

PROJECT REPORT
ON
UNIVERSITY PREDICTION SYSTEM
(SecureUni)

TABLE OF CONTENTS

Title Fly	Pages
Table of Contents	I
List of Figures	II
1. Chapter 1	1
1.1. Introduction	1
1.2. Problem Statement	1
1.3. Scope	2
1.4. Applications	2
2. Chapter 2	4
2.1. Literature Survey	4
2.2. Techniques Used	5
2.3. Software Requirements	8
2.4. Hardware Requirements	8
2.5. Functional Requirements	8
2.6. Design	9
3. Chapter 3	11
3.1 Implementation	11
4. Chapter 4	17
4.1. Conclusion	17
4.2. Future Scope	18
5. Chapter 5: References	19

LIST OF FIGURES

Fig 2.1: Sigmoid Function	7
Fig 2.2: Logarithmic transformation for $y=0$	7
Fig 2.3: Logarithmic transformation for $y=1$	7
Fig 2.4: Use case model	10
Fig 3.1: Dataset of recorded scores	11
Fig 3.2: Calculation of weights	12
Fig 3.3: Logistic Regression model	13
Fig 3.4: Prediction and Sigmoid Function	13
Fig 3.5: Theta Values	14
Fig 3.6: Substituting theta as weights	15
Fig 3.7: JavaScript input form	15
Fig 3.8: Prediction Results	16
Fig 3.9: Model Evaluation	16

CHAPTER 1

1.1. INTRODUCTION

A student while applying to a university faces many difficulties in deciding which universities to apply to. The fees for an application are too high and one does not want to waste their money in applying to a university where the chances of getting in are bleak. Also, the counselling provided by various companies is not so easy on the pocket. So, our University Assist System helps users get over this dilemma. We accept the scores of the user and based on these scores give them a realistic picture of where they stand. We give them the probabilities of getting into various universities so as to help them choose which university to apply to wisely. The user has to login and input his scores. As soon as the user makes the payment, an email is sent which gives him the overall review of his resume.

1.2. PROBLEM STATEMENT

Design an application/website which will help students find colleges worldwide based on their budget, percentage and their scores of internationally identified tests (GRE, SOP, etc.). All these factors should be taken into consideration to find out in which of the topmost colleges they are eligible for. This will help save students' time and will give all the relevant as well as best results.

1.3. SCOPE

This project traverses a lot of areas ranging from business concept to computing field. The area covers include:

- Studying what factors different universities give weightage to decide whether or not to admit an applicant.
- Logistic Regression used for the development of the application.
- General customers and the company's staff will be able to use the system effectively.
- Web-platform means that the system will be available for access 24/7.

1.4. APPLICATIONS

Users will be able to login and enter their credentials such as their GRE scores, their CGPA, their Statement of Purpose, their Work Experience, their budget and choice of country to see their chances of getting into the different universities of that region. The consumer will be able to select and pay for the service using credit/debit cards and E-Wallets. As soon as payment is made, a mail shall be sent to the user that will show him where he stands.

This is very useful to the students who are planning for studies abroad, as it will give them a clearer picture of where they stand with the score that they have obtained. The aim of this project is to be a helping hand to the student to work towards knowing how much more effort one needs to put in to get into their dream university. For students who do not even know the various top universities which they can apply to, and this project helps them by showing them all the 4 tiers of universities.

This project also helps the student understand the difference in the weightage of each score used to apply for a post graduate school abroad. Some universities give equal weightage to all aspects, whereas some others don't. The variation of the chances of

one getting in also depends on their budget, which is by far the most important aspect from the student point of view as most Indian students are looking for scholarships to complete their studies in these universities.

CHAPTER 2

2.1. LITERATURE SURVEY

Predicting Student Enrolment Based on Student and College Characteristics – Ahmad Slim, Don Hush, Tushar Ojah, Terry Babbitt.^[1]

Colleges are increasingly interested in identifying the factors that maximize their enrolment. These factors allow enrolment management administrators to identify the applicants who have higher tendency to enrol at their institutions and accordingly to better allocate their money rewards (i.e., scholarship and financial aid). In this paper we identify factors that affect the likelihood of enrolling. We use machine learning methods to statistically analyze the enrolment predictability of such factors. In particular, we use logistic regression (LR), support vector machines (SVMs) and semi-supervised probability methods. The LR and the SVMs methods predict the enrolment of applicants at an individual level whereas the semi-supervised probability method does that at a cohort level ^[6]. We validate our methods using real data for applicants admitted to the university of New Mexico (UNM). The results show that a small set of factors related to student and college characteristics are highly correlated to the applicant decision of enrolment. This outcome is supported by the relatively high prediction accuracy of the proposed methods.

GRADE: Machine Learning Support for Graduate Admissions - Austin Waters and Risto Miikkulainen^[2]

This paper describes GRADE, a statistical machine learning system developed to support the work of the graduate admissions committee at the University of Texas at Austin Department of Computer Science (UTCS). In recent years, the number of applications to the UTCS PhD program has become too large to manage with a

traditional review process. GRADE uses historical admissions data to predict how likely the committee is to admit each new applicant. It reports each prediction as a score similar to those used by human reviewers, and accompanies each by an explanation of what applicant features most influenced its prediction. GRADE makes the review process more efficient by enabling reviewers to spend most of their time on applicants near the decision boundary and by focusing their attention on parts of each applicant's file that matter the most. An evaluation over two seasons of PhD admissions indicates that the system leads to dramatic time savings, reducing the total time spent on reviews by at least 74%^[3]

2.2. TECHNIQUES USED

Logistic Regression

In statistics, the logistic model (or logit 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 (or logit regression) is estimating the parameters of a logistic model; it is a form of binomial regression. Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail, win/lose, alive/dead or healthy/sick; these are represented by an indicator variable, where the two values are labelled "0" and "1". In the logistic model, the log-odds (the logarithm of the odds) for the value labelled "1" is a linear combination of one or more independent variables ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value)^[7]. The corresponding probability of the value labelled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labelling; the function that converts log-odds to probability is the logistic function, hence the name. The unit of measurement for the log-odds scale is called a *logit*, from *logistic unit*, hence the alternative names. Analogous models with a different sigmoid function instead of the logistic function can also be used, such as the probit model; the defining characteristic of the logistic model is that increasing one

of the independent variables multiplicatively scales the odds of the given outcome at a *constant* rate, with each dependent variable having its own parameter; for a binary independent variable this generalizes the odds ratio.^[4]

The cost function for logistic regression:

$$\begin{aligned}
 J(\theta) &= \frac{1}{m} \sum_{i=1}^m \text{Cost}(h_{\theta}(x^{(i)}), y^{(i)}) \\
 &= -\frac{1}{m} \left[\sum_{i=1}^m y^{(i)} \log h_{\theta}(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_{\theta}(x^{(i)})) \right]
 \end{aligned}
 \dots(2.1)$$

Here,

$J(\Theta)$ is the cost you want the learning algorithm to pay if outcome is $h(x)$ and the actual outcome is 'y'.

$h(x)$ is also known as the sigmoid function or the logistic function.

To get a global minima using gradient descent we need a convex function,^[5]

$$\text{Cost}(h_{\theta}(x), y) = \begin{cases} -\log(h_{\theta}(x)) & \text{if } y = 1 \\ -\log(1 - h_{\theta}(x)) & \text{if } y = 0 \end{cases}$$

Note: $y = 0$ or 1 always

....(2.2)

The plotted graphs for the sigmoid function as well as the two logarithmic transformations are shown as follows:

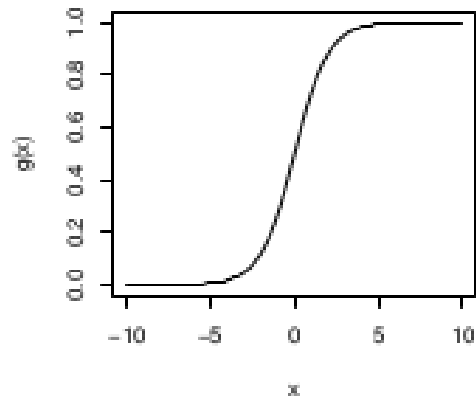


FIG 2.1: Sigmoid Function

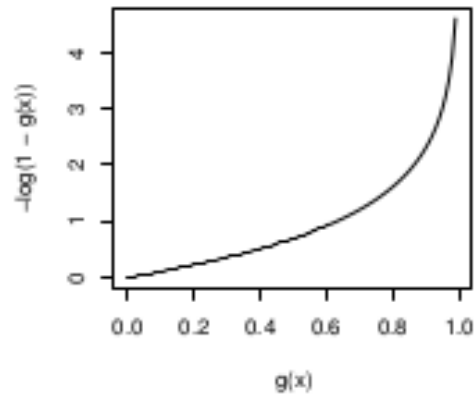


FIG 2.2: Logarithmic transformation for $y=0$

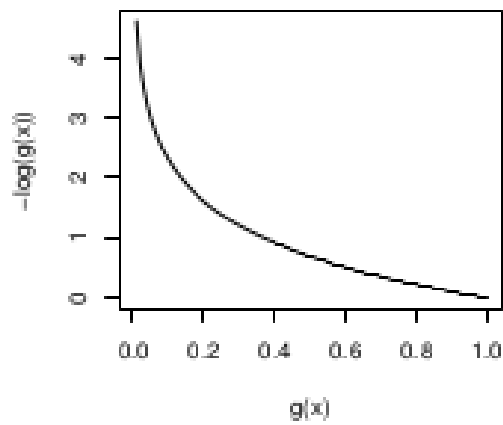


FIG 2.3: Logarithmic transformation for $y=1$

2.3. SOFTWARE REQUIREMENT

- The system is on server, so it requires any scripting language like JavaScript, PHP, etc.
- The system may also require a database to store any transaction of the system like MYSQL, etc.
- The system also requires DNS and for the user to have an active internet connection for browsing.

2.4. HARDWARE REQUIREMENT

- No extra hardware interfaces are needed.
- The system will use the standard hardware and data communication resources.

2.5. FUNCTIONAL REQUIREMENT

The functional requirements for this software are:

- Accept User Credentials: The website will accept the user's GRE score, CGPA, strength of SOP, preferred area, area of expertise, budget, TOEFL score, strength of LOR, E-mail.
- Prediction: A list of Universities along with the probabilities of the User getting into them will be mailed to the User.
- College Information: The website will also show the information of the college and also provide the cut-offs of past 3 years as well as give the link to the official website of the college.
- Feedback: The website will follow up with the User once they have secured admission to improve and update the performance of the prediction model.

2.6. DESIGN

University Assist System includes several functionalities described as below:

- User Login It allows the user to register and then login. The login credentials are verified and then the user is redirected to the home page.
- Checking for eligibility User can check whether he is eligible for admission in different universities based on his scores.
- Payment System Administrator/owner of the applications responsible for payment to the employee. An email is sent as soon as the payment is made.
- Maintenance Database Administrator maintains the data in a DBMS and makes sure that there are no redundancies

USE CASE MODEL

Use case model for the project is as given below:

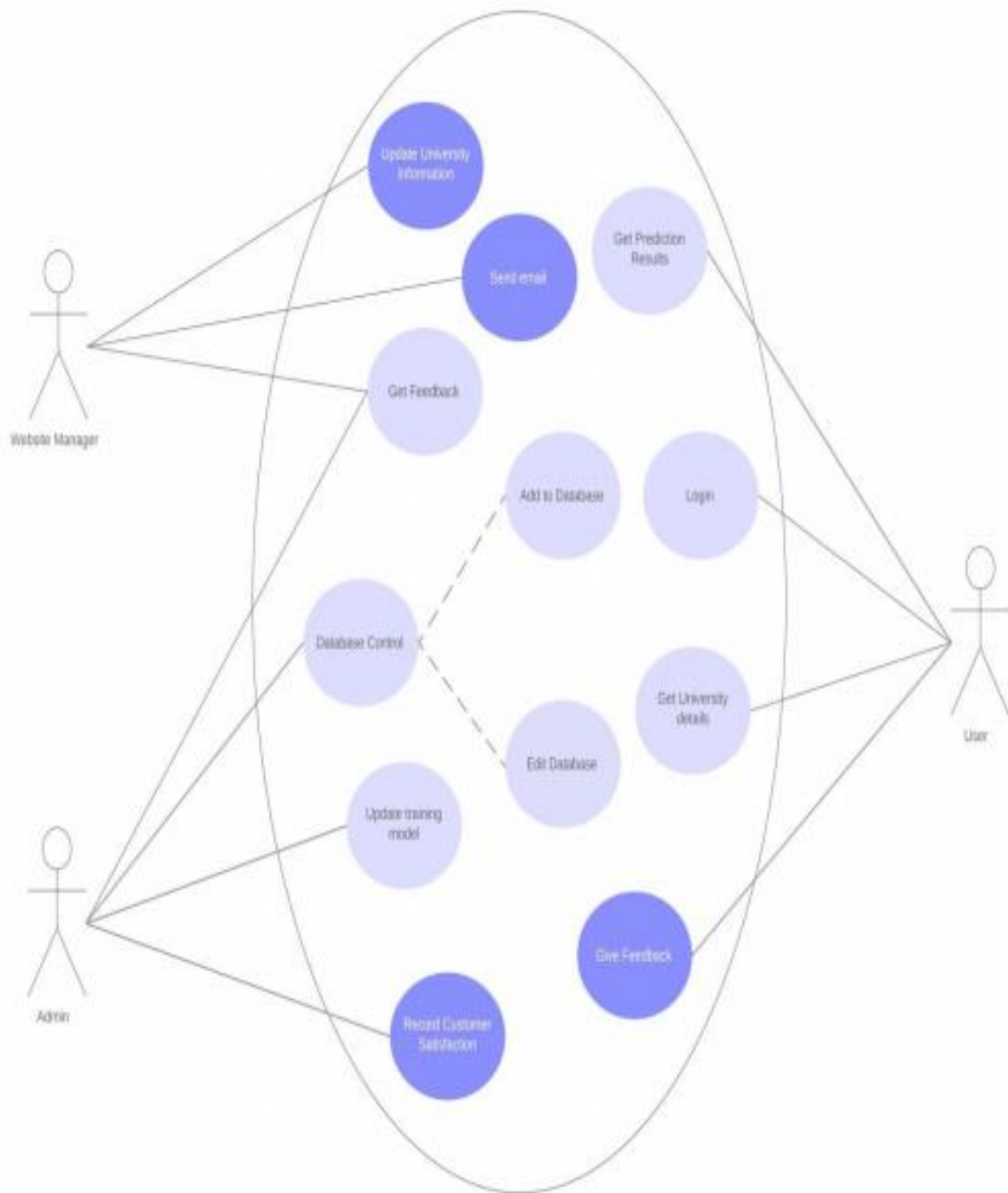


FIG 2.4: USE CASE MODEL

CHAPTER 3

3.1. IMPLEMENTATION

We started off by creating our own dataset consisting of 400 entries of GRE score, Overall GPA, Statement of Purpose level, Experience in years and the Budget. We then divided these individual components into 10 classes called ranks assigned from 1-10. Each entry is then used to calculate each university's score based on these values.

The universities are divided into 4 classes (tier-1, tier-2, tier-3, tier-4), each having their respective threshold criteria of the score. A persons' chances of getting into these universities are calculated using a cumulative score for each student, based on these parameters, using Normalization. We divided our set of 450 entries into 2 parts, i.e. 400 for training purposes and 50 for testing purposes.

	GRE Score		GPA		SOP		Experience		Funding	University 1		University 2		University 3		University 4		Final University score		
	score	rank	score	rank	score	rank	score	rank		funding %	Funding rank	funding %	Funding rank	funding %	Funding rank	funding %	Funding rank	Univ 1	Univ 2	Univ 3
21	317	7	9.3	9	5	10	3	6	20000	67%	7	50%	5	40%	4	44%	5	7.55	7.95	7.15
22	302	5	9.1	9	4	8	2	4	20000	67%	7	50%	5	40%	4	44%	5	6.55	6.55	6.25
23	340	10	9.8	10	5	10	3	6	40000	133%	10	100%	10	80%	8	89%	9	9.6	9.4	9.20
24	274	1	6.8	4	2	4	0	1	20000	67%	7	50%	5	40%	4	44%	5	4.15	2.9	3.25
25	298	4	8.1	7	3	6	2	4	20000	67%	7	50%	5	40%	4	44%	5	5.8	5.3	5.15
26	312	6	8.7	8	4	8	1	2	20000	67%	7	50%	5	40%	4	44%	5	6.5	6.3	6.10
27	285	3	7.7	6	3	6	2	4	30000	100%	10	75%	8	60%	6	67%	7	6.65	5.15	5.30
28	329	9	9	8	4	8	0	1	35000	117%	10	88%	9	70%	7	78%	8	8.35	7.3	7.55
29	299	5	7.9	6	3	6	2	4	30000	100%	10	75%	8	60%	6	67%	7	7.15	5.65	5.70
30	322	8	8.9	8	4	8	0	1	25000	83%	9	63%	7	50%	5	56%	6	7.7	6.85	6.75
31	290	3	7.2	5	2	4	2	4	30000	100%	10	75%	8	60%	6	67%	7	6.25	4.35	4.65
32	281	2	7.8	6	3	6	0	1	19000	63%	7	48%	5	38%	4	42%	5	4.9	4.15	4.35
33	324	8	9.5	9	5	10	1	2	35000	117%	10	88%	9	70%	7	78%	8	8.6	8	8.05
34	309	6	9	8	4	8	3	6	40000	133%	10	100%	10	80%	8	89%	9	8.1	7.4	7.50
35	276	1	6.2	3	1	2	1	2	30000	100%	10	75%	8	60%	6	67%	7	5.05	2.55	3.25
36	279	2	5.9	2	1	2	1	2	25000	83%	9	63%	7	50%	5	56%	6	4.8	2.5	2.90
37	297	4	7.9	6	2	4	1	2	35000	117%	10	88%	9	70%	7	78%	8	6.4	4.6	5.30
38	304	5	8.3	7	3	6	0	1	40000	133%	10	100%	10	80%	8	89%	9	6.95	5.6	6.40
39	315	7	8.8	8	4	8	0	1	45000	150%	10	113%	10	90%	9	100%	10	7.85	6.9	7.75
40	333	9	9.9	10	5	10	3	6	40000	133%	10	100%	10	80%	8	89%	9	8.15	8.85	8.80

FIG 3.1: Dataset of recorded scores

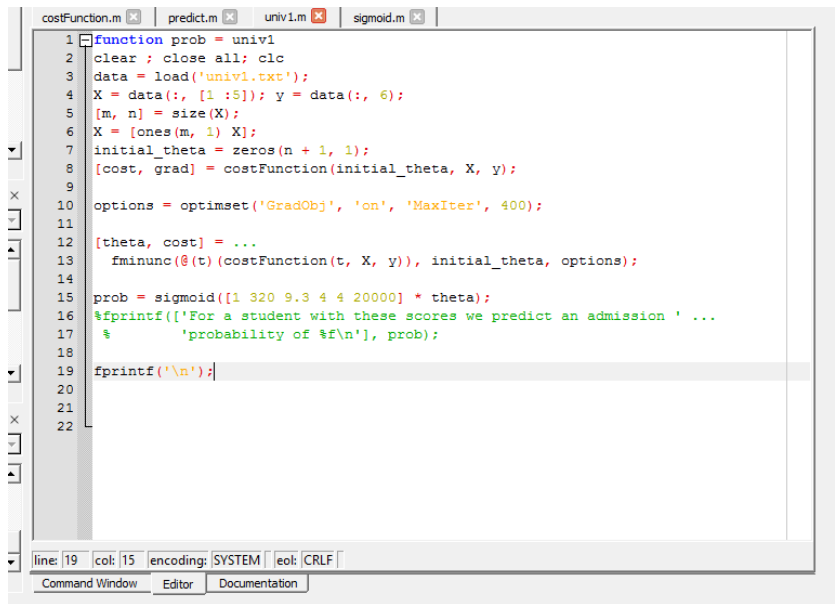
Each criterion is then given a score based on a range from 1-10. The total weight is then calculated using these criteria. Each university is given a different weight based on the tier it belongs to.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	GRE Score	GPA	SOP	Experience	Funding	Funding	Funding	Funding												
2	270	277	5	5.5	0	0.5	0	0.5		0	10.0%		10.0%		10.0%		10.0%			
3	277	284	5.5	6	0.5	1	0.5	1		10%	20%		20%		20%		20%			
4	284	291	6	6.5	1	1.5	1	1.5		20%	30%		30%		30%		30%			
5	291	298	6.5	7	1.5	2	1.5	2		30%	40%		40%		40%		40%			
6	298	305	7	7.5	2	2.5	2	2.5		40%	50%		50%		50%		50%			
7	305	312	7.5	8	2.5	3	2.5	3		50%	60%		60%		60%		60%			
8	312	319	8	8.5	3	3.5	3	3.5		60%	70%		70%		70%		70%			
9	319	326	8.5	9	3.5	4	3.5	4		70%	80%		80%		80%		80%			
10	326	333	9	9.5	4	4.5	4	4.5		80%	90%		90%		90%		90%			
11	333	340	9.5	10	4.5	5	4.5	100		90%	100%		100%		100%		100%			
12	total wt	GRE wt	GPA wt	SOP wt	Exp wt					Univ1 funding wt	Univ2 funding wt	Univ3 funding wt	Univ4 funding wt							
13	100%	25%	10%	15%	10%					40%										
14	100%	25%	20%	30%	15%						10%									
15	100%	20%	25%	20%	5%							30%								
16	100%	30%	25%	10%	10%								25%							
17																				
18										30000	40000	50000	45000							
19																				
20	GRE Score	GPA	SOP	Experience	Funding	University 1	University 2	University 3	University 4	Final University score										
21	score	rank	score	rank	score	rank	score	rank	score	rank	Univ 1	Univ 2	Univ 3							
22	217	7	8.3	8	6	10	3	6	30000	47%	7	60%	6	40%	8	44%	5	7.66	7.66	7.16

FIG 3.2: Calculation of Weights

We then optimize a Machine Learning algorithm to accurately predict suitable university for a particular student using Logistic Regression.

First, we load each university's data. The attributes are loaded into the variable X and their respective class is loaded into Y (1- getting admitted, 0-not getting admitted).

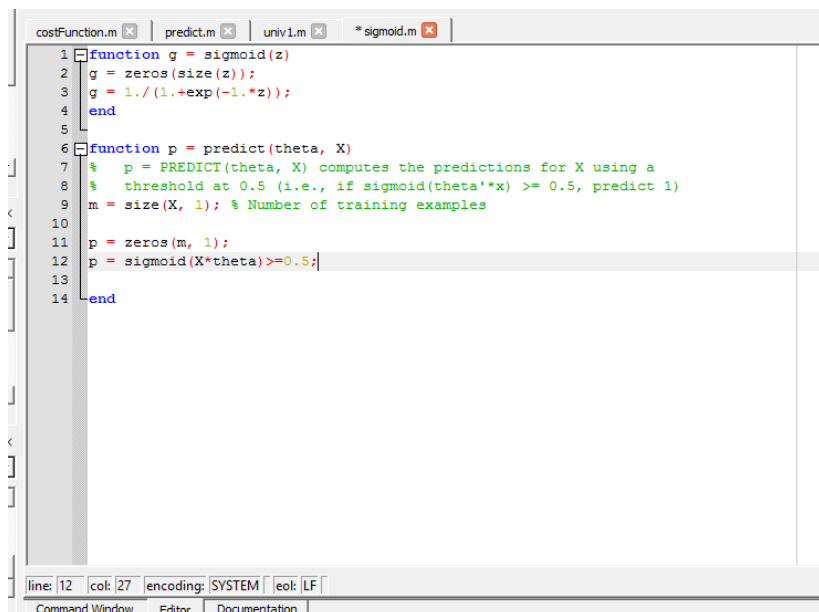


```

1 function prob = univ1
2 clear ; close all; clc
3 data = load('univ1.txt');
4 X = data(:, [1 :5]); y = data(:, 6);
5 [m, n] = size(X);
6 X = [ones(m, 1) X];
7 initial_theta = zeros(n + 1, 1);
8 [cost, grad] = costFunction(initial_theta, X, y);
9
10 options = optimset('GradObj', 'on', 'MaxIter', 400);
11
12 [theta, cost] = ...
13 fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);
14
15 prob = sigmoid([1 320 9.3 4 4 20000] * theta);
16 fprintf(['For a student with these scores we predict an admission ' ...
17         'probability of %f\n'], prob);
18
19 fprintf('\n');
20
21
22

```

FIG 3.3: Logistic Regression Model



```

1 function g = sigmoid(z)
2 g = zeros(size(z));
3 g = 1./(1.+exp(-1.*z));
4 end
5
6 function p = predict(theta, X)
7 % p = PREDICT(theta, X) computes the predictions for X using a
8 % threshold at 0.5 (i.e., if sigmoid(theta'*x) >= 0.5, predict 1)
9 m = size(X, 1); % Number of training examples
10
11 p = zeros(m, 1);
12 p = sigmoid(X*theta) >= 0.5;
13
14 end

```

FIG 3.4: Prediction and Sigmoid Functions

We calculate the values of theta to minimize our function, $J(\theta)$. The cost function will run 400 times and generate theta, resulting from minimization using gradient descent. Theta will be a matrix containing 6 values.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))]$$

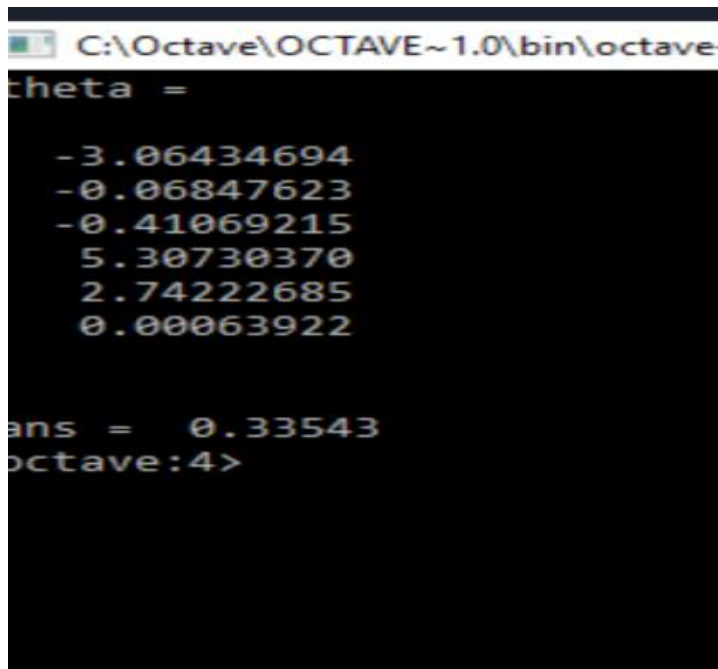
....(3.1)

Repeat {

$$\theta_j := \theta_j - \frac{\alpha}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

}

....(3.2)



```

C:\Octave\OCTAVE~1.0\bin\octave-
theta =

-3.06434694
-0.06847623
-0.41069215
 5.30730370
 2.74222685
 0.00063922

ans = 0.33543
octave:4>

```

FIG 3.5: Theta values

These values of theta are then used in our JavaScript code acting as weights to predict a probability percentage for each tier of universities.

```

104 a=(1*-3.064)+(v*-0.068)+(w*-0.41)+(x*5.307)+(y*2.742)+(z*0.0006);
105 b=(1*-355.44)+(v*0.77)+(w*8.95)+(x*3.86)+(y*2.89)+(z*0.00031);
106 c=(1*-358.90)+(v*0.468)+(w*12.093)+(x*9.54)+(y*2.042)+(z*0.00136);
107 d=(1*-1211.77)+(v*0.71)+(w*20.99)+(x*142.39)+(y*7.21)+(z*0.001);
108 a=Math.exp(a)/(Math.exp(a)+1);
109 b=Math.exp(b)/(Math.exp(b)+1);
110 c=Math.exp(c)/(Math.exp(c)+1);
111 d=Math.exp(d)/(Math.exp(d)+1);
112 if(a<0.0001)
113 {
114     a=0;
115 }
116 if(b<0.0001)
117 {
118     b=0;
119 }
120 if(c<0.0001)
121 {
122     c=0;
123 }
124 if(d<0.0001)
125 {
126     d=0;
127 }
128 a=a*100;
129 a=Math.round(a);
130 b=b*100;
131 b=Math.round(b);

```

FIG 3.6: Substituting Theta as Weights

INPUT:

The JavaScript form is filled with attributes belonging to our model by the user. The GRE score, CGPA, SOP rating, Experience, Budget and the Country of the student are inputted by the user.

SECUREUNI FIND YOUR UNIVERSITY TOP UNIVERSITIES

Enter GRE Score
335

Enter CGPA (1-10)
9

Enter SOP Rating (1-5)
4

Enter Experience in years (0-10)
2

Enter Budget in Dollars
40000

Enter Area of Interest (USA, UK, Canada, Germany)
USA

Submit

FIG 3.7: JavaScript Input form

OUTPUT:

Based on the input, the probability for each university is calculated using our Logistic Regression Model.

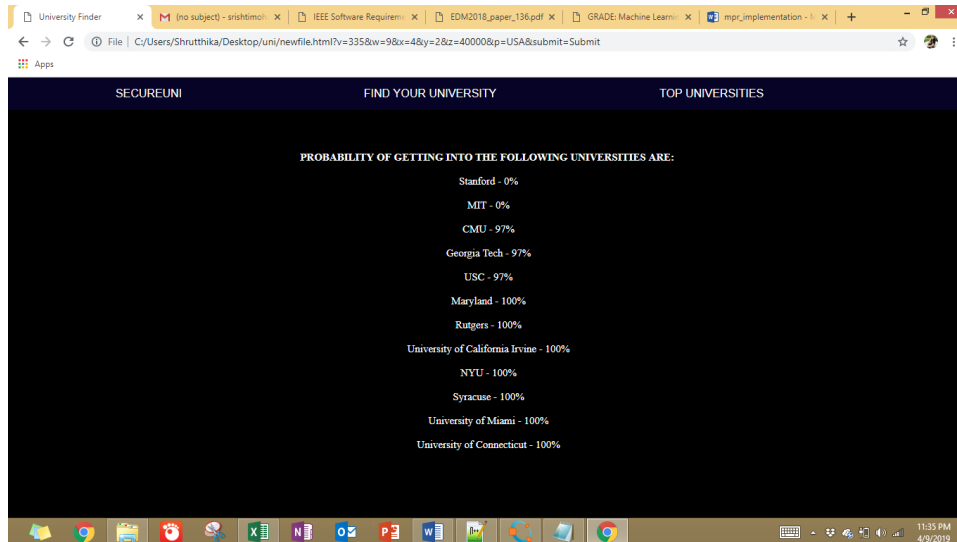


FIG 3.8: Prediction Results

Model Evaluation:

The model is evaluated for each tier and the accuracy is found out to be 92.59%

```
import matplotlib.pyplot as plt

In [3]: from sklearn.linear_model import LogisticRegression
        model = LogisticRegression()

In [4]: from sklearn.model_selection import train_test_split

In [6]: data1 = np.genfromtxt('XA.csv', delimiter=',')
        data2 = np.genfromtxt('YA.csv', delimiter=',')

In [27]: X_train, X_test, y_train, y_test = train_test_split(data1, data2, test_size=0.3)

In [28]: model.fit(X_train, y_train)

Out[28]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                             intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                             penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                             verbose=0, warm_start=False)

In [29]: print("Accuracy of model is :")
        print(model.score(X_test, y_test)*100)

Accuracy of model is :
92.5925925925926
```

FIG 3.9: Model Evaluation

CHAPTER 4

4.1. CONCLUSION

SecureUni is thus a platform which will help the students who are planning their further studies abroad to prepare themselves enough for the competitive world. Getting into any of the top universities all over the world is a very difficult task and SecureUni has been built just to make it much easier for the students to decide how well to prepare and maintain their good scores for admission into one of the top universities.

Apart from just telling you your chances of getting into any of the universities, SecureUni also provides you with the facility of all the past cut-off's and detailed information about each and every university, with their fee structures. SecureUni uses the logistic regression technique of machine learning to make these predictions, and will make the most accurate decision by it as much as possible.

SecureUni is a portal that shall be available at any given time and will be very accessible to students all over the world through its online portal. The cost of using this portal has been kept to a bare minimum too since we understand the students want to access this multiple times and check what they would want to keep as a threshold value for themselves while preparing. Thus, SecureUni with its user-friendly UI hopes to help a lot of students build their career and get the university of their choice. The portal too shall be updated with more and more data so as to make predictions more and more accurate for people in the forthcoming times.

4.2. FUTURE SCOPE

Predictions made by SecureUni can be made more accurate by adding new data as and when more people are admitted into these universities. Some other points that describe the further scope of this project are:

- Currently, SecureUni only looks at Universities offering MS. However, in future we aim at extending the reach for students who want to get into top B-Schools as well.
- Through constant updates the UI of the portal can be improved from time to time to make it more user-friendly.
- Including junior college admissions for various streams too is something that bothers Indian students the most and we shall do our best to try and include predictions to get into top junior colleges in India.

CHAPTER 5

REFERENCES

1. Bruggink, T. H., and Gambhir, V. 1996. Statistical models for college admission and enrollment: A case study for a selective liberal arts college. *Research in Higher Education* 37(2):221–240.
2. Deerwester, S.; Dumais, S.; Furnas, G.; Landauer, T.; and Harshman, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41(6):391–407.
3. Koh, K.; Kim, S.; and Boyd, S. 2007. An interior-point method for large-scale l_1 -regularized logistic regression. *Journal of Machine Learning Research* 8(8):1519–1555.
4. Moore, J. S. 1998. An expert system approach to graduate school admission decisions and academic performance prediction. *Omega* 26(5):659–670.
5. Saltelli, A.; Chan, K.; Scott, E.; et al. 2000. *Sensitivity Analysis*, volume 134. Wiley New York.
6. Tan, M.; Wang, L.; and Tsang, I. 2010. Learning sparse svm for feature selection on very high dimensional datasets. In *International Conference on Machine Learning*.
7. Van Rijsbergen, C.; Robertson, S.; and Porter, M. 1980. *New Models in Probabilistic Information Retrieval*. British Library Research and Development Department.