

Mode Choice Modeling Using Artificial Neural Networks

by
Praveen Kumar Edara

Thesis submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Civil Engineering

Dušan Teodorović, Chair
John Collura
Antonio Trani

October 2003
Falls Church, Virginia

Key words: Mode choice, Neural networks, Multinomial logit, Linear regression, Clustering, American travel survey database

ABSTRACT

Artificial intelligence techniques have produced excellent results in many diverse fields of engineering. Techniques such as neural networks and fuzzy systems have found their way into transportation engineering. In recent years, neural networks are being used instead of regression techniques for travel demand forecasting purposes. The basic reason lies in the fact that neural networks are able to capture complex relationships and learn from examples and also able to adapt when new data become available. The primary goal of this thesis is to develop mode choice models using artificial neural networks and compare the results with traditional mode choice models like the multinomial logit model and linear regression method. The data used for this modeling is extracted from the American Travel Survey data. Data mining procedures like clustering are used to process the extracted data. The results of three models are compared based on residuals and error criteria. It is found that neural network approach produces the best results for the chosen set of explanatory variables. The possible reasons for such results are identified and explained to the extent possible. The three major objectives of this thesis are to: present an approach to handle the data from a survey database, address the mode choice problem using artificial neural networks, and compare the results of this approach with the results of traditional models vis-à-vis logit model and linear regression approach. The results of this research work should encourage more transportation researchers and professionals to consider artificial intelligence tools for solving transportation planning problems.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Dušan Teodorović, for all the encouragement and ideas he has provided during my research. I am greatly indebted to Dr. Antonio Trani and Dr. John Collura for being on my thesis advisory committee and for their support and assistance.

In addition, I thank Dr. Hojong Baik and Junhyuk Kim for the interesting ideas and suggestions they have made which were beneficial to my research.

This research work is done as part of the NEXTOR project, sponsored by the Federal Aviation Administration (FAA). I thank FAA for supporting the project.

I extend my gratitude to Tammy Prailey who has helped me in writing this thesis document.

Finally, I would like to thank my parents and brother for so many things.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	v
LIST OF TABLES	vii
1 INTRODUCTION	1
2 RESEARCH GOAL AND OBJECTIVES	4
3 MOTIVATION FOR RESEARCH	5
4 THESIS ORGANIZATION	6
5 LITERATURE REVIEW	7
5.1 Introduction.....	7
5.2 Logit model.....	7
5.3 Linear regression approach.....	9
5.4 Artificial neural networks approach.....	11
6 DESCRIPTION OF NEURAL NETWORKS	12
6.1 Introduction.....	12
6.2 Characteristics of neural networks.....	12
6.3 Training of a neural network.....	15
6.4 Sample calculations	16
6.5 Generalization	18
6.6 Testing of a neural network	18
7 MODELING	19
7.1 Data extraction.....	20
7.2 Data processing.....	22
7.2.1 K-means clustering procedure	23
7.3 Development of mode choice models.....	25
7.3.1 Artificial neural networks approach.....	25
7.3.2 Multinomial logit model	28
7.3.3 Linear regression method.....	28
7.3.4 Comparison of models	29
8 RESULTS	30
9 CONCLUSIONS	41
9.1 Comments on using neural networks.....	42
9.2 Scope for future research	43
10 REFERENCES	44
11 APPENDICES	46
11.1 Appendix A – GLOSSARY OF TERMS	46
11.2 Appendix B – MATLAB CODES	47
11.3 Appendix C- SAS OUTPUTS	62
12 VITA	64

LIST OF FIGURES

Figure 1:	Four step planning procedure	1
Figure 2:	Processing element: artificial neuron with activation function	13
Figure 3:	Two-layered feedforward neural network	14
Figure 4:	Taxonomy of training a multilayered perceptron: the input signal extends- forward and the computed error backward.	16
Figure 5 :	Sample calculation of the output	17
Figure 6:	Overview of the Mode choice modeling procedure	19
Figure 7:	Data Processing step	23
Figure 8:	Best Neural network model	27
Figure 9:	Commercial airline mode - <i>Comparison of models-</i> <i>based on model probability vs real probability-</i> a) Neural network model b) Logit model c) Linear regression model.	32
Figure 10:	Automobile mode - <i>Comparison of models-</i> <i>based on model probability vs real probability-</i> a) Neural network model b) Logit model c) Linear regression model.	34
Figure 11:	General Aviation mode - <i>Comparison of models-</i> <i>based on model probability vs real probability-</i> a) Neural network model b) Logit model c) Linear regression model.	36
Figure 12:	Commercial airline mode - <i>Comparison of models-</i> <i>based on Absolute Residuals-</i> a) Neural network model b) Logit model c) Linear regression model.	38

Figure 13: Automobile mode - *Comparison of models-* 39
based on Absolute Residuals – a) Neural network model b) Logit model c)
Linear regression model.

Figure 14: General aviation mode - *Comparison of models-* 40
based on Absolute Residuals – a) Neural network model b) Logit model c)
Linear regression model.

LIST OF TABLES

Table 1:	Travel Time Assumptions	21
Table 2:	Travel Cost Assumptions	21
Table 3:	Comparison of the models- neural networks vs. multinomial logit vs. linear regression- <i>mean square error and average relative variance</i>	31

1 INTRODUCTION

Travel demand forecasting is an essential aspect of transportation planning. Given the land-use pattern, socioeconomic, and environmental conditions, travel demand can be expressed as the number of persons or vehicles per unit time that can be expected to travel on a given segment of a transportation system. There is a need to forecast the travel demand in order to estimate the vehicular volume for the future, based on which various transportation system alternatives can be designed or modified. The methods for forecasting travel demand can range from a simple extrapolation of observed trends to a sophisticated computerized process involving extensive data gathering and mathematical modeling. The travel demand forecasting process is as much an art as it is a science because judgments are required concerning the various parameters; that is, population, car ownership, and so forth that provide the basis for a travel forecast. In general, transportation researchers have identified three main factors that affect the demand for urban travel: the location and intensity of land use, the socioeconomic characteristics of people living in the area, and the extent, cost, and quality of available transportation services. These factors are incorporated in most travel forecasting procedures.

(Garber and Hoel (2002))

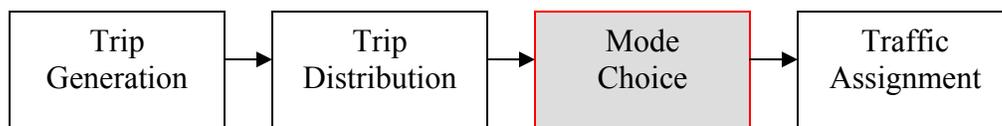


Figure 1: Four step planning procedure

The traditional four-step planning procedure, as shown in Figure 1 has four main steps involved- *Trip Generation, Trip Distribution, Mode Choice, and Traffic Assignment*. In

the trip generation, the number of trips generating from all the zones are modeled. Trips generated by the zones consist of two types: productions and attractions. Once the total trips generated by each zone are estimated, the trip distribution procedure calculates the trip interchanges between the zones. In the mode choice step, mode specific trip interchanges are estimated. This thesis is related to the mode choice modeling. Mode choice can be defined as that aspect of the demand analysis process that determines the number (or percentage) of trips between zones that are made by different types of available modes. The selection of one mode or another is a complex process that depends on factors such as the traveler's income; the availability of transit service or auto ownership; and the relative advantages of each mode in terms of travel time, cost, comfort, convenience, and safety. Mode choice models attempt to replicate the relevant characteristics of the traveler, the transportation system, and the trip itself, such that a realistic estimate of the number of trips by each mode for each zonal pair is obtained.

In United States, intercity trips are made using mainly five modes: automobile, commercial airline, bus, rail, and general aviation (privately owned aircraft). If we consider trip distances of over 100 miles, the market shares of bus and the rail mode are significantly low as compared to the market shares of automobile and commercial airline mode; hence, only three competing modes of transportation are available to the trip maker. Two main reasons for determining the demand specific to all modes are: 1) to effective demand management, and 2) to estimate the changes required to be made to the supply. For example, fleet sizes of commercial airlines can be adjusted according to the estimated demand of people traveling by the commercial airline mode. The 1995 American Travel Survey (ATS) that was conducted by the Bureau of Transportation

Statistics (BTS) contains information about the long-distance travel of persons living in the United States. The survey collected quarterly information related to the characteristics of persons, households, and trips of 100 miles or more for approximately 80,000 American households. The ATS data provides detailed information on state-to-state travel as well as travel to and from metropolitan areas by mode of transportation (15). The survey data can be used for developing various demand forecasting models. This paper discusses three different methodologies of developing mode choice models that can be calibrated and validated using the ATS data. In the past, mode choice modeling was done using techniques like linear regression and the logit model. An alternative approach for addressing the mode choice problem is by using artificial intelligence (AI) techniques. This research concentrates on one of the popular AI techniques - *artificial neural networks*. Applications of artificial neural networks have produced excellent results in many areas of research. Artificial intelligence tools like neural networks are not used extensively in solving the transportation problems. This research verifies the applicability of AI tools to travel demand forecasting procedures (mode choice in particular) in transportation planning. Neural networks have become well known as ‘universal approximators’, thus, this research is an effort to see if the neural networks can approximate the travel behavior, and in turn, be efficient in predicting the ride share of each competing mode.

2 RESEARCH GOAL AND OBJECTIVES

The three major objectives of this thesis are to: present an approach to handle the data from a survey database, address the mode choice problem using artificial neural networks, and compare the results of this approach with the results of traditional models vis-à-vis logit model and linear regression approach.

3 MOTIVATION FOR RESEARCH

The main motivation in conducting this research is to study the applicability of artificial intelligence tools in transportation planning areas. Neural networks have achieved excellent results in many diverse areas of engineering. In recent years, neural networks have become well known as “universal approximators” and this aspect has triggered interest in transportation planners. In transportation planning, there is always a need to model the trips using various modeling techniques, which is same as estimating the trip maker behavior. In this way, the idea of developing mode choice models using neural networks is justified.

4 ORGANIZATION OF THESIS

Now that the research objectives and motivation have been explained, we shall, in the next section discuss the literature review of the mode choice procedures. The section on literature review covers the literature of logit model, linear regression and artificial neural networks. Though there is not much literature pertaining to the use of neural networks in mode choice modeling, few important works are cited. After this section, there is a section on description of neural networks. This section explains the basics of neural networks covering the definition of an artificial neuron, characteristics of a neural network, and finally training and validation of the neural network. The next section focuses on modeling which describes three important aspects of our research: data extraction, data processing and development of mode choice models. In the following section, the results of modeling are shown and various conclusions are made based on those results, finally the best modeling technique is identified.

5 LITERATURE REVIEW

5.1 Introduction

Mode Choice problem has been approached by transportation planners in many different ways. In a broad way all these approaches can be classified into two categories- discrete choice models and non-discrete choice models. Discrete choice models include probit model, multinomial logit model and nested logit model. Non-discrete choice models include regression approach, cross classification tables and diversion curves. In our research we shall concentrate on three models - *Logit model, Linear regression approach, and the Artificial neural networks approach*. All three methods model the probability of a passenger choosing a given mode of transportation.

5.2 Logit model

One of the best known stochastic choice models is the logit model (Mc Fadden (1974), Kanafani (1983), Ben-Akiva and Lerman (1985)). This section on the description of logit model is based on a working paper of Teodorovic (2003). Logit Model was first introduced in the case of binary choice. The Multinomial Logit represents the generalization of the original model to more than two alternatives. The logit model is based on the assumption that random variables representing random terms are independent and distributed by Gumbel's probability density function. We denote by $f(\epsilon)$ the probability density function of random variable that is Gumbel distributed. The $f(\epsilon)$ equals:

$$f(\epsilon) = \mu e^{-\mu(\epsilon-\eta)} \exp[-e^{-\mu(\epsilon-\eta)}]$$

where:

η - location parameter

μ - scale parameter ($\mu > 0$)

Using assumption about Gumbel's distribution, it can be shown, that the probability that a given passenger q chooses alternative A_i within the choice set C_q equals:

$$P(i) = \frac{e^{V_i}}{\sum_k e^{V_k}}$$

(The index notation is simplified for clarity of presentation).

In more general case, (when scale parameter $\mu \neq 1$), this probability equals:

$$P(i) = \frac{e^{\mu V_i}}{\sum_k e^{\mu V_k}}$$

The standard utility function of the logit model is linear. In some cases, the choice function is not linear so the least squares method or the multiple regression technique can only be used to estimate the parameters of the choice function in special cases. The maximum likelihood method is used to estimate parameters of choice functions. Choice models can be based on disaggregated or aggregated data. Choice models based on disaggregated data presume that each passenger evaluates the advantages and defects of each alternative differently, so that the passenger population must be polled before estimating the parameters of the choice function. In this case, each pooled passenger corresponds to a different choice function value. For choice models based on aggregate data, all passengers correspond to the same choice function value, since it is assumed that the variables in the choice function are equal for all passengers. Therefore, the probability of choosing a certain alternative for aggregated data is equal for all passengers.

It is fair to say that the logit model has given some unrealistic results in certain situations. Logit model is characterized by Independence from Irrelevant Alternatives Property (IIA). This property states that for a specific decision-maker the ratio of the choice probabilities of any two alternatives is completely unaffected by the utilities of any other alternatives. This is the consequence of the basic assumption in the derivation of the logit model which states that the error terms of alternatives are independent. There are many situations when this assumption is not a realistic assumption. When there are correlations among alternatives, logit model is not capable to capture these correlations. This weakness of the logit model is illustrated by McFadden's famous blue bus/red bus paradox in the modal choice context. If a decision maker has to choose between two available modes – car and blue bus, and say the probabilities of choosing each mode are 0.75 and 0.25 respectively. If a new set of buses painted in red are introduced into the market, then according to the logit model the red buses are considered as a third set of competing mode. As a result of this, the probability of choosing car would also decrease. This is not appropriate because the decision maker's probability to choose car mode would not change due to the addition of buses of different color. So this is the drawback of logit model. The Nested logit model (Ben Akiva (1985)) addresses the issue of dependency of one variable on the other. It basically, groups all the dependent alternatives into one nest rather than treating them as separate independent variables.

5.3 Linear regression approach

Using the data obtained from behavioral surveys or from the past data, the probability of choosing a particular mode can be estimated using the equation below:

$$P = a(\Delta C) + b(\Delta T) + d \quad (\text{Equation of the Stopher London model})$$

where, ΔC denotes the cost difference between a given mode and the cheapest mode, ΔT denotes the travel time difference between a given mode and the fastest mode and P denotes the probability of choosing a particular mode.

To determine the coefficients, a , b , and d , linear regression is used. The linear regression method has been used for various travel demand forecasting purposes including trip generation and mode choice procedures. If an output variable is known to be linearly dependent on one or more input variables; using linear regression the output can be mapped against these explanatory input variables. The name 'linear' stresses the fact that the equation thus developed using regression is a linear equation. Similar to the neural networks model, the calibration and validation of the model is to be done using the available data. The most important thing to check after the calibration procedure of any linear regression procedure is to test the null hypothesis (to check if the estimated parameters are different from zero). Two standard tests called the t-test and the f-test are used to check for this hypothesis.

It is always a good idea to plot the explanatory input variables against each other to check for any correlation between them. If a high correlation is observed among any two input variables then only one of them should be considered since the other variable can be expressed as a linear function of this chosen variable. The most common goodness of fit measure for the regression models is the regression coefficient. Many commercial software packages have a provision for regression analysis within them.

This technique has its own disadvantages; for instance, it could lead to unbounded probabilities (<0.0 or >1.0) for extreme values of independent variables. It also violates some multiple regression assumptions, such as errors are not normally distributed and the

variance of the error term is not constant – it changes with the independent variable value.

5.4 Artificial neural networks approach

Over the last few years, artificial intelligence techniques like neural networks are being used in the place of traditional regression methods for developing various travel demand forecasting models. Neural networks are capable of learning from examples, generalizing the knowledge learnt and apply to new data, and above all they are able to capture complex relationships in a relatively easier way than other computational methods. Among other authors, neural networks have been used in the transportation demand forecasting by Chin et al (1992), Teodorović and Vukadinovic (1998), and forecasting intercity flows by Nijkamp et al (1996). Conceptually, feedforward neural networks approximate unknown functions to a good level of accuracy, i.e., they can be considered as “universal approximators”. The theorem proved by Hornik et al (1989) and Cybenko (1989) states that a multilayered feedforward neural network with one hidden layer can approximate any continuous function up to a desired degree of accuracy provided it contains a sufficient number of nodes in the hidden layer.

6 DESCRIPTION OF NEURAL NETWORKS

6.1 Introduction

Artificial neural networks as the name suggests are inspired by the biology of a brain's neuron. Human beings can perform a wide range of complex tasks in a relatively easier way as compared to computers. So the researchers are looking for ways in which human intelligence can be incorporated into machines so that they can also perform certain complex tasks easily. Artificial neurons have the characteristics of a biological neuron and these neurons are organized in a way that is reminiscent of the human brain. ANN also display a striking number of brain's properties like learning from experience, generalization from previous instances and apply to new data, etc.

The theorem proved by Hornik et al. (1989) and Cybenko (1989) states that a multilayered feedforward neural network with one hidden layer can approximate any continuous function up to a desired degree of accuracy provided it contains a sufficient number of nodes in the hidden layer. This means that conceptually, feedforward neural networks approximate unknown functions which means, they can be considered as universal approximators.

6.2 Characteristics of neural networks

The first model of an artificial neuron was proposed by McCulloch and Pitts (1943). It was a binary device with a binary input, binary output, and fixed activation threshold. In the Figure 2 below, an artificial neuron is shown along with the tasks performed by it.

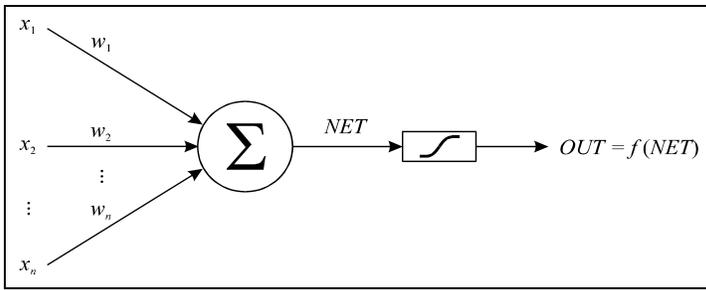


Figure 2: Schematic of an artificial neuron with activation function (Teodorović and Vukadinovic (1998))

The input signals x_1, x_2, \dots, x_n representing the output signals of other neurons, are multiplied by the associated connection strengths w_1, w_2, \dots, w_n (also called weights). The output signal NET is equal to the weighted sum of input signals. The range of the weighted sum of input signals, NET , is compressed by an ‘S’ curve such that the value of the output signal, OUT , never exceeds a relatively low level regardless of the value of NET . Most commonly used activation functions are step function, sigmoid function, hyper tangent function and identity function. The transformation of input signals by a logistic curve enables the receiving and processing of very weak and very strong signals. The present neural network architecture is based on a simplified model of the brain, the processing task being distributed over numerous neurons (nodes or processing elements).

Any neural network has the following characteristics: a) a set of processing elements, b) connectivity of those elements, c) the rule of signal propagation through the network, d) activation or transfer functions, e) training algorithms (learning rules or learning algorithms), and f) environment in which the network functions.

These characteristics can be better understood with the help of the following example shown in Figure 3 below.

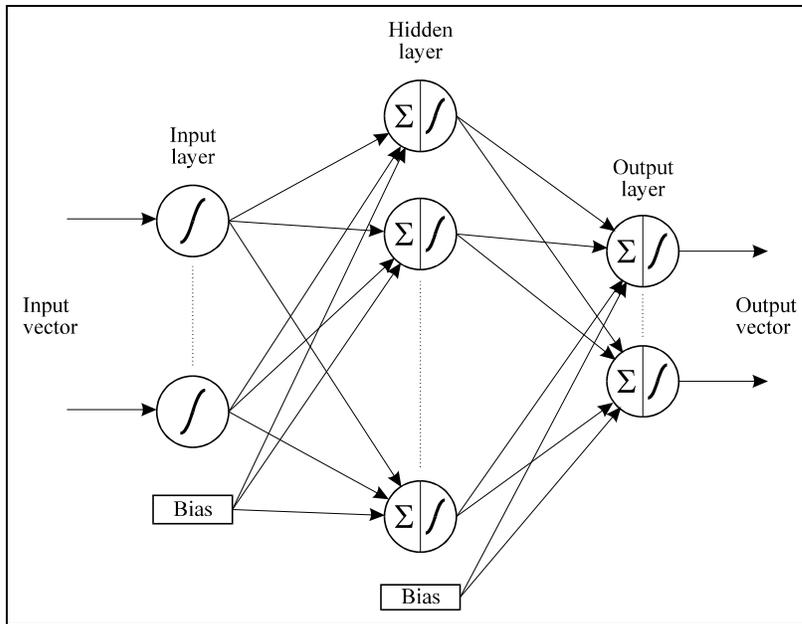


Figure 3: Two-layered feedforward neural network (Teodorović and Vukadinovic (1998))

In this network we can see three layers through which the input signal has to pass through. Each layer has a certain number of processing elements (nodes) as shown in the figure. The number of nodes varies depending on the problem that is being addressed. Any neural network has three types of nodes – input, output and hidden. Input nodes receive input signals from sources outside the network. Output nodes transmit signals that is, output values outside the network. All other nodes not belonging to the input/output layers belong to the hidden layers. The nodes of one layer are connected to the nodes of the adjacent layer. This connectivity can be partial or full connectivity. Each node transmits signals of different strengths to its neighboring nodes. The connection strengths are also called as weights of the connections. The propagation of input signal usually follows certain rules; in this case since it is a multilayered feed forward network, the input signal extends forward through several layers, while it is being processed to estimate the network's output signal. Each node is a processing element associated with

the corresponding activation function by which the weighted sum of input values is transformed to determine the output value. To each node's input only the outputs of nodes from a previous layer are supplied and the output signal is transmitted to the nodes of the next layer.

6.3 Training of a neural network

After the building of neural network, the input data is fed into the network through the input nodes, along with the desired output data. The neural networks self-adapt to the data and incite appropriate responses. This process of making the network adapt to the data is known as training of a neural network and the algorithms used for this purpose are known as training algorithms. These algorithms can be classified according to their modeling, learning, and validation properties. The modeling abilities of an algorithm determine the range of nonlinear functions that it is able to precisely reproduce. The chosen structure of a neural network model can influence the convergence rate of a training algorithm and even determine the type of learning to be used.

The multilayered neural networks have come into use after the development of an error backpropagation algorithm, which was used for training a network. Various researches have independently developed a suitable and currently most popular algorithm for training a multilayered feedforward neural network (Rumelhart and McClelland (1986), Le Cun (1985), Parker (1985)). The proposed backpropagation algorithm is a gradient procedure. The activation functions of nodes are bounded, continuous, monotonously increasing, nonlinear, differentiable functions. The output function of the network is a continuous, differentiable weight function enabling the search of the extremum by the “gradient descent” algorithm.

The optimal weights, w_{ij} , are determined by the rule of gradient descent (*delta rule*, *generalized delta rule*) minimizing the criterion function or error. Each iteration of the algorithm (cycle or epoch defined as the process of transmission of one or a few training pairs through the network whereby the error is calculated) contains two passes (Figure 4):

- Propagation of one or a set of input signals forward to the output layer (in the original algorithm input signals were brought to the network individually)
- Backward pass where the computed error extends backward in order to calculate the changes of parameters (weight of the network's branches).

The procedure is performed in numerous iterations using the same training pairs until the error becomes “sufficiently” small.

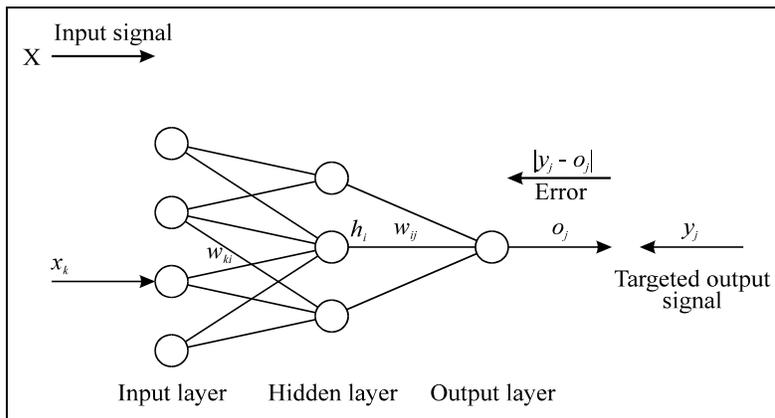


Figure 4: Taxonomy of training a multilayered perceptron: the input signal extends forward and the computed error backward. (Teodorović and Vukadinovic (1998)).

6.4 Sample calculations

This section illustrates the procedure used by artificial neural network to calculate output from the given inputs.

Consider the simple network shown in Figure 5 below:

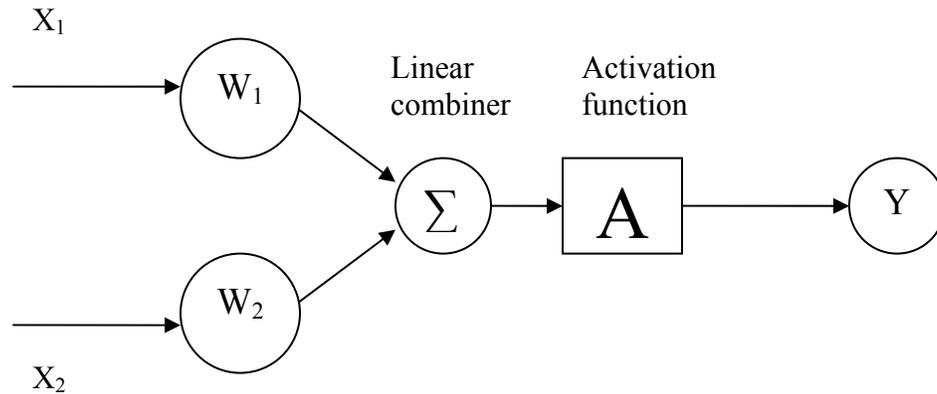


Figure 5 : Sample calculation of the output

As the inputs X_1 and X_2 pass through the input layer of neurons, the weights W_1 and W_2 are assigned arbitrarily by the network. The linear combiner calculates the weighted sum of these inputs, $NET = W_1 * X_1 + W_2 * X_2$. The range of NET, is compressed by an ‘S’ curve such that the value of the output signal, Y, never exceeds a relatively low level regardless of the value of NET. Most commonly used activation functions are step function, sigmoid function, hyper tangent function and identity function. The transformation of input signals by a logistic curve enables the receiving and processing of very weak and very strong signals. Let the threshold value be defined by Θ . Now consider the following numerical example: $X_1 = 0$, $X_2 = 1$, $W_1 = 0.2$, $W_2 = -0.1$, $\Theta = 0.2$, and the Step function as activation function ($Y=0$ if $NET < \Theta$, else $Y=1$). The output is calculated as follows, $W_1 * X_1 + W_2 * X_2 = 0.2 * 0 + (-0.1) * 1 = -0.1 < 0.2$, so $Y = 0$. In this way the output is calculated by the neural network, the weights are adjusted by using the learning rules as described earlier and the same procedure is followed to calculate the output using new set of weights.

6.5 Generalization

Any neural network is of use if it is able to generalize correctly from a limited number of samples, which means that the algorithm has to interpolate and locally extrapolate rather precisely. Once a neural network has been designed, its reaction can, to a certain extent, be insensitive to minor variations in the input set of data. This ability to differentiate in the presence of noise and distortion of shapes is of key importance. It is worthwhile to note that the artificial neural network automatically makes generalizations due to its structure, without the use of a human intelligence that would be embedded in it in the form of *ad hoc* computer programs.

6.6 Testing of a neural network

Any model has to be validated using some data. A trained neural network is validated using testing data. The available data is always divided into three parts prior to the training – training data, cross-validation data, and testing data. The training data is used during the training purposes; the cross-validation data is also used during the training but not to train the network, instead to check the learning of the network during the training process. The testing data is totally a different set of data that the network is unaware of; this data is used for validation of the trained network. If the network is able to generalize rather precisely the output for this testing data, then it means that the neural network is able to predict the output correctly for new data and hence the network is validated. The amount of data that is to be used for training and testing purposes is dependent on the availability of the data, but in general the training data is $2/3^{\text{rd}}$ of the full data and the remaining is used for testing purposes. The cross-validation data can be $1/10^{\text{th}}$ of the training data.

7 MODELING

There are two main steps in developing the mode choice models using ATS data (Figure 6). In the first step, called Data Extraction and Processing, relevant data is extracted from the survey database, the variables to be used in modeling are defined, and data mining procedures (clustering) are applied to the extracted data in order to create artificial passengers. In the second step, the data from the first step is used to develop mode choice models using three different approaches – artificial neural networks, multinomial logit and linear regression. These two steps along with the sub routines are explained in detail in the following pages.

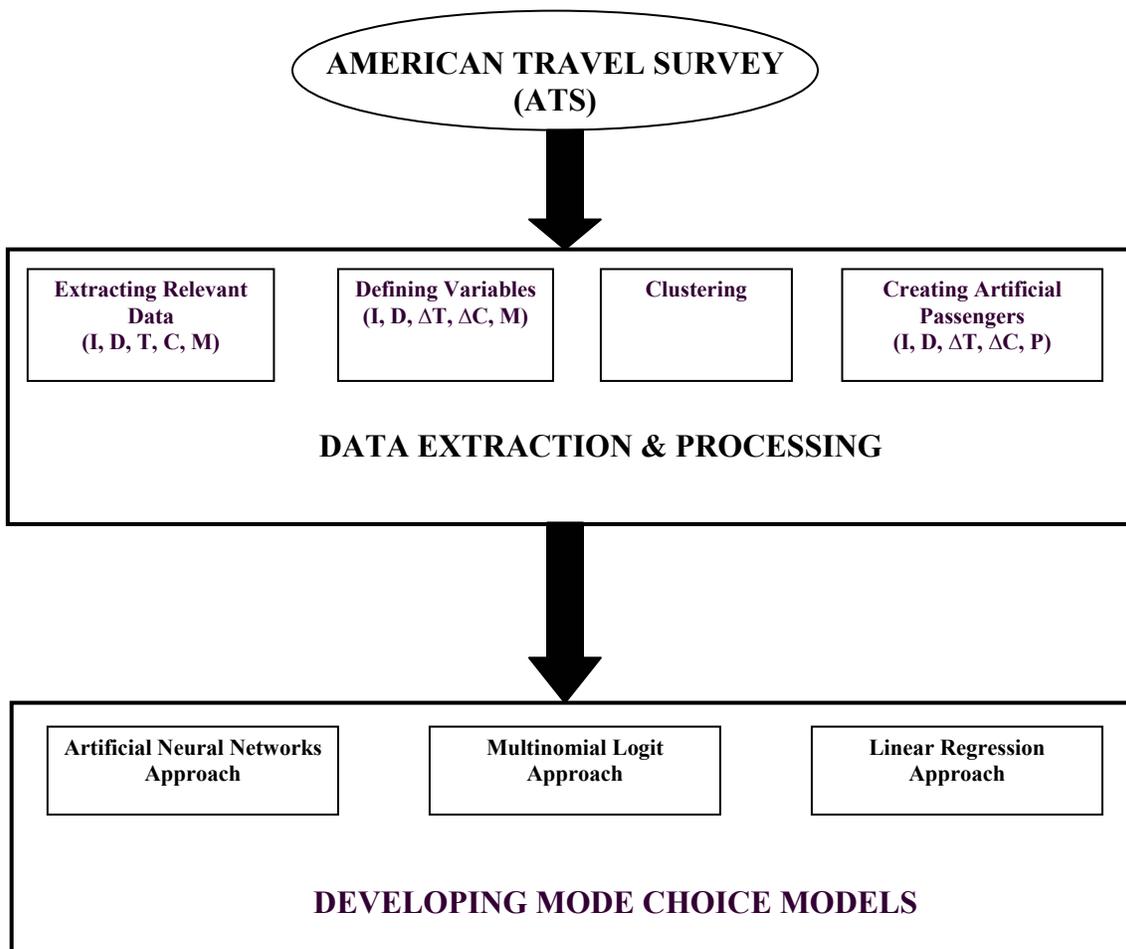


Figure 6: Overview of the Mode choice modeling procedure

7.1 Data extraction

ATS conducted by BTS in 1995 (16) is one of the most valuable data sources for long-distance trip analysis. The survey contains 337,520 household trips and 556,026 person trips, and it surveyed more than 300 variables for each trip. The variables include: 1) trip characteristics such as trip purpose, trip origin and destination, trip distance, transportation mode, access/egress distance to/from airports, etc., and 2) traveler's personal information such as income, family size, education level, etc. The survey, however, does not include information on travel cost and travel time that trip maker spent to complete the trip.

In order to overcome this practical difficulty, the travel cost and travel time for each mode for a trip are synthesized using assumptions. As shown in Table 1, for example, travel speed of automobile is assumed to be 30 and 60 miles per hour in urban area and highway respectively. Total travel time for an air traveler is the sum of access/egress time to/from airport, processing time at airports and flying time. In order to consider hub-and-spoke network structure in the commercial airline flights, a detouring factor is also applied.

As to travel cost, flat unit cost is applied to automobile and GA modes. Depending on the type of airline, two simple fare models are derived using Airline Origin and Destination Survey data (also known as DB1B) (17). The two models are

Major airline fare (cent/trip) = $0.016 \cdot \text{distance} + 276.5$, and

Low-cost airline fare (cent/trip) = $0.05 \cdot \text{distance} + 85$.

It is also assumed that the driving time shall not exceed a limit of 8 hours per day. If the driving time for a trip is more than the maximum, certain amount per Diems is added to the travel cost. The detailed travel costs assumed are summarized in Table 2.

Table 1: Travel Time Assumptions

	Automobile	Airline	GA
Speed (or Travel time)	In urban area: 30 (mph) In highway: 60 (mph)	Table function ¹⁾	Table function ²⁾
Access/egress Distance	-	30 (miles)	15 (miles)
Processing Time	-	Orig. airport = 60 (min) Dest. airport = 30 (min)	Orig. airport = 30 (min) Dest. airport = 20 (min)
Detouring ratio ³⁾	-	1.2	-

¹⁾ Travel distance vs. flight time, Airline (min)

Miles	300	600	...	3000	3300	3600
Time	50.0	89.0	...	397	443	485

²⁾ Travel distance vs. flight time, GA (min)

Miles	300	600	...	2700	3000
Time	100.0	178.0	...	702	974

³⁾ Accounts for increment of trip distance due to the connecting flight.

Table 2: Travel Cost Assumptions

	Automobile	Airline ¹⁾	GA
Cost per mile (Cent/mile)	30 (cent/mile)	Equations ²⁾	150 (cent/mile)
Access/egress Distance	-	30 (miles)	15 (miles)
Lodging cost ³⁾	150 (\$/day)	-	150 (\$/day)
Average Occupancy	Business trip: 1.2 Non-business trip: 3.0 (travelers/car)	-	All trips: 1.2 (travelers /flight)

¹⁾ It is assumed that 15 % of air travelers take low-cost carriers.

²⁾ Major airline fare (cent/trip) = 0.016(dist) + 276.5.

Low-cost airline fare (cent/trip) = 0.05(dist) + 85.

³⁾ If driving time (or flight time for GA) is longer than maximum hours a day, the lodging cost is added to the total trip cost. We assume the maximum driving/flight hours/day to be 8 hours.

7.2 Data processing

After the data extraction step, the following input data for each trip maker is available for both business trips and non-business trips – Income (I), Distance traveled in that trip (D), Travel Times (T) and Travel Costs (C) for three modes (commercial aircraft (CA), automobile (A) and general aviation (GA) and finally the Mode (M) chosen by trip maker for this trip.

In the next step, travel time and travel cost variables are transformed into Time Differential (ΔT) and Cost Differential (ΔC) (e.g.: for commercial aircraft mode ΔT is the difference between the travel time for the most competing mode (quickest among Auto and General Aircraft) and the travel time for the Commercial aircraft) and delta C (difference between the trip costs for the most competing mode (cheapest among Auto and General Aircraft) and the trip cost for Commercial aircraft). The rationale behind developing such differential variables is to include the marginal utility that a trip maker gets when he chooses a particular mode. In this way we considered the socio-economic characteristics of the trip maker (Income), the trip characteristics (Distance traveled, trip purpose) and the service characteristics (Time Differential and Cost Differential).

As stated earlier, the mode chosen by a trip maker is mentioned in the ATS database (1-A, 2-CA and 3-GA). To develop mode choice models using different approaches there is a need to define a probability of trip maker to choose a particular mode. To do this the whole data is grouped using a clustering algorithm and then a probability for each data cluster is defined. This procedure could be formally described by the following algorithm:

Step 1: Classified the whole data set using K-Means clustering. Form clusters. Each cluster represents one “artificial passenger”.

Step 2: Calculate characteristic variables of every “artificial passenger” as the average of all the passengers belonging to that cluster.

Step 3: For every cluster, count the number of passengers within a cluster choosing a particular mode. Using these counts, calculate for every “artificial passenger” the probability of “artificial passenger” choosing a particular mode. (See Figure 7 for an illustration)

The initial data for business travelers consisted of 118000 trips, K-Means clustering is applied and the data is classified into 1200 clusters. All the programming is performed in MATLAB and MS-Excel and the codes are attached in the Appendix B.

The K-Means procedure used for clustering is explained in the next section.

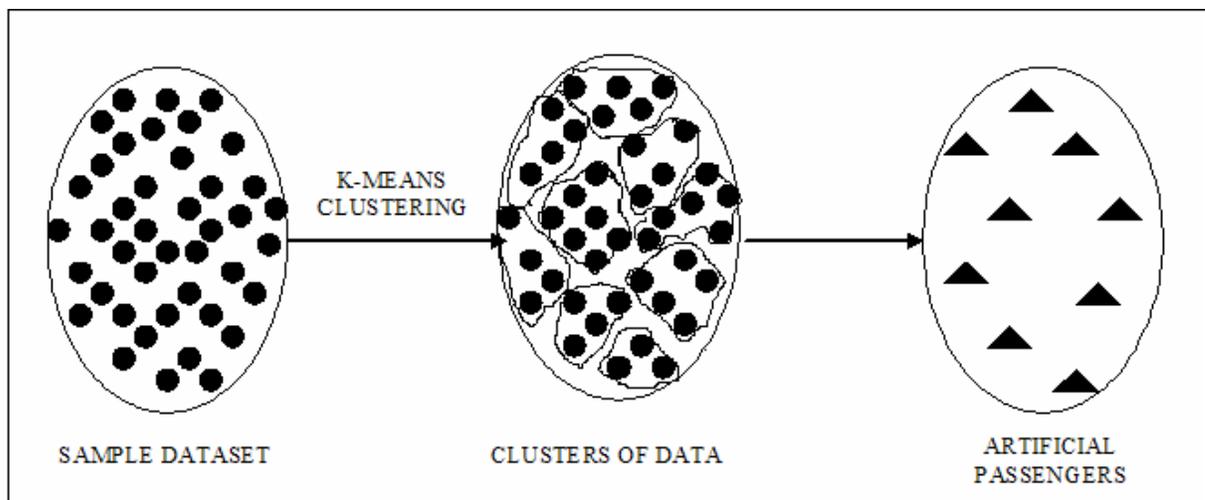


Figure 7: Creation of “Artificial Passengers”

7.2.1 K-means clustering procedure

This algorithm has as an input, a predefined number of clusters, k . Means stands for an average, an average location of all the members of a particular cluster. When dealing with

clustering techniques, one has to adopt a notion of a high dimensional space, or space in which orthogonal dimensions are all attributes from the table of data we are analyzing. The value of each attribute of an example represents a distance of the example from the origin along the attribute axes. Of course, in order to use this geometry efficiently, the values in the data set must all be numeric and should be normalized in order to allow fair computation of the overall distances in a multi-attribute space.

K-means algorithm is a simple, iterative procedure, in which a crucial concept is the one of *centroid*. *Centroid* is an artificial point in the space of records which represents an average location of the particular cluster. The coordinates of this point are averages of attribute values of all examples that belong to the cluster. The steps of the K-means algorithm are given below (18):

1. Choose randomly k points (can be examples also) to be the seeds for the *centroids* of k clusters.
2. Assign each example to the *centroid* closest to the example, forming in this way k exclusive clusters of examples.
3. Update the location of each centroid. The attributes of centroid are average of all attribute values of the examples belonging to the same cluster (*centroid*).
4. Check if the cluster *centroids* have changed their location (coordinates). If yes, start again from the step 2. If not, cluster detection is completed.

The convergence of such an algorithm usually needs only a few iterations. The distance metric that is commonly used for clustering purposes is the Euclidean metric defined below. The Euclidean distance between two points, $p = (p_1, p_2 \dots)$ and $q = (q_1, q_2 \dots)$ is defined as:

$$d = \sqrt{\sum_{i=1}^k (p_i - q_i)^2}$$

The number of clusters k in the K-means method has to be chosen so to match the natural structure of the data in order to obtain good results. A trial and error procedure is performed with different values for k . In principle the best k value will exhibit the smallest intra-cluster distances and largest inter-cluster distances (18).

7.3 Development of mode choice models

In the earlier section the data extraction and processing procedures were explained. The processed data is now used to develop the mode choice models using three different techniques – artificial neural networks approach, multinomial logit and linear regression technique. It is important to mention that the techniques adopted are also a consequence of the nature of the processed data.

7.3.1 Artificial neural networks approach

The building of neural network is done using a commercial Neural Network tool called Neuro Solutions. Three separate networks are developed, each network predicting the probability of each mode (auto, commercial airline and general aviation). The input layer consists of four inputs as mentioned in the data processing step. The output layer consists of only one output namely probability of a person choosing a particular mode. The data obtained from the “data processing” step is divided into two parts - training data and testing data. As mentioned in the earlier section, the total number of “artificial passengers” is equal to 1200. Four-fifths of this data (960) is used for training purposes and One-fifth is used for testing purposes (240). Training data is used for training the neural network. When this data is fed as input to the model the neural network assigns

appropriate weights to each link in the network and calculates the output. Supervised learning is used during the training process. The back propagation error is the absolute value of the difference between the predicted probability and the real probability, and the weights of the network are adjusted using the delta rule. Another set of data called the cross-validation data (CV) (20% of the training data) is not directly used for training purpose but it is used for validation purposes during the training process. This CV data is different from the testing data that it is not any data that the model does not know, whereas testing data is totally a different set of data, unknown to the network. In the search of the best network that can predict the output well, the number of layers, number of neurons in each layer, the transfer functions and the amount of CV data are chosen by experience and some trial and error procedure. The best networks obtained are shown in Figure 8.

As we are developing a different network for each mode separately, the sum of probabilities would not be equal to one. Since this is a mode choice problem, the sum of probabilities of each mode should be equal to one. We have taken care of this issue by using a process of normalization.

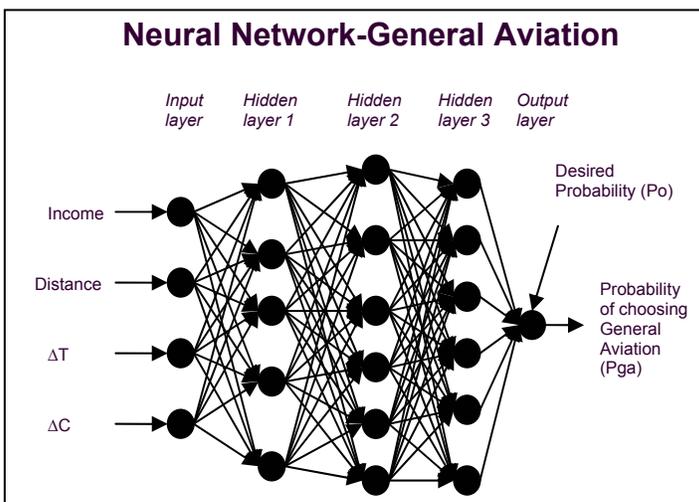
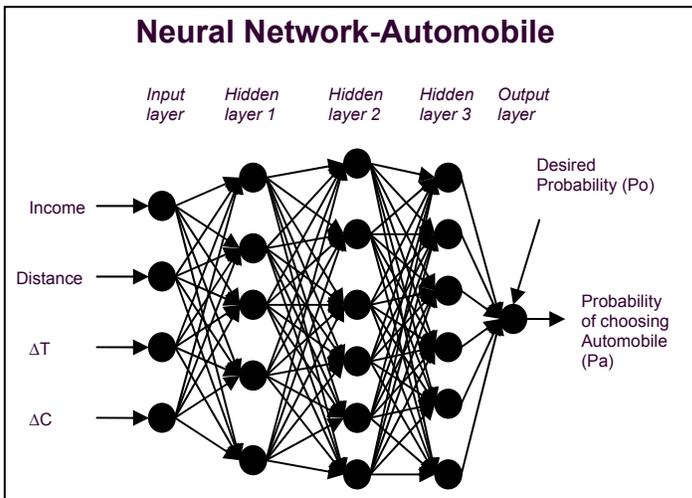
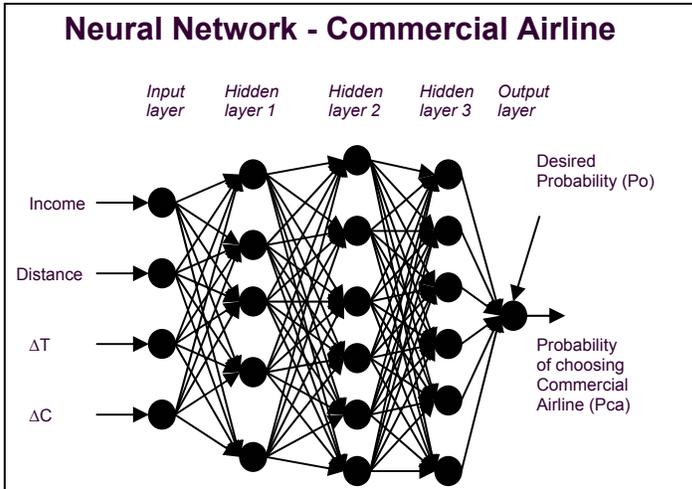


Figure 8: Final Neural network model (number of neurons in a layer is chosen as 5-6 for illustration purposes only, in reality they are many more in number).

7.3.2 Multinomial logit model

All three modes are independent of each other, so it is justified to use the multinomial logit rather than the nested logit. The calibration of the model is done in SAS software using the MDC procedure. For calibration of parameters of the utility function, SAS uses the maximum likelihood technique. McFadden's likelihood index, which is analogous to the r-square in the linear regression model, is calculated. Other goodness of fit measures like the Estrella, Cragg-Uhler1, Cragg-Uhler2, Aldrich-Nelson and Veall-Zimmermann measures are also calculated by SAS. The probabilities for testing data are calculated by using the calibrated utility functions and hence the validity of the model is tested.

(SAS outputs of regression coefficients and parameters are enclosed in the Appendix C).

7.3.3 Linear regression method

We are trying to develop a function approximation to estimate the probability of a trip maker choosing a particular mode. For this purpose linear regression can also be used. The only difference between linear regression approach and the neural networks is that the neural network approach does not give a functional relationship between the output variable and the input variables. So, after defining the artificial passengers, the same set of training data that is used for training neural networks is used for developing the regression equation. Statistical tests including t-test and f-test are conducted to check the significance of the obtained coefficients.

7.3.4 Comparison of models

To compare these different methodologies (Nijkamp et al (1996)) the following statistical indicators have been used - Mean Square Error (MSE) and Average Relative Variance (ARV). The Mean Squared Error is a very common indicator and can be represented as below:

$$MSE = \frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{N} \quad n = 1, 2, \dots, N$$

where y_n denotes the expected values and \hat{y}_n denotes the values calculated from the model by using a data sample with N observations.

Another indicator, often used in the literature on Neural Networks, is the Average Relative Variance (ARV (N)). It is defined as below:

$$ARV(N) = \frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{\sum_{n=1}^N (y_n - \bar{y})^2} \quad n = 1, 2, \dots, N$$

where y_n denotes the expected values and \hat{y}_n denotes the values calculated from the model and \bar{y} is the average of the expected values belonging to the set of data N .

By performing the residual analysis the errors of the test data set can be illustrated. The absolute residual can be defined as below:

$$AR = y_n - \hat{y}_n$$

8 RESULTS

This section presents the results obtained for all three approaches.

a) The calibration of Multinomial logit model in SAS yields the following utility function,

$$U = [3.4681 \cdot 10^{-6}] \cdot I + [4.22 \cdot 10^{-4}] \cdot D - [1.695 \cdot 10^{-1}] \cdot \Delta T + [8.714 \cdot 10^{-3}] \cdot \Delta C,$$

For the testing data, the probabilities for each mode are calculated using this utility function.

b) The results of linear regression are shown below; the value in the parenthesis indicates the value obtained from t-test (for null hypothesis to be false $t \in [-1.96, 1.96]$)

$$P_A = [9.67 \cdot 10^{-1}] - [6.906 \cdot 10^{-7}] \cdot I + [7.3 \cdot 10^{-2}] \cdot \Delta T - [1 \cdot 10^{-3}] \cdot \Delta C$$

(-5.809)
(4.329)
(-7.76)

$$P_{CA} = [-1.22 \cdot 10^{-1}] - [9.003 \cdot 10^{-7}] \cdot I + [1 \cdot 10^{-3}] \cdot D - [5 \cdot 10^{-3}] \cdot \Delta T$$

(8.392)
(13.032)
(-3.122)

$$P_{GA} = [3.947 \cdot 10^{-5}] \cdot D - [3.393 \cdot 10^{-5}] \cdot \Delta C$$

(2.166)
(2.083)

c) Upon several iterations by changing the number of layers, number of neurons in each layer, type of transfer function, learning criteria, etc the final network is as shown in Figure 8. The network has the following attributes - three hidden layers, tanhaxon transfer function, supervised learning and 1000 epochs.

The Mean Square Errors, Average Relative Variances are shown in Table 3. The estimated probabilities from each model are plotted against the real probabilities in Figures 9, 10, and 11. The absolute residuals for all modes are plotted in Figures 12, 13, and 14. **From Table 3 we can see that the logit model has a high mean square error**

and average relative variance as compared to the neural networks model and linear regression method. In the plots of residuals it can be seen that the oscillations of residuals are highest for logit model. Linear regression also has high oscillations as compared to neural network model but less than the oscillations of the logit model.

Table 3: Comparison of the models- neural networks vs. multinomial logit vs. linear regression - *mean square error and average relative variance*

MODE	CRITERIA	NEURAL NETS	LOGIT	REGRESSION
COMMERCIAL AIRLINE	<i>MSE</i>	0.007	0.030	0.014
	<i>ARV</i>	0.062	0.277	0.126
AUTO	<i>MSE</i>	0.011	0.047	0.019
	<i>ARV</i>	0.101	0.420	0.168
GENERAL AVIATION	<i>MSE</i>	0.001	0.010	0.001
	<i>ARV</i>	0.970	17.518	0.969

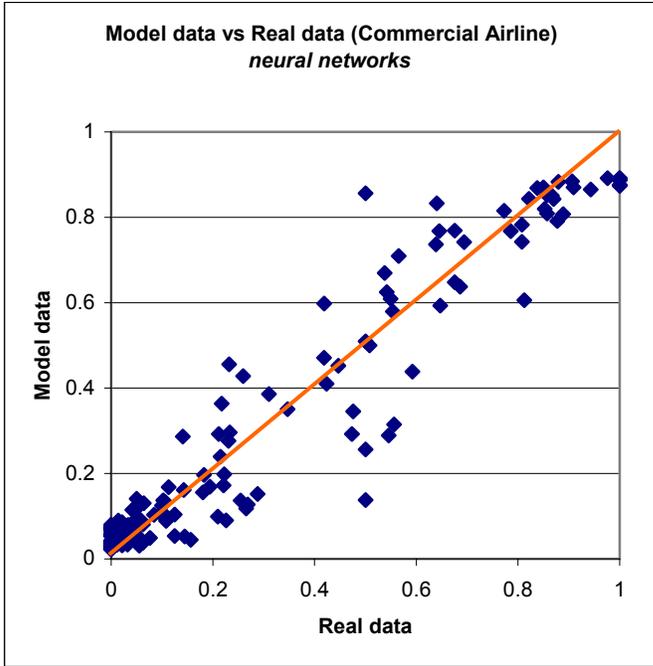


Figure 9a: Commercial airline mode - Comparison of models based on model probability vs real probability –neural network model

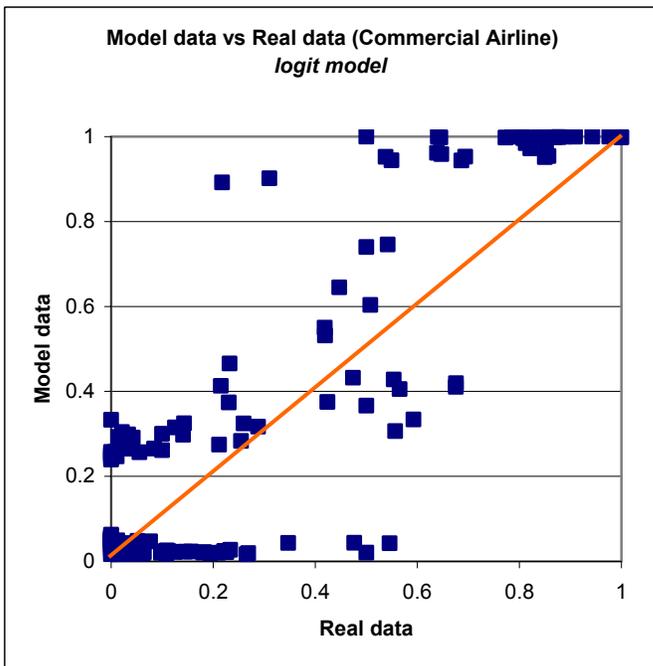


Figure 9b: Commercial airline mode - Comparison of models based on model probability vs real probability –logit model

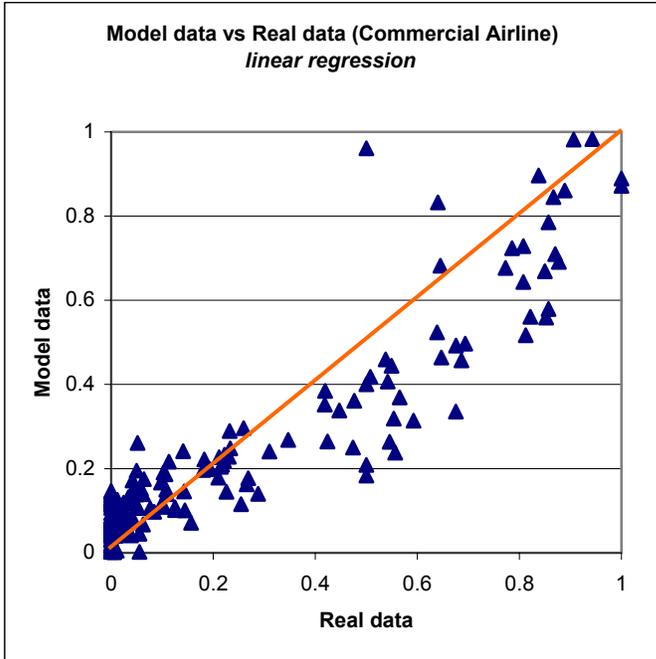


Figure 9c: Commercial airline mode - *Comparison of models based on model probability vs real probability – linear regression model.*

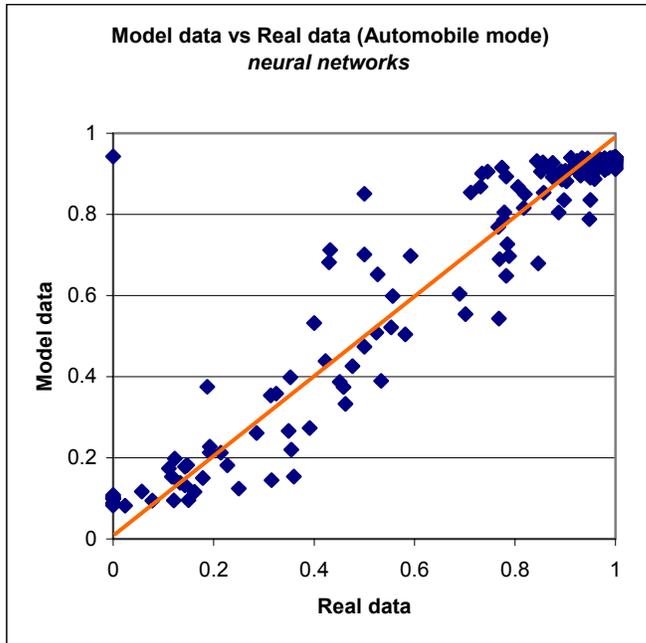


Figure 10a: Automobile mode - Comparison of models based on model probability vs real probability-neural networks

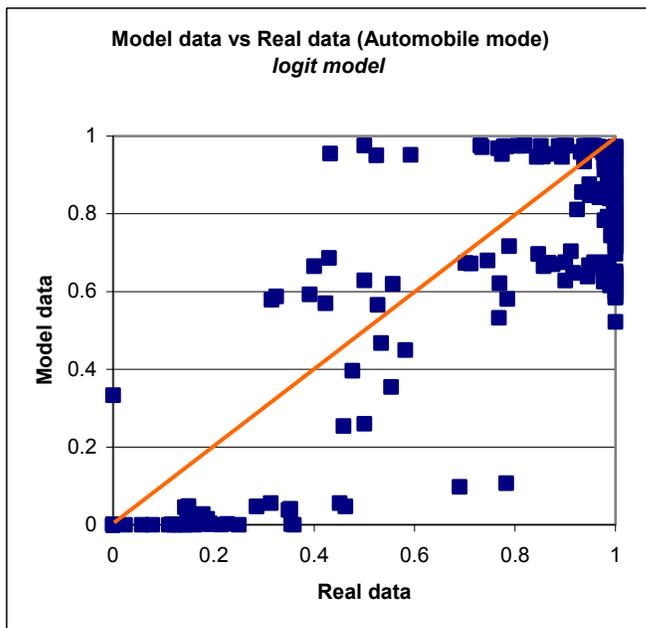


Figure 10b: Automobile mode - Comparison of models based on model probability vs real probability-logit model.

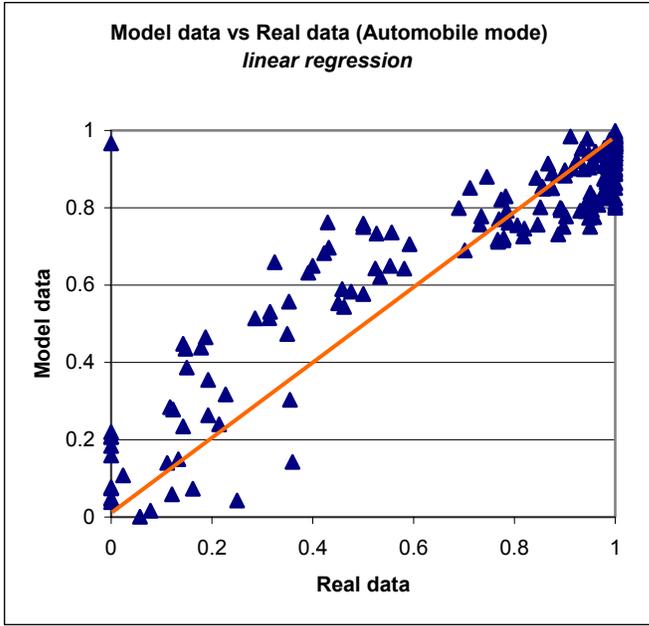


Figure 10c: Automobile mode - Comparison of models based on model probability vs real probability – linear regression model.

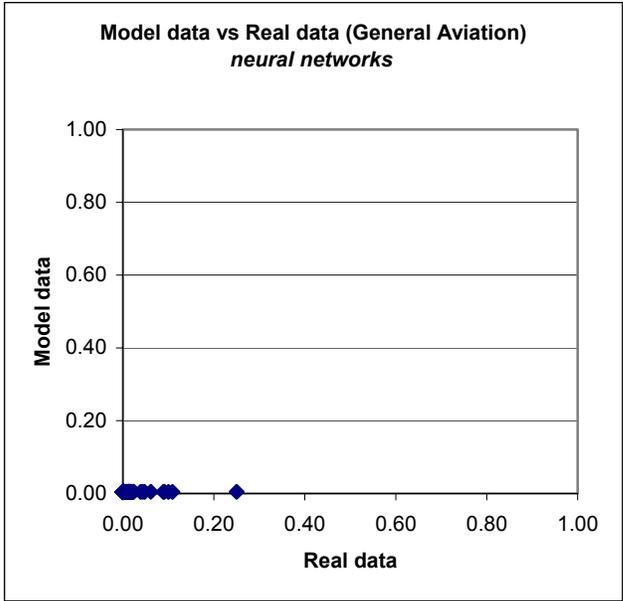


Figure 11a: General aviation mode - Comparison of models based on model probability vs real probability – neural network model

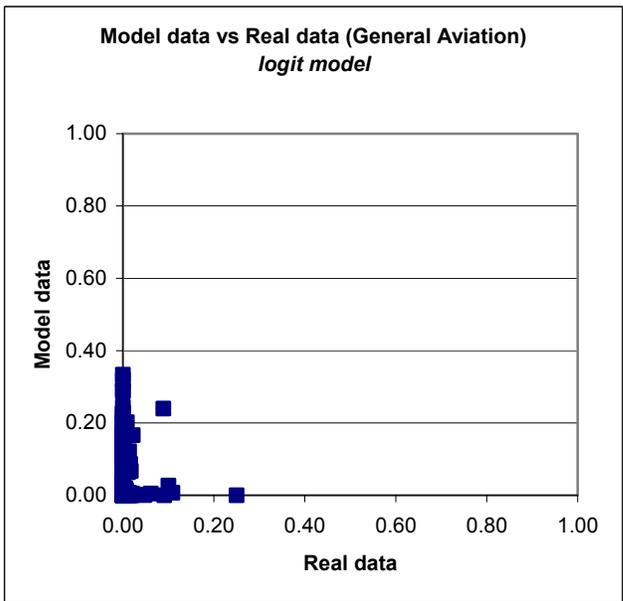


Figure 11b: General aviation mode - Comparison of models based on model probability vs real probability – logit model

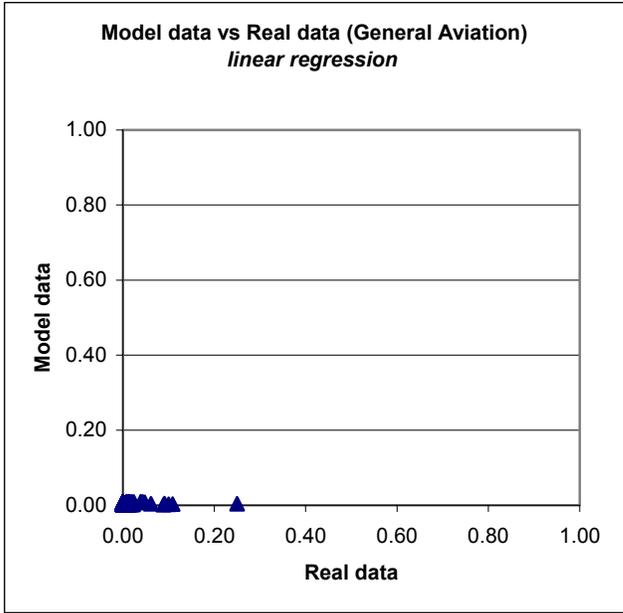


Figure 11c: General aviation mode - *Comparison of models based on model probability vs real probability – linear regression*

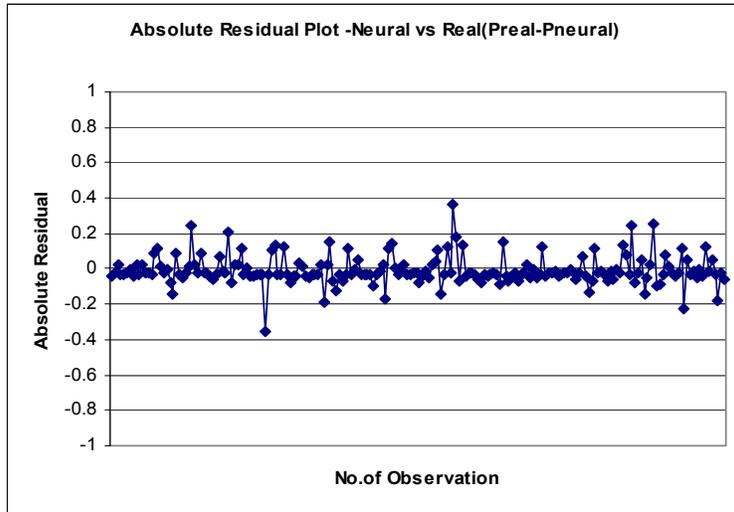


Figure 12a: Commercial airline mode - *Absolute Residuals* – neural network model

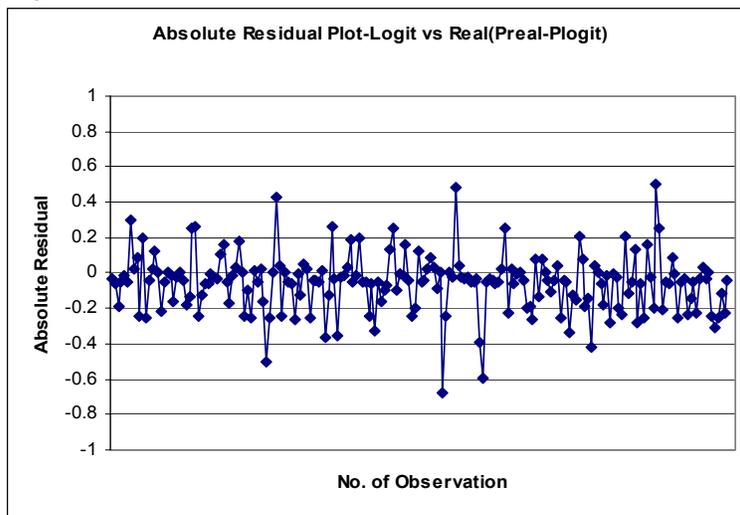


Figure 12b: Commercial airline mode - *Absolute Residuals* – logit model

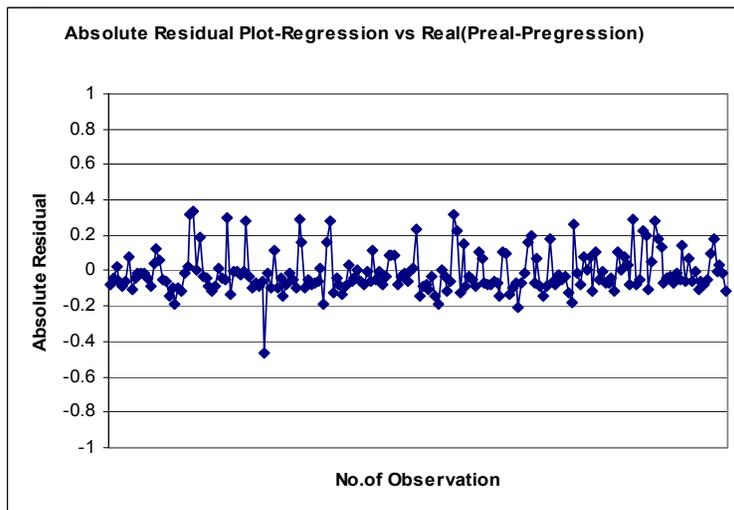


Figure 12c: Commercial airline mode - *Absolute Residuals* – linear regression

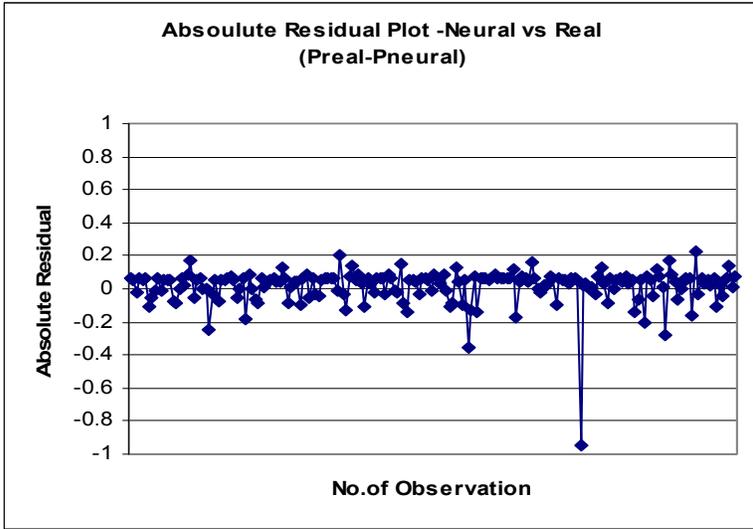


Figure 13a: Automobile mode - *Absolute Residuals* – neural networks

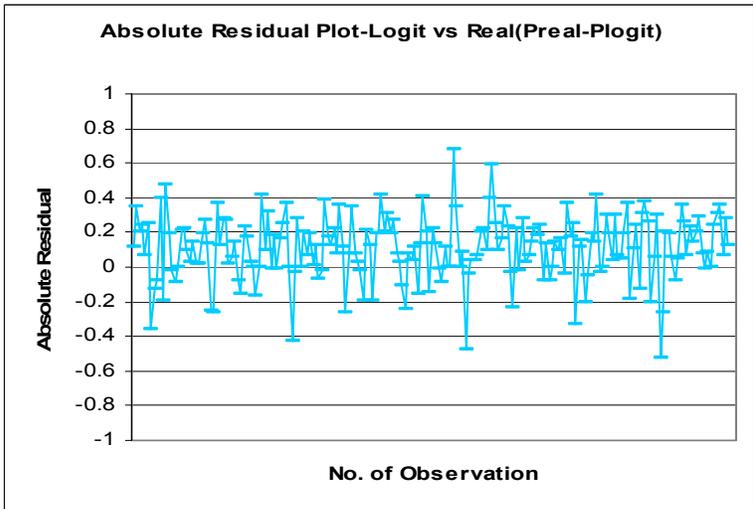


Figure 13b: Automobile mode - *Absolute Residuals* – logit model

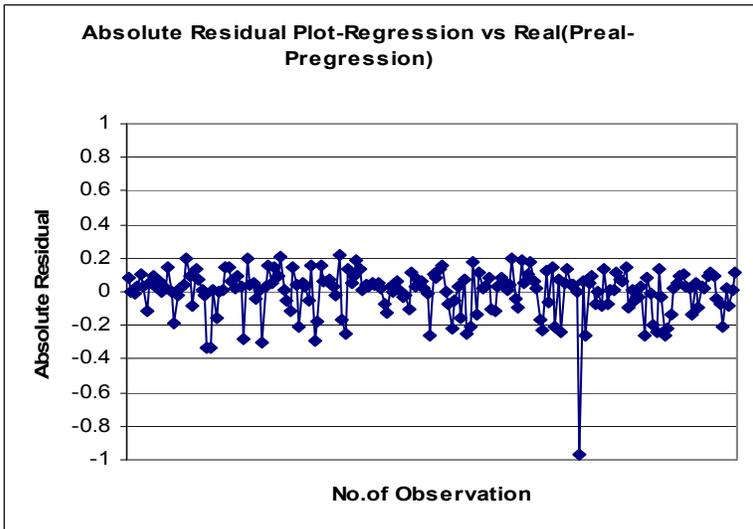


Figure 13c: Automobile mode - *Absolute Residuals* – linear regression

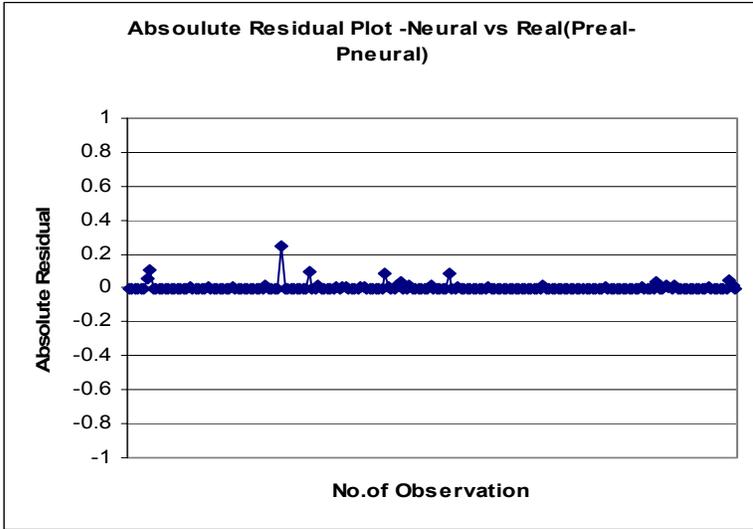


Figure 14a: General aviation mode - *Absolute Residuals – neural networks*

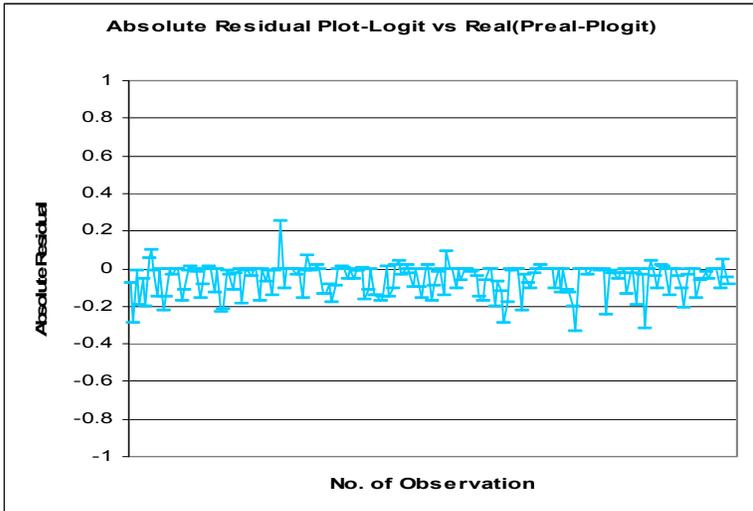


Figure 14b: General aviation mode - *Absolute Residuals – logit model*

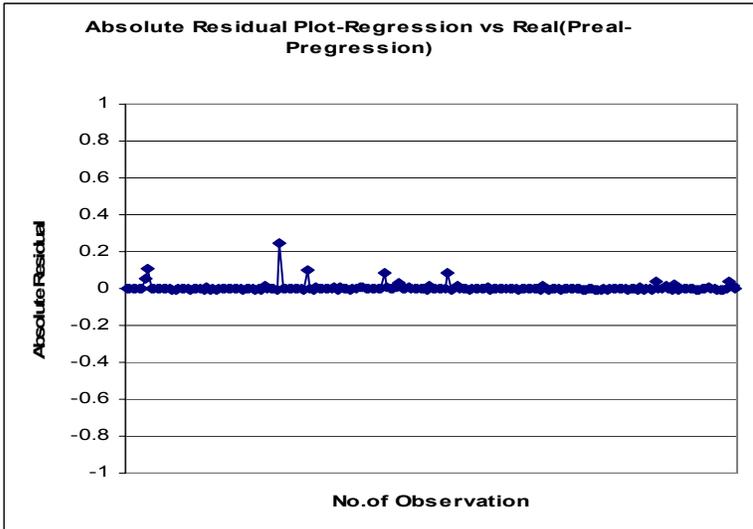


Figure 14c: General aviation mode - *Absolute Residuals – linear regression*

9 CONCLUSIONS

To conclude, in this thesis we have tried to accomplish three objectives: a) present an approach to handle the data from a survey database, b) address the mode choice problem using artificial neural networks, and c) compare the results of this approach with the results of traditional models vis-à-vis logit model and linear regression approach. From the results obtained, the following conclusions can be made:

1. From the results of mean square error, average relative variance, and residuals estimates, it can be concluded that the neural networks approach gives better results than logit and regression models. This suggests that neural networks are “universal approximators”.
2. These positive results should tend to encourage other researchers to apply artificial intelligence methods (e.g.: neural networks) to the field of transportation. Artificial neural networks are capable of producing good results in travel demand forecasting areas, and this research has supported this capability.
3. From these results, it can not be generalized that in all situations neural networks produce better results than logit; it only means that for this problem (and this set of variables) neural networks offer better results than logit and regression methods.
4. These results should encourage other practitioners and researchers in fields other than transportation also to use AI tools for areas such as forecasting, function approximations, predictions, classifications, and pattern recognitions.

9.1 Comments on using neural networks

1. It is relatively easy to model using neural networks, in part because there are many commercial software available that are very user friendly; consequently, the modeling can be done without any extensive knowledge of coding.
2. The time taken to obtain the solution is usually less than or equal to the time taken by traditional models.
3. When modeling, using neural networks, a physical equation is not obtained that can predict the number of trips. But instead, a trained network is obtained which has weights stored in it. This is one of the criticisms of the researchers who support logit and other traditional models that give a physical equation. But, it is to be noted that, the trained network is as good as an equation and can be directly used on new data for predicting the output.

9.2 Scope for future research

In the current research, artificial neural networks have been used to address the mode choice problem. In the future, it would be interesting to study the same mode choice problem using other artificial intelligence tools like fuzzy systems, agent based modeling, etc. It is also noted in the research results that, though neural networks are able to model the general aviation mode share better than the other two models, there is a need to work more on developing a more appropriate model for general aviation mode.

10 REFERENCES

1. Garber, N., and Hoel, L. *Traffic and Highway Engineering*. Brooks/ Cole publishers, 3rd edition, 2002.
2. Nijkamp, P., Reggiani, A. and Tritapepe, T. Modelling inter-urban transport flows in Italy: A comparison between Neural network analysis and logit analysis. *Transportation Research C*, 4, 1996, pp.323-338.
3. Mc Fadden, D. Conditional logit analysis of qualitative choice behaviour. *Frontiers in Econometrics*, 1974, pp.105-142.
4. Kanafani, A., *Transportation Demand Analysis*. McGraw Hill, 1983.
5. Ben-Akiva, M., and Lerman, S., *Discrete Choice Analysis: Theory and Application to Travel Demand*. The MIT Press, 1985.
6. Teodorović, D., “Logit Model: Theory and Transportation Applications”, Working Paper, Virginia Tech, 2003.
7. Teodorović, D., and Vukadinovic, K. *Traffic Control and Transport Planning: A Fuzzy Sets and Neural Networks Approach*. Kluwer Academic Publishers, 1st edition, December 1998.
8. Chin, S.M., Hwang, H.L., Miaou, S.P., “Transportation Demand Forecasting with a Computer Simulated Neural Network Model”, in S.G. Ritchie and C. Hendrickson (eds), *Proceedings of the International Conference on Artificial Intelligence Applications in Transportation Engineering*, San Buenaventura, CA, Institute of Transportation Studies, University of California, Irvine, 1992.
9. Hornik, K., Stinchcombe, M., White, H., “Multilayer Feedforward Networks are Universal Approximators”, *Neural Networks*, 2, 1989, pp.359-366.

10. Cybenko, G., "Approximation by Superposition of Sigmoidal Functions",
Mathematics of Control, Signals and System, 2, 1989, pp.193-204.
11. McCulloch, W.S., and Pitts, W. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics*, 5, 1943, pp.115-133.
12. Rumelhart, D.E., and McClelland, J.L. *Parallel Distributed Processing: Explorations in the microstructure of cognition*. (vols. 1 and 2). Cambridge, MA: MIT Press, 1986.
13. Le Cun, Y. "A Learning Procedure for Assymmetric Threshold Network." *Proceedings of Cognitiva*, 85, 1985, pp.599-604.
14. Parker, D.B. *Learning-logic* (TR-47). Massachusetts Institute of Technology, Cambridge, MA, 1985.
15. Bureau of transportation statistics
www.transtats.bts.gov Accessed Nov 2002.
16. American travel survey (ATS) 1995.
http://www.transtats.bts.gov/Tables.asp?DB_ID=505&DB_Name=American%20Travel%20Survey%20%28ATS%29%201995&DB_Short_Name=ATS. Accessed Nov 2002.
17. Airline Origin and Destination survey (DB1B).
http://www.transtats.bts.gov/Tables.asp?DB_ID=125&DB_Name=Airline%20Origin%20and%20Destination%20Survey%20%28DB1B%29&DB_Short_Name=Origin%20and%20Destination%20Survey. Accessed Nov 2002.
18. Clustering Techniques.
http://dms.irb.hr/tutorial/tut_clustering_short.php. Accessed Jul 2003.

11 APPENDICES

11.1 Appendix A – GLOSSARY OF TERMS

ATS - American Travel Survey

AI - Artificial Intelligence

ANN - Artificial Neural Networks

BTS - Bureau of Transportation Statistics

IIA - Independence from Irrelevant Alternatives

MLP - Multilayer Perceptron

CA - Commercial Airline

GA - General Aviation

CV - Cross Validation data

MSE - Mean Square Error

ARV - Average Relative Variance

AR - Absolute Residual

FAA - Federal Aviation Administration

11.2 Appendix B – MATLAB CODES

1) Complete_code_ann_total.m

```
% PRAVEEN EDARA, 19 MAY 2003, VIRGINIA TECH
% The output of this program is used to develop mode choice model by using neural nets.
% Data file(1) : logitData_business_all.mat for business trip
% Data file(2) : logitData_nonBusiness_all.mat for Non-business trip
```

```
% This file is a struct array with the following fields
% hhIncome    : personal income
% tripDist1way: trip distance
% TT_com      : travel time if he uses commercial airline
% TT_GA       : travel time if he uses general aviation
% TT_Auto     : travel time if he uses auto
% TC_com      : travel cost if he uses commercial airline
% TC_GA       : travel cost if he uses general aviation
% TC_Auto     : travel cost if he uses auto
% mode        : 1-auto 2-commercial airline 3-GA
```

```
% data handling
% compute the value of delta_T = min(TT_GA,TT_Auto) - TT_com
% compute the value of delta_C = min(TC_GA,TC_Auto) - TC_com
```

```
load logitData_business_all.mat;
```

```
% get the size of structure array
getsize = size(logitData_business_all);
size_data = getsize(1,2);
```

```
for i = 1:size_data
    data_business_total(i)=struct('income',logitData_business_all(i).hhIncome...
        ,'distance',logitData_business_all(i).tripDist1way...

        ,'delta_T_1_Auto',(min(logitData_business_all(i).TT_GA,logitData_business_all(i).TT_c
om)-logitData_business_all(i).TT_Auto)...

        ,'delta_T_1_Com',(min(logitData_business_all(i).TT_GA,logitData_business_all(i).TT_
Auto)-logitData_business_all(i).TT_com)...

        ,'delta_T_1_GA',(min(logitData_business_all(i).TT_Auto,logitData_business_all(i).TT_c
om)-logitData_business_all(i).TT_GA)...

        ,'delta_C_1_Auto',(min(logitData_business_all(i).TC_GA,logitData_business_all(i).TC_c
om)-logitData_business_all(i).TC_Auto)...

        ,'delta_C_1_Com',(min(logitData_business_all(i).TC_GA,logitData_business_all(i).TC_
```

```

Auto)-logitData_business_all(i).TC_com)...
,'delta_C_1_GA',(min(logitData_business_all(i).TC_Auto,logitData_business_all(i).TC_c
om)-logitData_business_all(i).TC_GA)...
,'mode',logitData_business_all(i).mode);
end

clear logitData_business_all;

save data_business_total.mat data_business_total;

% New data files came from the Data files
% New data file(1) : data_business.mat
% New data file(2) : data_nonBusiness.mat

% These files are struct array having the field as follows,
% income : personal income
% distance : trip distance
% delta_T : min(TT_GA,TT_Auto) - TT_com
% delta_C : min(TC_GA,TC_Auto) - TC_com
% mode : 1-auto 2-commercial airline 3-GA

data_size = size(data_business_total);
data_size = data_size(1,2);

for i=1:1:data_size
    data_business_matrix_nan(i,1) = data_business_total(i).income;
    data_business_matrix_nan(i,2) = data_business_total(i).distance;
    data_business_matrix_nan(i,3) = data_business_total(i).delta_T_1_Auto;
    data_business_matrix_nan(i,4) = data_business_total(i).delta_T_1_Com;
    data_business_matrix_nan(i,5) = data_business_total(i).delta_T_1_GA;
    data_business_matrix_nan(i,6) = data_business_total(i).delta_C_1_Auto;
    data_business_matrix_nan(i,7) = data_business_total(i).delta_C_1_Com;
    data_business_matrix_nan(i,8) = data_business_total(i).delta_C_1_GA;
    data_business_matrix_nan(i,9) = data_business_total(i).mode;
end

% delete row if that row contains the NaN in third column!
for i=1:1:data_size
    if(~sum(isnan(data_business_matrix_nan(i,:))))
        data_business_matrix_total(i,:) = data_business_matrix_nan(i,:);
    end
end

clear data_business_matrix_nan;

```

```

save data_business_matrix_total.mat data_business_matrix_total;
% clear data_business_matrix_nan;
% save data_business_matrix.mat data_business_matrix;

data_size_1 = size(data_business_matrix_total);
data_size_1 = data_size_1(1,1);

load IDX1;
total_cluster1 = data_business_matrix_total(1:20000,1:9);
total_cluster1(:,10) = IDX1(1:20000,1);

load IDX2;
total_cluster2 = data_business_matrix_total(20001:40000,1:9);
total_cluster2(:,10) = IDX2(1:20000,1);

load IDX3;
total_cluster3 = data_business_matrix_total(40001:60000,1:9);
total_cluster3(:,10) = IDX3(1:20000,1);

load IDX4;
total_cluster4 = data_business_matrix_total(60001:80000,1:9);
total_cluster4(:,10) = IDX4(1:20000,1);

load IDX5;
total_cluster5 = data_business_matrix_total(80001:100000,1:9);
total_cluster5(:,10) = IDX5(1:20000,1);

load IDX6;
total_cluster6 = data_business_matrix_total(100001:data_size_1,1:9);
total_cluster6(:,10) = IDX6(1:18355,1);

save total_cluster1.mat total_cluster1;
save total_cluster2.mat total_cluster2;
save total_cluster3.mat total_cluster3;
save total_cluster4.mat total_cluster4;
save total_cluster5.mat total_cluster5;
save total_cluster6.mat total_cluster6;

% Let us create Artificial Passengers.
% One artificial passenger per one cluster number. His attributes are the average values
of all the people belonging to that cluster number.

% The fifth column of data_business_clustering matrices is mode choice. It
% has 3 possible values 1-Auto, 2-Commercial Airline, 3-GA. Since we need

```

```
% to know the probability of an artificial passenger choosing Commercial Airline for his  
% trip,  
% Artificial passengers for the each cluster can be obtained as below
```

```
data_processed1_total= artpass_ann(total_cluster1,20000);  
data_processed2_total= artpass_ann(total_cluster2,20000);  
data_processed3_total= artpass_ann(total_cluster3,20000);  
data_processed4_total= artpass_ann(total_cluster4,20000);  
data_processed5_total= artpass_ann(total_cluster5,20000);  
data_processed6_total= artpass_ann(total_cluster6,18355); % The size is obtained from  
running the main1_vector
```

```
% The next step is to use this artificial passenger data as input to a  
% neural network model...for working with neural nets we would be using a  
% commercial software called Neuro Solutions.
```

2) Ann_total.m

```
% This function calculates the attributes of artificial passenger, there
% are 9 attributes for each art passengenger,
function data_processed=artpass_ann(data_business_clustering,n)
for i=1:1:200

sum1=0;sum2=0;sum3=0;sum4=0;sum5=0;sum6=0;sum7=0;sum8=0;hit1=0;hit2=0;hit3=
0;count=0;
    for j=1:1:n
        if (data_business_clustering(j,10) == i)

            sum1=sum1+data_business_clustering(j,1); % 1-Income, 2-Distance, 3,4,5-dT for
Auto,CA and GA respectively,
            sum2=sum2+data_business_clustering(j,2); % 6,7,8-dC for Auto,CA and GA
respectively
            sum3=sum3+data_business_clustering(j,3);
            sum4=sum4+data_business_clustering(j,4);
            sum5=sum5+data_business_clustering(j,5);
            sum6=sum6+data_business_clustering(j,6);
            sum7=sum7+data_business_clustering(j,7);
            sum8=sum8+data_business_clustering(j,8);

            if (data_business_clustering(j,9)==1)
                hit1=hit1+1;
            end

            if (data_business_clustering(j,9)==2)
                hit2=hit2+1;
            end

            if (data_business_clustering(j,9)==3)
                hit3=hit3+1;
            end
            count=count+1;
            tot=[sum1 sum2 sum3 sum4 sum5 sum6 sum7 sum8 hit1 hit2 hit3];
        end
    end
    data_processed(i,:)=tot./count;
end
data_processed;
```

3) Complete_code_logit.m

```
% PRAVEEN EDARA, 19 MAY 2003, VIRGINIA TECH
% The output of this program is used to develop mode choice models using-
% -Multinomial Logit calibration procedure in SAS.
% Data file(1) : logitData_business_all.mat for business trip
% Data file(2) : logitData_nonBusiness_all.mat for Non-business trip

% These file is struct array having the field as follows,
% hhIncome    : personal income
% tripDist1way : trip distance
% TT_com      : travel time if he uses commercial airline
% TT_GA       : travel time if he uses general aviation
% TT_Auto     : travel time if he uses auto
% TC_com      : travel cost if he uses commercial airline
% TC_GA       : travel cost if he uses general aviation
% TC_Auto     : travel cost if he uses auto
% mode        : 1-auto 2-commercial airline 3-GA

% data handling
% compute the value of  $\Delta T = \min(TT\_GA, TT\_Auto) - TT\_com$ 
% compute the value of  $\Delta C = \min(TC\_GA, TC\_Auto) - TC\_com$ 

load logitData_business_all.mat;

% get the size of structure array
getsize = size(logitData_business_all);
size_data = getsize(1,2);

for i = 1:size_data
    data_business_logit(i)=struct('income1',logitData_business_all(i).hhIncome...
        , 'income2',logitData_business_all(i).hhIncome...
        , 'income3',logitData_business_all(i).hhIncome...
        , 'distance1',logitData_business_all(i).tripDist1way...
        , 'distance2',logitData_business_all(i).tripDist1way...
        , 'distance3',logitData_business_all(i).tripDist1way...

        , 'delta_T_1_Auto',(min(logitData_business_all(i).TT_GA,logitData_business_all(i).TT_c
om)-logitData_business_all(i).TT_Auto)...

        , 'delta_T_1_Com',(min(logitData_business_all(i).TT_GA,logitData_business_all(i).TT_
Auto)-logitData_business_all(i).TT_com)...

        , 'delta_T_1_GA',(min(logitData_business_all(i).TT_Auto,logitData_business_all(i).TT_c
om)-logitData_business_all(i).TT_GA)...
```

```

,'delta_C_1_Auto',(min(logitData_business_all(i).TC_GA,logitData_business_all(i).TC_c
om)-logitData_business_all(i).TC_Auto)...

,'delta_C_1_Com',(min(logitData_business_all(i).TC_GA,logitData_business_all(i).TC_
Auto)-logitData_business_all(i).TC_com)...

,'delta_C_1_GA',(min(logitData_business_all(i).TC_Auto,logitData_business_all(i).TC_c
om)-logitData_business_all(i).TC_GA)...
    ,'mode',logitData_business_all(i).mode);
end

clear logitData_business_all;

save data_business_logit.mat data_business_logit;

% New data files came from the Data files
% New data file(1) : data_business.mat
% New data file(2) : data_nonBusiness.mat

% These files are struct array having the field as follows,
% income : personal income
% distance : trip distance
% delta_T : min(TT_GA,TT_Auto) - TT_com
% delta_C : min(TC_GA,TC_Auto) - TC_com
% mode : 1-auto 2-commercial airline 3-GA

data_size = size(data_business_logit);
data_size = data_size(1,2);

for i=1:1:data_size
    data_business_matrix_nan(i,1) = data_business_logit(i).income1;
    data_business_matrix_nan(i,2) = data_business_logit(i).income2;
    data_business_matrix_nan(i,3) = data_business_logit(i).income3;
    data_business_matrix_nan(i,4) = data_business_logit(i).distance1;
    data_business_matrix_nan(i,5) = data_business_logit(i).distance2;
    data_business_matrix_nan(i,6) = data_business_logit(i).distance3;
    data_business_matrix_nan(i,7) = data_business_logit(i).delta_T_1_Auto;
    data_business_matrix_nan(i,8) = data_business_logit(i).delta_T_1_Com;
    data_business_matrix_nan(i,9) = data_business_logit(i).delta_T_1_GA;
    data_business_matrix_nan(i,10) = data_business_logit(i).delta_C_1_Auto;
    data_business_matrix_nan(i,11) = data_business_logit(i).delta_C_1_Com;
    data_business_matrix_nan(i,12) = data_business_logit(i).delta_C_1_GA;
    data_business_matrix_nan(i,13) = data_business_logit(i).mode;
end

```

```

% delete row if that row contains the NaN in third column!
for i=1:1:data_size
    if(~sum(isnan(data_business_matrix_nan(i,:))))
        data_business_matrix_logit(i,:) = data_business_matrix_nan(i,:);
    end
end

clear data_business_matrix_nan;
save data_business_matrix_logit.mat data_business_matrix_logit;

% Calibration data to be used for the logit model
% Using vectors rather than for loops

% The reason for splitting data_business_matrix_logit into 22 parts is
% -that, matlab cannot display more than 65536 variables as output. so the
% -idea is to split into small sets whose values can be displayed in matlab
% -and thereafter we can copy them to excel and merge these sets in such a
% -way that three groups of 12 sets, 6sets and 4 sets are formed -
% -first two are used for calibration of the logit model and the third set
% -is used to check the estimation capability of the model to unknown data.

datasize_logit = size(data_business_matrix_logit);
datasize=datasize_logit(1,1);
calibration_data1 = data_business_matrix_logit(1:5000,:);
calibration_data2 = data_business_matrix_logit(5001:10000,:);
calibration_data3 = data_business_matrix_logit(10001:15000,:);
calibration_data4 = data_business_matrix_logit(15001:20000,:);
calibration_data5 = data_business_matrix_logit(20001:25000,:);
calibration_data6 = data_business_matrix_logit(25001:30000,:);
calibration_data7 = data_business_matrix_logit(30001:35000,:);
calibration_data8 = data_business_matrix_logit(35001:40000,:);
calibration_data9 = data_business_matrix_logit(40001:45000,:);
calibration_data10 = data_business_matrix_logit(45001:50000,:);
calibration_data11 = data_business_matrix_logit(50001:55000,:);
calibration_data12 = data_business_matrix_logit(55001:60000,:);
calibration_data13 = data_business_matrix_logit(60001:65000,:);
calibration_data14 = data_business_matrix_logit(65001:70000,:);
calibration_data15 = data_business_matrix_logit(70001:75000,:);
calibration_data16 = data_business_matrix_logit(75001:80000,:);
calibration_data17 = data_business_matrix_logit(80001:85000,:);
calibration_data18 = data_business_matrix_logit(85001:90000,:);
calibration_data19 = data_business_matrix_logit(90001:95000,:);
calibration_data20 = data_business_matrix_logit(95001:100000,:);
calibration_data21 = data_business_matrix_logit(100001:105000,:);
calibration_data22 = data_business_matrix_logit(105001:110000,:);

```

```
% Some of the data were missing so they were filled with zeros initially
% we need to eliminate all these data points.
% Use of a function non_zero speeds up the procedure
data1=non_zero(calibration_data1);
data2=non_zero(calibration_data2);
data3=non_zero(calibration_data3);
data4=non_zero(calibration_data4);
data5=non_zero(calibration_data5);
data6=non_zero(calibration_data6);
data7=non_zero(calibration_data7);
data8=non_zero(calibration_data8);
data9=non_zero(calibration_data9);
data10=non_zero(calibration_data10);
data11=non_zero(calibration_data11);
data12=non_zero(calibration_data12);
data13=non_zero(calibration_data13);
data14=non_zero(calibration_data14);
data15=non_zero(calibration_data15);
data16=non_zero(calibration_data16);
data17=non_zero(calibration_data17);
data18=non_zero(calibration_data18);
data19=non_zero(calibration_data19);
data20=non_zero(calibration_data20);
data21=non_zero(calibration_data21);
data22=non_zero(calibration_data22);
```

4) **Non_zero.m**

% This function is used by the above code to remove the missing data

```
function data = non_zero(calibration_data)
```

```
x=0;
```

```
for i=1:1:5000
```

```
if (calibration_data(i,13)~=0)
```

```
    x=x+1;
```

```
    data(x,:)=calibration_data(i,:);
```

```
end
```

```
end
```

5) Complete_code.m

```
% PRAVEEN EDARA, Dt:19 MAY 2003, VIRGINIA TECH
% This output of this program is used to develop mode choice model for commercial-
% -airline mode using neural networks in Neuro solutions software
% Data file(1) : logitData_business_all.mat for business trip
% Data file(2) : logitData_nonBusiness_all.mat for Non-business trip
```

```
% These files are struct array having the field as follows,
% hhIncome    : personal income
% tripDist1way: trip distance
% TT_com      : travel time if he uses commercial airline
% TT_GA       : travel time if he uses general aviation
% TT_Auto     : travel time if he uses auto
% TC_com      : travel cost if he uses commercial airline
% TC_GA       : travel cost if he uses general aviation
% TC_Auto     : travel cost if he uses auto
% mode        : 1-auto 2-commercial airline 3-GA
```

```
load logitData_business_all.mat;
```

```
% get the size of structure array
getsize = size(logitData_business_all);
size_data = getsize(1,2);
```

```
% make the new data having the fields
% Income, distance, delta_T, delta_C, mode
```

```
for i = 1:size_data
    data_business(i)=struct('income',logitData_business_all(i).hhIncome...
        , 'distance',logitData_business_all(i).tripDist1way...
        , 'delta_T',(min(logitData_business_all(i).TT_GA,logitData_business_all(i).TT_Auto)-
logitData_business_all(i).TT_com)...
        , 'delta_C',(min(logitData_business_all(i).TC_GA,logitData_business_all(i).TC_Auto)-
logitData_business_all(i).TC_com)...
        , 'mode',logitData_business_all(i).mode);
end
```

```
clear logitData_business_all;
```

```
save data_business.mat data_business;
```

```
% New data file(1) : data_business.mat
% New data file(2) : data_nonBusiness.mat
```

```

% These files are struct array having the field as follows,
% income : personal income
% distance : trip distance
% delta_T : min(TT_GA,TT_Auto) - TT_com
% delta_C : min(TC_GA,TC_Auto) - TC_com
% mode : 1-auto 2-commercial airline 3-GA

% This part of the code converts the structure array to double array and
% process the data inorder to apply clustering algorithm in the next step

data_size = size(data_business);
data_size = data_size(1,2);

for i=1:1:data_size
    data_business_matrix_nan(i,1) = data_business(i).income;
    data_business_matrix_nan(i,2) = data_business(i).distance;
    data_business_matrix_nan(i,3) = data_business(i).delta_T;
    data_business_matrix_nan(i,4) = data_business(i).delta_C;
    data_business_matrix_nan(i,5) = data_business(i).mode;
end

% delete row if that row contains the NaN in third column!
for i=1:1:data_size
    if(~sum(isnan(data_business_matrix_nan(i,:))))
        data_business_matrix(i,:) = data_business_matrix_nan(i,:);
    end
end

clear data_business_matrix_nan;
save data_business_matrix.mat data_business_matrix;

data_size_1 = size(data_business_matrix);
data_size_1 = data_size_1(1,1);

clear data_business;

% Let us divide the data into 6 subsets. In the future we will be applying clustering
algorithm to each of these 6 subsets.

data_business_clustering1 = data_business_matrix(1:20000,1:4);
data_business_clustering2 = data_business_matrix(20001:40000,1:4);
data_business_clustering3 = data_business_matrix(40001:60000,1:4);
data_business_clustering4 = data_business_matrix(60001:80000,1:4);
data_business_clustering5 = data_business_matrix(80001:100000,1:4);
data_business_clustering6 = data_business_matrix(100001:data_size_1,1:4);

```

```

save data_business_clustering1.mat data_business_clustering1;
save data_business_clustering2.mat data_business_clustering2;
save data_business_clustering3.mat data_business_clustering3;
save data_business_clustering4.mat data_business_clustering4;
save data_business_clustering5.mat data_business_clustering5;
save data_business_clustering6.mat data_business_clustering6;

% This part of the code performs "CLUSTERING DATA"
% This code clusters each of the 6subset matrices. kmeans clustering procedure is
applied.
% After clustering is done, all the data_business_clustering matrices have only 4
columns. We need-
% -to include the fifth(the output mode choice) and sixth(Number of cluster) column
also.

% "Clustering data based on k-means algorithm" followed by the "Creation of complete
data_business_clustering matrix including mode choice and cluster label variables "

IDX1 = KMEANS(data_business_clustering1,200,'EmptyAction','singleton');
data_business_clustering1(:,5) = data_business_matrix(1:20000,5);
data_business_clustering1(:,6) = IDX1(1:20000,1);

IDX2 = KMEANS(data_business_clustering2,200,'EmptyAction','singleton');
data_business_clustering2(:,5) = data_business_matrix(20001:1:40000,5);
data_business_clustering2(:,6) = IDX2(1:20000,1);

IDX3 = KMEANS(data_business_clustering3,200,'EmptyAction','singleton');
data_business_clustering3(:,5) = data_business_matrix(40001:1:60000,5);
data_business_clustering3(:,6) = IDX3(1:20000,1);

IDX4 = KMEANS(data_business_clustering4,200,'EmptyAction','singleton');
data_business_clustering4(:,5) = data_business_matrix(60001:1:80000,5);
data_business_clustering4(:,6) = IDX4(1:20000,1);

IDX5 = KMEANS(data_business_clustering5,200,'EmptyAction','singleton');
data_business_clustering5(:,5) = data_business_matrix(80001:1:100000,5);
data_business_clustering5(:,6) = IDX5(1:20000,1);

IDX6 = KMEANS(data_business_clustering6,200,'EmptyAction','singleton');
data_business_clustering6(:,5) = data_business_matrix(100001:1:data_size_1,5);
data_business_clustering6(:,6) = IDX6(1:18355,1);

% Let us create Artificial Passengers.
% One artificial passenger per one cluster number. His attributes are the average values
of all the people belonging to that cluster number.

```

```
% The fifth column of data_business_clustering matrices is mode choice. It
% has 3 possible values 1-Auto, 2-Commercial Airline, 3-GA. Since we need
% to know the probability of an artificial passenger choosing Commercial Airline for his
trip,
```

```
% Artificial passengers for the each cluster can be obtained as below
```

```
data_processed1= artpass(data_business_clustering1,20000); % 'Artpass' is a function
that calculates the values of the variables for each artificial passenger
data_processed2= artpass(data_business_clustering2,20000);
data_processed3= artpass(data_business_clustering3,20000);
data_processed4= artpass(data_business_clustering4,20000);
data_processed5= artpass(data_business_clustering5,20000);
data_processed6= artpass(data_business_clustering6,18355); % The size is obtained from
running the main1_vector
```

```
% The next step is to use this artificial passenger data as input to a
% neural network model...for working with neural nets we would be using a
% commercial software called Neuro Solutions.
```

6) Artpass.m

```
% This function calculates the attributes of artificial passenger for commercial airline
% mode
% (attributes include income, distance, travel time diff and travel cost differential and
% mode chosen)
function data_processed=artpass(data_business_clustering,n)
for i=1:1:200
    sum1=0;sum2=0;sum3=0;sum4=0;hit=0;count=0;
    for j=1:1:n
        if (data_business_clustering(j,6) == i)
            sum1=sum1+data_business_clustering(j,1);
            sum2=sum2+data_business_clustering(j,2);
            sum3=sum3+data_business_clustering(j,3);
            sum4=sum4+data_business_clustering(j,4);
            if (data_business_clustering(j,5)==2)
                hit=hit+1;
            end
            count=count+1;
            tot=[sum1 sum2 sum3 sum4 hit];
        end
    end
    data_processed(i,:)=tot./count;
end
data_processed;
```

11.3 Appendix C- SAS OUTPUTS

The SAS System

The MDC Procedure

Conditional Logit Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t value	Approx Pr > t	Gradient
Income	0	3.406E-6	.	.	.	-1.89E-9
Distance	0	0.000446	.	.	.	-204E-13
dC	1	0.008344	0.000009	84.05	<.0001	-0.41961
dT	1	-0.13220	0.006824	-19.37	<.0001	-0.00639

The SAS System

The MDC Procedure

Conditional Logit Estimates

Model Fit Summary

Goodness of Fit Measures for Discrete Choice Models

Measure	Value	Formula
Likelihood Ratio (R)	18516	$2*(\text{LogL}-\text{LogL0})$
Upper Bound of R (U)	32954	$-2*\text{LogL0}$
Aldrich-Nelson	0.5525	$R/(R+N)$
Cragg-Uhler 1	0.7090	$1-\exp(-R/N)$
Cragg-Uhler 2	0.7977	$(1-\exp(-R/N)) / (1-\exp(-U/N))$
Estrella	0.8369	$1-(1-R/U)^{(U/N)}$
Adjusted Estrella	0.8367	$1-((\text{LogL}-K)/\text{LogL0})^{(-2/N*\text{LogL0})}$
McFadden's LRI	0.5619	R/U
Veall-Zimmermann	0.8039	$(R*(U+N)) / (U*(R+N))$

N = # of observations, K= # of regressors

12 VITA

Praveen Kumar Edara was born on 29th February 1980 in the state of Andhra Pradesh, India. He has done all his schooling and college in Atomic Energy Central School, Hyderabad. Praveen did his Bachelors in Technology in Civil Engineering from Indian Institute of Technology, Madras. He joined the Masters program in Civil Engineering at Virginia Polytechnic Institute and State University in August 2002.

Praveen would be continuing his studies at Virginia Tech in the pursuit of Doctoral degree in Civil Engineering under the supervision of Dr. Dušan Teodorović and Dr. John Collura.