0,0,0,0,0,
0,20,72,0,0,0,44,0,0,0,0,0,0,24,64,0,0,0,52,0,0,0,1,0,0,80,0,0,0,0,0,0,
0,0,0,0,0,0,36,36,0,0,20,0,0,0,0,0,0,0,80,0,0,0,32,0,0,0,0,0,0,0,28,48,
0,0,20,0,0,1,0,0,0,0,36,36,0,0,20,0,0,0,0,0,0,0,52,20,0,0,32,0,0,0,0,0,
0,0,48,12,0,0,24,0,0,0,0,0,0,0,16,44,12,0,0,36,0,0,0,0,0,0,16,64,0,0,0,40
,0,0,0,0,0,0,0.2,0.0,10.2,0.0,0.0,0.0,0.6,2.3,40.8,58.2,0.0,-
0.5,12.9,0.0,0.0,0.0,0.9,2.0,45.9,61.1,-0.2,-0.6,4.2,0.0,0.0,0.0,0.7,-
0.7,12.7,7.4,-0.1,-11.5,0.0,0.0,0.0,0.0,-0.9,-2.0,-46.0,-59.2,-
0.2,0.0,3.3,-
0.9,0.0,0.0,0.7,1.0,4.3,9.9,0.4,0.0,8.4,0.0,0.0,0.0,0.5,1.0,33.6,41.4,0
.6,0.0,3.4,-10.4,0.0,0.0,-0.8,-0.3,-20.2,-23.2,1.2,0.0,5.6,-
5.3,0.0,0.0,-0.5,3.4,0.5,35.1,0.8,0.0,8.3,-
2.5,0.0,0.0,0.4,3.3,19.0,52.6,0.2,0.0,12.7,-
1.1,0.0,0.0,0.3,1.9,29.8,47.2,0.2,-0.5,13.6,-
0.4,0.0,0.0,0.2,1.9,29.3,46.4,0.1,-
0.6,12.4,0.0,0.0,0.0,0.3,1.7,39.2,54.1,1

47,1,150,48,75,132,350,169,65,36,45,68,40,?,76,0,48,0,0,0,24,0,0,0,0,0,
0,0,44,28,0,0,28,0,0,0,0,0,0,0,40,40,0,0,24,0,0,0,0,0,0,40,32,0,0,0,44,
0,0,0,0,0,0,0,60,0,0,0,20,0,0,0,0,0,0,0,40,36,0,0,24,0,0,0,0,0,0,0,20,4
4,0,0,12,0,0,0,0,0,0,0,28,44,0,0,20,0,0,0,0,0,0,0,48,20,0,0,20,0,0,0,0,
0,0,0,48,24,0,0,28,0,0,0,0,0,0,0,48,36,0,0,28,0,0,0,0,0,0,0,48,32,0,0,2
8,0,0,0,0,0,0,0.1,0.0,5.6,0.0,0.0,0.0,0.3,1.7,13.4,25.9,0.1,0.0,7.9,-
1.8,0.0,0.0,1.2,2.3,14.8,31.8,-0.1,0.0,2.6,-
1.8,0.0,0.0,0.8,0.6,1.6,6.0,0.0,-6.7,1.2,0.0,0.0,0.0,-0.9,-1.8,-11.5,-
24.8,-0.1,0.0,1.7,0.0,0.0,0.0,0.1,0.5,5.1,8.8,0.0,0.0,5.3,-
1.9,0.0,0.0,0.8,1.4,7.2,17.5,1.2,0.0,1.4,-13.7,0.0,0.0,-0.9,0.5,-28.7,-
21.8,1.6,0.0,2.5,-11.9,0.0,0.0,-0.8,4.1,-22.6,33.9,1.5,0.0,5.9,-
3.3,0.0,0.0,0.6,7.0,10.8,107.4,0.1,0.0,9.6,-
2.9,0.0,0.0,0.7,3.1,19.6,45.0,0.0,0.0,9.6,-
1.6,0.0,0.0,0.7,2.6,20.2,41.5,0.0,0.0,7.7,-
0.8,0.0,0.0,0.6,1.7,17.2,31.1,1
47,0,171,59,82,145,347,169,61,77,75,77,75,?,67,0,48,0,0,0,20,0,0,0,0,0,
0,0,52,36,0,0,28,0,0,0,0,0,0,0,52,36,0,0,28,0,0,0,0,0,0,52,32,0,0,0,56,
0,0,0,0,0,0,48,32,0,0,0,60,0,0,0,0,0,0,0,52,32,0,0,28,0,0,0,0,0,0,0,16,
48,20,0,72,0,0,0,0,0,0,0,20,48,20,0,76,0,0,0,0,0,0,0,24,52,0,0,12,0,0,0
,0,0,0,0,56,28,0,0,40,0,0,0,0,0,0,0,52,36,0,0,32,0,0,0,0,0,0,0,48,36,0,
0,28,0,0,0,0,0,0,-0.5,0.0,2.2,0.0,0.0,0.0,0.4,0.7,5.2,10.9,-
0.2,0.0,9.8,-1.7,0.0,0.0,1.4,2.6,22.4,42.6,0.0,0.0,7.5,-
1.2,0.0,0.0,1.2,2.0,17.4,32.6,0.2,-6.0,1.1,0.0,0.0,0.0,-0.9,-1.6,-
13.9,-26.3,0.0,-2.6,0.4,0.0,0.0,0.0,-0.4,-0.7,-5.6,-10.0,-0.1,0.0,8.6,-
1.5,0.0,0.0,1.3,2.2,19.9,36.1,1.3,0.0,0.5,-8.7,1.8,0.0,-0.6,-0.4,-
18.6,-22.2,2.4,0.0,0.8,-9.7,3.0,0.0,-0.4,0.8,-19.4,-12.2,1.6,0.0,1.4,-
11.8,0.0,0.0,0.4,4.4,-29.0,10.6,0.1,0.0,7.6,-
3.4,0.0,0.0,0.4,4.7,16.5,60.6,-0.5,0.0,11.5,-
2.6,0.0,0.0,0.4,2.8,25.3,47.1,-0.4,0.0,9.4,-
1.7,0.0,0.0,0.6,2.3,19.5,41.1,10
46,1,158,58,70,120,353,122,52,57,49,-
2,54,?,70,0,48,0,0,0,24,0,0,0,0,0,0,0,48,0,0,0,28,0,0,0,0,0,0,0,44,12,0
,0,24,0,0,0,0,0,0,48,16,0,0,0,52,0,0,0,0,0,0,0,24,0,0,0,8,0,0,0,0,0,0,0
,48,0,0,0,28,0,0,0,0,0,0,60,0,0,0,0,0,0,0,0,0,1,60,0,0,0,0,0,0,0,0,0,
0,0,0,32,44,0,0,16,0,0,0,0,0,0,0,48,40,0,0,32,0,0,0,0,0,0,0,48,36,0,0,2
8,0,0,0,0,0,0,0,52,0,0,0,28,0,0,0,0,0,0,-
0.5,0.0,3.2,0.0,0.0,0.0,0.2,0.5,7.6,10.5,0.3,0.0,6.0,0.0,0.0,0.0,0.1,0.
7,14.4,18.3,0.6,0.0,3.0,-0.4,0.0,0.0,-0.2,0.2,6.4,7.3,0.0,-
4.6,0.4,0.0,0.0,0.0,-0.4,-0.6,-10.7,-14.1,-
0.5,0.0,0.8,0.0,0.0,0.0,0.3,0.1,0.9,1.4,0.4,0.0,4.4,0.0,0.0,0.0,0.1,0.4
,10.5,12.6,0.6,-10.4,0.0,0.0,0.0,0.0,0.2,0.5,-31.2,-24.4,0.1,-
6.7,0.0,0.0,0.0,0.0,0.3,0.6,-20.1,-12.1,0.1,0.0,1.7,-
4.1,0.0,0.0,0.5,1.3,-6.3,11.3,-0.4,0.0,8.4,-
2.9,0.0,0.0,0.4,0.9,14.3,20.2,-0.4,0.0,8.4,-
1.8,0.0,0.0,0.3,0.5,16.9,19.6,-
0.6,0.0,6.6,0.0,0.0,0.0,0.3,0.7,17.1,20.8,1
73,0,165,63,91,154,392,175,83,73,-
24,61,42,?,66,0,44,56,0,0,20,0,0,0,0,0,0,0,84,0,0,0,28,0,0,0,0,0,0,16,7
2,0,0,0,44,0,0,0,0,0,0,76,0,0,0,0,0,0,0,0,0,0,0,36,40,0,0,12,0,0,0,0,
0,0,0,68,0,0,0,32,0,0,0,0,0,0,72,0,0,0,0,0,0,0,0,0,0,0,24,64,0,0,16,0
,0,0,0,0,0,0,40,52,0,0,28,0,0,0,0,0,0,0,44,52,0,0,32,0,0,0,0,0,0,0,48,5
2,0,0,28,0,0,0,0,0,0,0,64,0,0,0,28,0,0,0,0,0,0,0.0,0.0,2.4,-
0.8,0.0,0.0,0.7,1.0,3.0,11.6,