A MINI PROJECT REPORT

On

HR Case Study:
Matching HRs with right Interns

Submitted by

Name: Anujay Jain
Roll No: 161500106

Name: Nihit Jain
Roll No: 161500350

Name: Utkarsh Rai
Roll No: 161500599

Name: Shahaban Ali
Roll No:161500496

To
Mr. Rahul Pradhan

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University
Mathura- 281406, INDIA

December, 2018

Declaration

*We hereby declare that the work which is being presented in the Mini Project "Multiple Regression and Model Building**",** in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of our own work carried under the supervision of **Mr. Rahul Pradhan, Assistant Professor, GLA University, Mathura**.*

Anujay Jain

Sign:_________

Nihit Jain

Sign:_________

Utkarsh Rai

Sign:_________

Shahaban Ali

Sign:_________

Department of Computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,

Mathura – 281406

CERTIFICATE

*This is to certify that the project entitled **"HR Case Study: Matching HRs with right Interns"** carried out in Mini Project – I Lab is a bonafide work done by **Anujay Jain (161500106), Nihit Jain (161500350), Shahaban Ali(161500599) and Utkarsh Rai (161500599)**and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).*

Signature of Supervisor:

Name of Supervisor:

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to ***Dr. (Prof). Anand Singh Jalal,*** *Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.*

We owe special debt of gratitude to ***Mr. Rahul Pradhan,*** *Assistant Professor Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.*

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Anujay Jain
Nihit Jain
Utkarsh Rai
Shahaban Ali

Abstract

In this problem, we have been provided with the information about various internships posted on Internshala. This includes various attributes about the internships like location, duration, start_date of internship etc. We have also been provided with information about the students who have applied for the internship. These include type_of_institute, current_year, academic performance of the student etc. Any student is free to apply for any internship on the portal.

While employers get high response to their posting, it is difficult to go through a high number of applications for the employers. They might need to go through high number of applications to shortlist the most relevant candidates. Hence an intelligent matching algorithm can help our users get better experience and enhance chances of meaningful profile matches.

Table of Contents

Declaration ii

Certificate iii

Acknowledgments iv

Abstract v

Table of Contents vi

1. Chapter 1 **1**

1.1 Business Understanding……………………..1

1.2 Motivation ………………………………………………..............................1

1.3 Scope……………………………………………………………………….1

1.4 Drawbacks in existing system…………………………………………………...1

2. Chapter 2 **2**

2.1 General Description………………………………………………… ………..2

2.2 Project plan……….……………..….…………………………………...…2

2.2.1 Objective…………………………………………………………………...2

2.2.2 Goal………………………………………………………………………...2

3. Chapter 3 **3**

3.1 Project Implementation…………………………………………………………3

3.2 Understanding the dataset……………………………………………………....3

3.3 Describe Data……………………………………………………………………..5

3.4 Data cleaning…………………………………………………………………6

3.5 Model used…………………………………………………………………….7

3.6 Data insights…………………………………………………………………7

4. Chapter 4 **12**

4.1 Appendices……..……………………….………………………………………12

CHAPTER 1

1.1 Business Understanding

While employers get high response to their posting, it is difficult to go through a high number of applications for the employers. They might need to go through high number of applications to shortlist the most relevant candidates. Hence an intelligent matching algorithm can help our users get better experience and enhance chances of meaningful profile matches.

1.2 Motivation

The main motivation for us to go for this project was that a lot of internships are provided on internshala and a lot of applications for these internship are there, now if it the selection of the application is done manually it will consume lots of time and man-power and there are chances of human error. It is not possible to reduce the error of selection completely but if we can design a programme to select the most suitable candidate for a given internship then we ca reduce the time consumed and the workload and as it is being done by a machine the chances of error will also will be reduced and that is our goal.

1.3 Scope

The scope of our analysis is that we can reduce the workload for the company while selecting the most suitable interns for them among all the applicants. This is so, as based on our analysis we can find out those attributes which are most significant for the selection of the applicants thus helping us to focus on only those attributes rather than wasting our effort and time on other unrelated areas. Thus it will help as follows:

1. Reduce the workload.
2. Reduce the time consumed while selecting the suitable candidates.
3. It can also be used by the applicants to know which are the more suitable internship for them.

1.4 Drawbacks in existing system

- These days the selection for the internship is done manually which consumes more time.
- These days the number of domains for which you can apply is increasing and the criteria for applying for an intern as well as the specification which the applicants are searching for has increased a lot thus increasing the chance of error while selecting the most suitable applicants.

CHAPTER 2

2.1 General Description

- Data collected from Internshala.
- A data is recorded in a file - Internship.csv, Student.csv and train.csv

2.2 Project Plan

2.2.1 Objective

- Build the best multiple regression model that can predict the most suitable candidate for the various internship present on the internshala, using all the other variables as the predictors.
- Determine which variables must be made into indicator variables.
- Determine which variables might be superfluous.

2.2.2 Goals

- To reduce the total number of applicants a company has to go over during the selection for the internship.
- To reduce the chances of error of selection a less suitable applicant when a more suitable one is present.

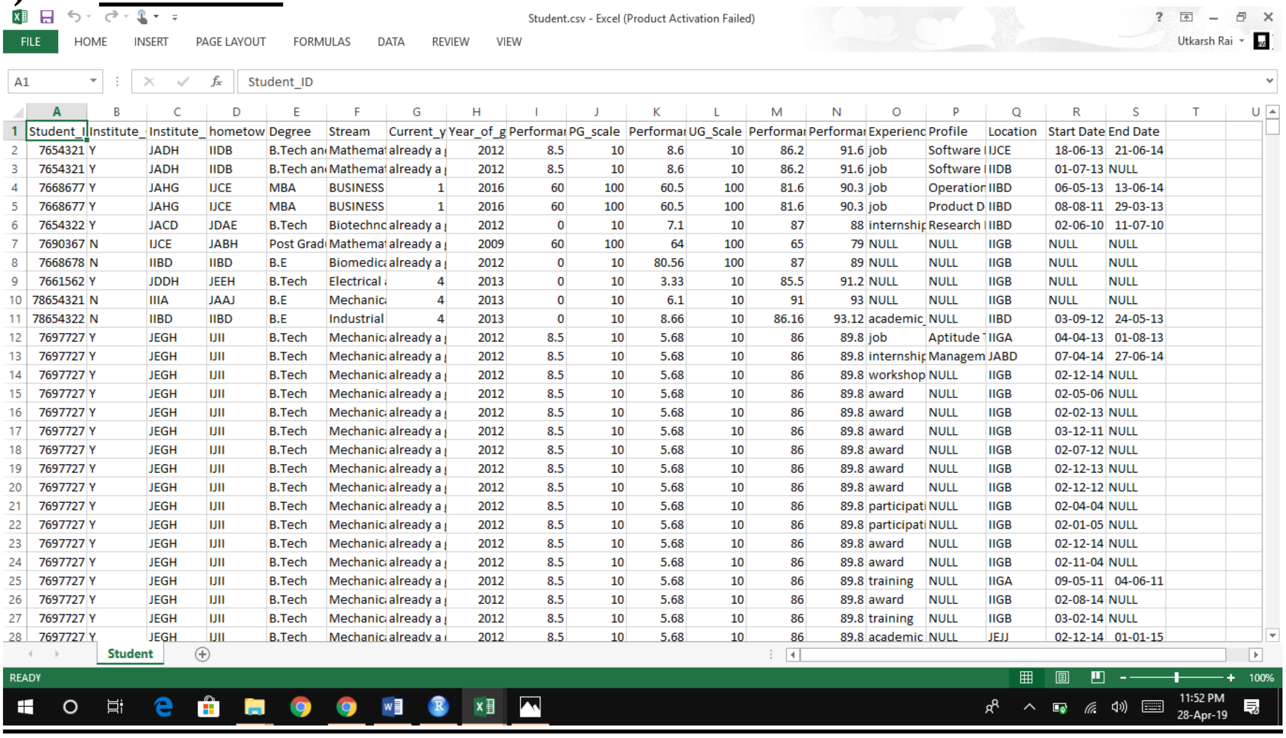
CHAPTER 3

3.1 Project Implementation

It includes the steps taken to implement the project.

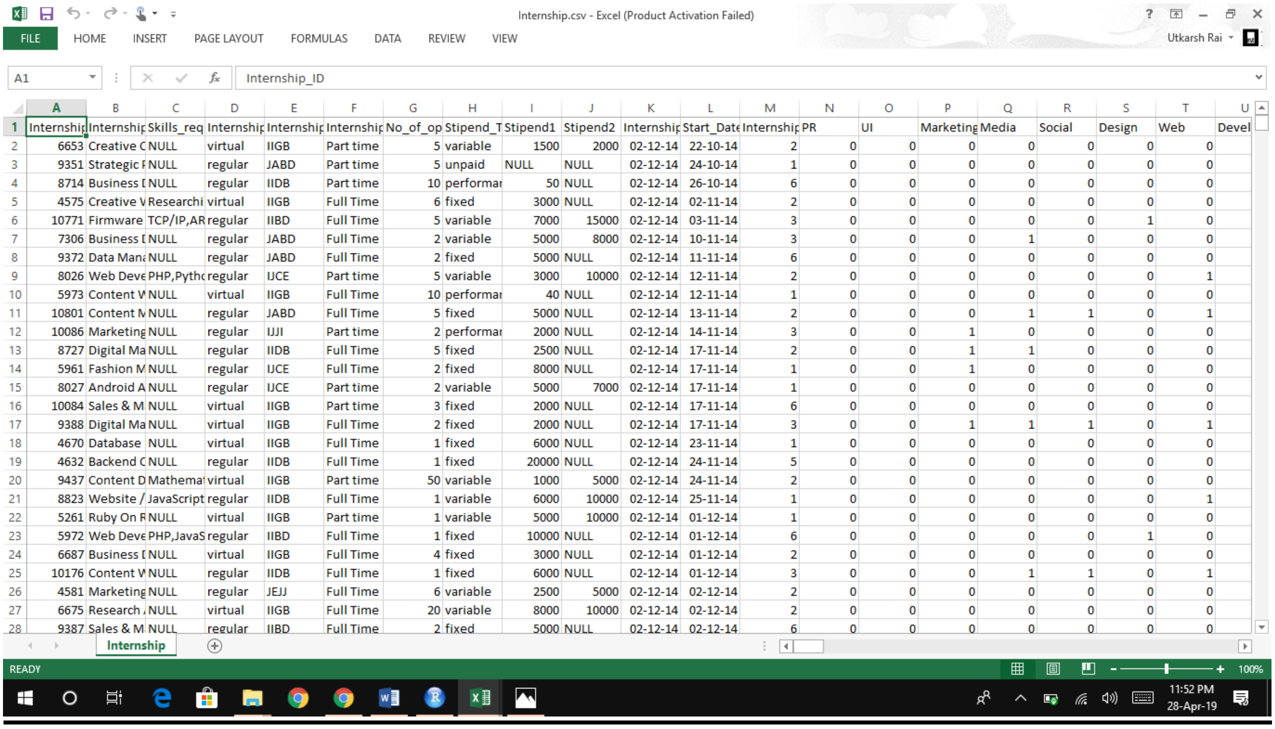
3.2 Understanding the Dataset

i) Student.csv



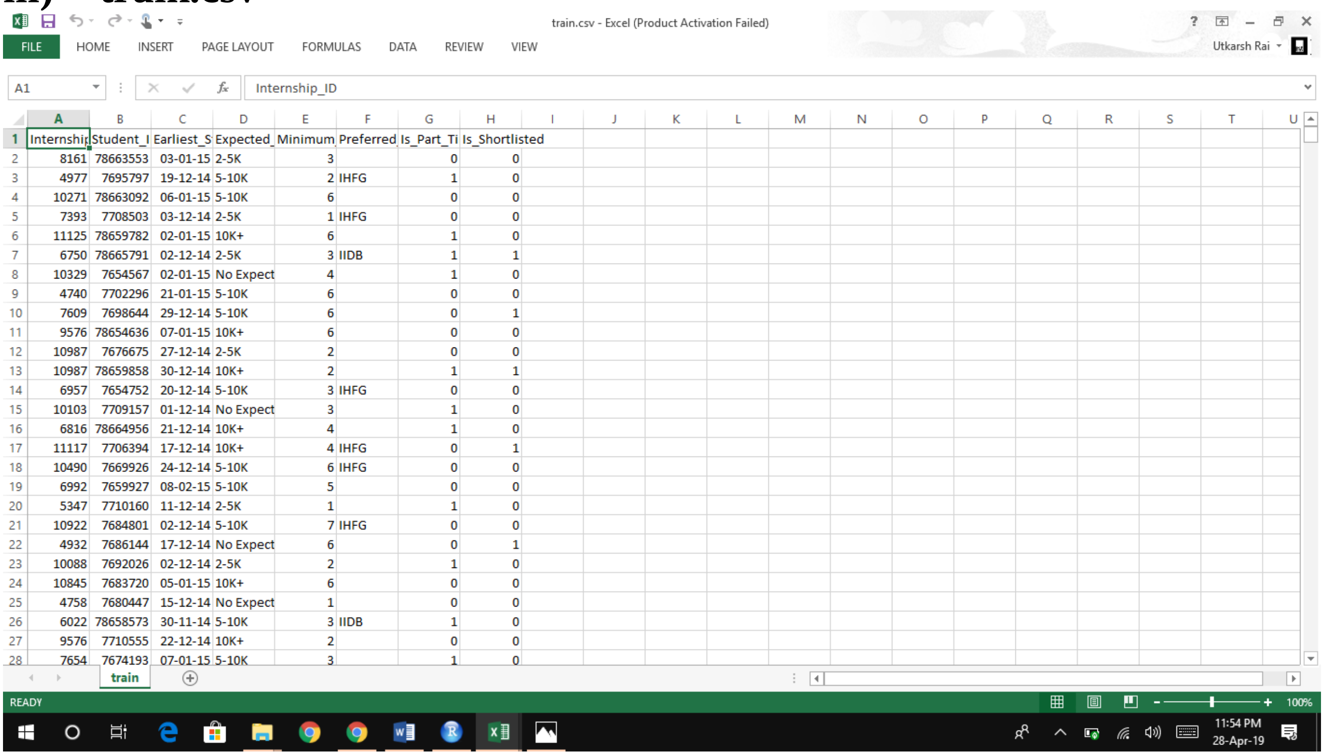
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Student_I	Institute_	Institute_	hometow	Degree	Stream	Current_y	Year_of_g	Performa	PG_scale	Performa	UG_Scale	Performa	Performa	Experienc	Profile	Location	Start Date	End Date
2	7654321	Y	JADH	IIDB	B.Tech an	Mathemat	already a	2012	8.5	10	8.6	10	86.2	91.6	job	Software I	IJCE	18-06-13	21-06-14
3	7654321	Y	JADH	IIDB	B.Tech an	Mathemat	already a	2012	8.5	10	8.6	10	86.2	91.6	job	Software I	IIDB	01-07-13	NULL
4	7668677	Y	JAHG	IJCE	MBA	BUSINESS	1	2016	60	100	60.5	100	81.6	90.3	job	Operation	IIBD	06-05-13	13-06-14
5	7668677	Y	JAHG	IJCE	MBA	BUSINESS	1	2016	60	100	60.5	100	81.6	90.3	job	Product D	IIBD	08-08-11	29-03-13
6	7654322	Y	JACD	JDAE	B.Tech	Biotechno	already a	2012	0	10	7.1	10	87	88	internship	Research I	IIBD	02-06-10	11-07-10
7	7690367	N	IJCE	JABH	Post Grad	Mathemat	already a	2009	60	100	64	100	65	79	NULL	NULL	IIGB	NULL	NULL
8	7668678	N	IIBD	IIBD	B.E	Biomedica	already a	2012	0	10	80.56	100	87	89	NULL	NULL	IIGB	NULL	NULL
9	7661562	Y	JDDH	JEEH	B.Tech	Electrical	4	2013	0	10	3.33	10	85.5	91.2	NULL	NULL	IIGB	NULL	NULL
10	78654321	N	IIIA	JAAJ	B.E	Mechanic	4	2013	0	10	6.1	10	91	93	NULL	NULL	IIGB	NULL	NULL
11	78654322	N	IIBD	IIBD	B.E	Industrial	4	2013	0	10	8.66	10	86.16	93.12	academic_	NULL	IIBD	03-09-12	24-05-13
12	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	job	Aptitude T	IIGA	04-04-13	01-08-13
13	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	internship	Managem	JABD	07-04-14	27-06-14
14	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	workshop	NULL	IIGB	02-12-14	NULL
15	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-05-06	NULL
16	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-02-13	NULL
17	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	03-12-11	NULL
18	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-07-12	NULL
19	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-12-13	NULL
20	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-12-12	NULL
21	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	participati	NULL	IIGB	02-04-04	NULL
22	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	participati	NULL	IIGB	02-01-05	NULL
23	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-12-14	NULL
24	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-11-04	NULL
25	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	training	NULL	IIGA	09-05-11	04-06-11
26	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	award	NULL	IIGB	02-08-14	NULL
27	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	training	NULL	IIGB	03-02-14	NULL
28	7697727	Y	JEGH	IJII	B.Tech	Mechanic	already a	2012	8.5	10	5.68	10	86	89.8	academic	NULL	JEJJ	02-12-14	01-01-15

ii) <u>Internship.csv</u>



	Internship	Internship	Skills_req	Internship	Internship	Internship	No_of_op	Stipend_T	Stipend1	Stipend2	Internship	Start_Date	Internship	PR	UI	Marketing	Media	Social	Design	Web	Devel
2	6653	Creative C	NULL	virtual	IIGB	Part time	5	variable	1500	2000	02-12-14	22-10-14	2	0	0	0	0	0	0	0	
3	9351	Strategic F	NULL	regular	JABD	Part time	5	unpaid	NULL	NULL	02-12-14	24-10-14	1	0	0	0	0	0	0	0	
4	8714	Business I	NULL	regular	IIDB	Part time	10	performar	50	NULL	02-12-14	26-10-14	6	0	0	0	0	0	0	0	
5	4575	Creative V	Researchi	virtual	IIGB	Full Time	6	fixed	3000	NULL	02-12-14	02-11-14	2	0	0	0	0	0	0	0	
6	10771	Firmware	TCP/IP,AR	regular	IIBD	Full Time	5	variable	7000	15000	02-12-14	03-11-14	3	0	0	0	0	0	1	0	
7	7306	Business I	NULL	regular	JABD	Full Time	2	variable	5000	8000	02-12-14	10-11-14	3	0	0	0	1	0	0	0	
8	9372	Data Mana	NULL	regular	JABD	Full Time	2	fixed	5000	NULL	02-12-14	11-11-14	6	0	0	0	0	0	0	0	
9	8026	Web Deve	PHP,Pytho	regular	IJCE	Part time	5	variable	3000	10000	02-12-14	12-11-14	2	0	0	0	0	0	0	1	
10	5973	Content W	NULL	virtual	IIGB	Full Time	10	performar	40	NULL	02-12-14	12-11-14	1	0	0	0	0	0	0	0	
11	10801	Content M	NULL	regular	JABD	Full Time	5	fixed	5000	NULL	02-12-14	13-11-14	2	0	0	0	1	1	0	1	
12	10086	Marketing	NULL	regular	IJJI	Part time	2	performar	2000	NULL	02-12-14	14-11-14	3	0	0	1	0	0	0	0	
13	8727	Digital Ma	NULL	regular	IIDB	Full Time	5	fixed	2500	NULL	02-12-14	17-11-14	2	0	0	1	1	0	0	0	
14	5961	Fashion M	NULL	regular	IJCE	Full Time	2	fixed	8000	NULL	02-12-14	17-11-14	1	0	0	1	0	0	0	0	
15	8027	Android A	NULL	regular	IJCE	Part time	2	variable	5000	7000	02-12-14	17-11-14	1	0	0	0	0	0	0	0	
16	10084	Sales & M	NULL	virtual	IIGB	Part time	3	fixed	2000	NULL	02-12-14	17-11-14	6	0	0	0	0	0	0	0	
17	9388	Digital Ma	NULL	virtual	IIGB	Full Time	2	fixed	2000	NULL	02-12-14	17-11-14	3	0	0	1	1	1	0	1	
18	4670	Database	NULL	virtual	IIGB	Full Time	1	fixed	6000	NULL	02-12-14	23-11-14	1	0	0	0	0	0	0	0	
19	4632	Backend C	NULL	regular	IIDB	Full Time	1	fixed	20000	NULL	02-12-14	24-11-14	5	0	0	0	0	0	0	0	
20	9437	Content D	Mathemat	virtual	IIGB	Part time	50	variable	1000	5000	02-12-14	24-11-14	2	0	0	0	0	0	0	0	
21	8823	Website /	JavaScript	regular	IIDB	Full Time	1	variable	6000	10000	02-12-14	25-11-14	1	0	0	0	0	0	0	1	
22	5261	Ruby On R	NULL	virtual	IIGB	Part time	1	variable	5000	10000	02-12-14	01-12-14	1	0	0	0	0	0	0	0	
23	5972	Web Deve	PHP,JavaS	regular	IIBD	Full Time	1	fixed	10000	NULL	02-12-14	01-12-14	6	0	0	0	0	0	1	0	
24	6687	Business I	NULL	virtual	IIGB	Full Time	4	fixed	3000	NULL	02-12-14	01-12-14	2	0	0	0	0	0	0	0	
25	10176	Content W	NULL	regular	IIDB	Full Time	1	fixed	6000	NULL	02-12-14	01-12-14	3	0	0	0	1	1	0	1	
26	4581	Marketing	NULL	regular	JEJJ	Full Time	6	variable	2500	5000	02-12-14	02-12-14	2	0	0	0	0	0	0	0	
27	6675	Research	NULL	virtual	IIGB	Full Time	20	variable	8000	10000	02-12-14	02-12-14	2	0	0	0	0	0	0	0	
28	9387	Sales & M	NULL	regular	IIBD	Full Time	2	fixed	5000	NULL	02-12-14	02-12-14	6	0	0	0	0	0	0	0	

iii) train.csv



	Internship	Student_I	Earliest_S	Expected_	Minimum	Preferred	Is_Part_Ti	Is_Shortlisted
2	8161	78663553	03-01-15	2-5K	3		0	0
3	4977	7695797	19-12-14	5-10K	2	IHFG	1	0
4	10271	78663092	06-01-15	5-10K	6		0	0
5	7393	7708503	03-12-14	2-5K	1	IHFG	0	0
6	11125	78659782	02-01-15	10K+	6		1	0
7	6750	78665791	02-12-14	2-5K	3	IIDB	1	1
8	10329	7654567	02-01-15	No Expect	4		1	0
9	4740	7702296	21-01-15	5-10K	6		0	0
10	7609	7698644	29-12-14	5-10K	6		0	1
11	9576	78654636	07-01-15	10K+	6		0	0
12	10987	7676675	27-12-14	2-5K	2		0	0
13	10987	78659858	30-12-14	10K+	2		1	1
14	6957	7654752	20-12-14	5-10K	3	IHFG	0	0
15	10103	7709157	01-12-14	No Expect	3		1	0
16	6816	78664956	21-12-14	10K+	4		1	0
17	11117	7706394	17-12-14	10K+	4	IHFG	0	1
18	10490	7669926	24-12-14	5-10K	6	IHFG	0	0
19	6992	7659927	08-02-15	5-10K	5		0	0
20	5347	7710160	11-12-14	2-5K	1		1	0
21	10922	7684801	02-12-14	5-10K	7	IHFG	0	0
22	4932	7686144	17-12-14	No Expect	6		0	1
23	10088	7692026	02-12-14	2-5K	2		1	0
24	10845	7683720	05-01-15	10K+	6		0	0
25	4758	7680447	15-12-14	No Expect	1		0	0
26	6022	78658573	30-11-14	5-10K	3	IIDB	1	0
27	9576	7710555	22-12-14	10K+	2		0	0
28	7654	7674193	07-01-15	5-10K	3		1	0

3.3 Describe data

The description of the column is as follows:

i) Student.csv → It has 19 columns and 151191 rows.

Variable_Name	Definition
Student_ID	Student_ID
Institute_Category	Tier1 (Y)/ Not (N)
Institute_location	Location_code
hometown	Location_code
Degree	Degree
Stream	Stream of education
Current_year	Current In which year of UG and PG
Year_of_graduation	Year of graduation
Performance_PG	Score of PG
PG_scale	Scale (could be 4, 10, 100)
Performance_UG	Score of UG
UG_Scale	Scale (could be 4, 10, 100)
Performance_12th	Performance in 12th (10 + 2)
Performance_10th	Performance in 10th
Experience_Type	Type of past experience(Job, Internship, Award, Academic Projects .....)
Profile	Profile in past experience
Location	Location of work experience
Start Date	Start Date of Work experience
End Date	End Date of Work experience

ii) Internship.csv → It has 286 columns and 6899 rows.

Variable_Name	Definition
Internship_ID	Internship_ID
Internship_Profile	Profile of the internship posted (as per company)
Skills_required	Required skills for internship (as per company)
Internship_Type	Type of Internship (Regular/ Virtual)
Internship_Location	Location code
Internship_category	Category of Internship (Parttime/ Full Time)
No_of_openings	Total number of open internships
Stipend_Type	Type of Stipend(Fixed, Variable, Unpaid, Performance)
Stipend1	Minimum Stipend (as per company)
Stipend2	Maximum Stipend (as per company)
Internship_deadline	Internship_Deadline_Date for application
Start_Date	Internship_Start_Date
Internship_Duration(Months)	Duration of Internship
Column14-Column286	Sparse matrix of skills (derived from Internship Responsibilities)

iii) train.csv → It has 8 columns and 192582 rows.

Variable_Name	Definition
Internship_ID	Internship_ID; Each internship has a unique id numb
Student_ID	Student_ID - unique for each student
Earliest_Start_Date	Earliest date student can start their Internship
Expected_Stipend	Expected stipend by student
Minimum_Duration	Months students is available for Internship
Preferred_location	Preferred location code
Is_Part_Time	Available for Part_time(1)/ Full_Time(0)
Is_Shortlisted	Target Variable (1: Shortlisted, 0: Not Shortlisted)

3.4 Data cleaning

The data which is obtained may need to be processed before it can be actually used, like there may be some values missing which need to be filled otherwise they will cause problem when doing the analyses on the data.

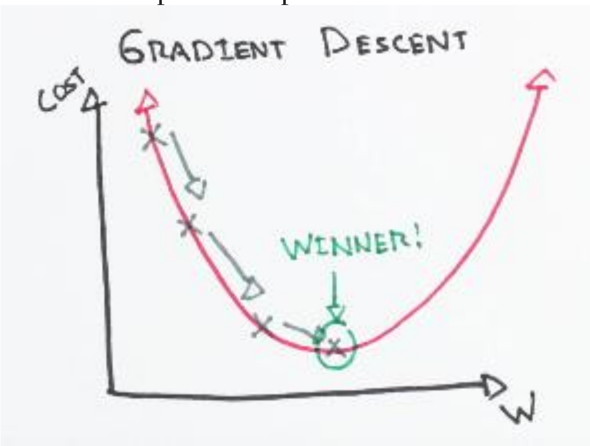
	A	B	C	D	E	F	G	H	I	J
1	Internship	Internship	Skills_req	Internship	Internship	Internship	No_of_op	Stipend_T	Stipend1	Stipend2
2	6653	Creative C	NULL	virtual	IIGB	Part time	5	variable	1500	2000
3	9351	Strategic F	NULL	regular	JABD	Part time	5	unpaid	NULL	NULL
4	8714	Business D	NULL	regular	IIDB	Part time	10	performar	50	NULL
5	4575	Creative V	Researchi	virtual	IIGB	Full Time	6	fixed	3000	NULL
6	10771	Firmware	TCP/IP,AR	regular	IIBD	Full Time	5	variable	7000	15000
7	7306	Business D	NULL	regular	JABD	Full Time	2	variable	5000	8000
8	9372	Data Mana	NULL	regular	JABD	Full Time	2	fixed	5000	NULL
9	8026	Web Deve	PHP,Pytho	regular	IJCE	Part time	5	variable	3000	10000
10	5973	Content W	NULL	virtual	IIGB	Full Time	10	performar	40	NULL
11	10801	Content M	NULL	regular	JABD	Full Time	5	fixed	5000	NULL
12	10086	Marketing	NULL	regular	IJJI	Part time	2	performar	2000	NULL
13	8727	Digital Ma	NULL	regular	IIDB	Full Time	5	fixed	2500	NULL
14	5961	Fashion M	NULL	regular	IJCE	Full Time	2	fixed	8000	NULL
15	8027	Android A	NULL	regular	IJCE	Part time	2	variable	5000	7000
16	10084	Sales & M	NULL	virtual	IIGB	Part time	3	fixed	2000	NULL
17	9388	Digital Ma	NULL	virtual	IIGB	Full Time	2	fixed	2000	NULL

In the above image we can see that there are some null values so we need to fill them before we do anything else.

3.5 Model used

Since our problem is a classification problem and has a very large size of more than 100k samples , we decided to use the technique of Gradient descent for which the most suitable algorithm is Gradient boosting which is able to solve the problem of large size , handling data of mixed type and missing values, robust to outliers in input space and has a good interpretability and predictive power.

Gradient descent is an optimization algorithm used to minimize some function by iteratively moving in the direction of steepest descent as defined by the negative of the gradient. In machine learning, we use gradient descent to update the parameters of our model.



Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

3.6 Data Insights

Here are certain insights of the data which may help in analyses

1. A general view of the datasets.

   train.csv

In [11]:
```
# View Training Data
df_train.head(2)
```

Out[11]:

	Internship_ID	Student_ID	Earliest_Start_Date	Expected_Stipend	Minimum_Duration	Preferred_location	Is_Part_Time	Is_Shortlisted
0	8161	78663553	03-01-2015	2-5K	3	NaN	0	0
1	4977	7695797	19-12-2014	5-10K	2	IHFG	1	0

Internship.csv

In [9]:
```
# View Internships
df_internship.head(2)
```

Out[9]:

	Internship_ID	Internship_Profile	Skills_required	Internship_Type	Internship_Location	Internship_category	No_of_openings	Stipend_Type	Stipend1	Stipend
0	6653	Creative Content Writing	NaN	virtual	IIGB	Part time	5	variable	1500.0	2000.
1	9351	Strategic Philanthropy	NaN	regular	JABD	Part time	5	unpaid	NaN	Nal

2 rows × 286 columns

Student.csv

In [10]:
```
# VIew Students Applicants
df_student.head(2)
```

Out[10]:

	Student_ID	Institute_Category	Institute_location	hometown	Degree	Stream	Current_year	Year_of_graduation	Performance_PG	PG_scale	Performan
0	7654321	Y	JADH	IIDB	B.Tech and M.Tech (Dual Degree)	Mathematics & Computing	already a graduate	2012	8.5	10	
1	7654321	Y	JADH	IIDB	B.Tech and M.Tech (Dual Degree)	Mathematics & Computing	already a graduate	2012	8.5	10	

2. Summary of the datasets before preprocessing

```
df_internship.iloc[:,:13].describe()
```

	Internship_ID	No_of_openings	Stipend1	Stipend2	Internship_Duration(Months)
count	6899.000000	6899.000000	6771.000000	3151.000000	6.899000e+03
mean	8016.000000	4.447601	5673.532270	10518.329102	5.849465e+03
std	1991.714086	6.395352	4318.323717	7407.088517	3.432311e+05
min	4567.000000	1.000000	1.000000	100.000000	0.000000e+00
25%	6291.500000	2.000000	3000.000000	5000.000000	2.000000e+00
50%	8016.000000	2.000000	5000.000000	10000.000000	3.000000e+00
75%	9740.500000	5.000000	8000.000000	15000.000000	4.000000e+00
max	11465.000000	100.000000	50000.000000	150000.000000	2.016033e+07

```
df_student.describe()
```

	Student_ID	Year_of_graduation	Performance_PG	PG_scale	Performance_UG	UG_Scale	Performanc
count	1.511910e+05	151191.000000	151191.000000	151191.000000	151191.000000	151191.000000	151191.000(
mean	2.173736e+07	2015.225152	4.560760	25.360002	32.506286	47.842054	77.652632
std	2.828474e+07	1.434272	16.150917	34.032013	30.976177	44.642229	14.597772
min	7.654321e+06	2001.000000	0.000000	4.000000	0.000000	4.000000	0.000000
25%	7.671942e+06	2015.000000	0.000000	10.000000	7.200000	10.000000	69.000000
50%	7.690571e+06	2015.000000	0.000000	10.000000	8.660000	10.000000	80.000000
75%	7.708364e+06	2016.000000	0.000000	10.000000	66.000000	100.000000	89.000000
max	7.866876e+07	2020.000000	100.000000	100.000000	100.000000	100.000000	100.000000

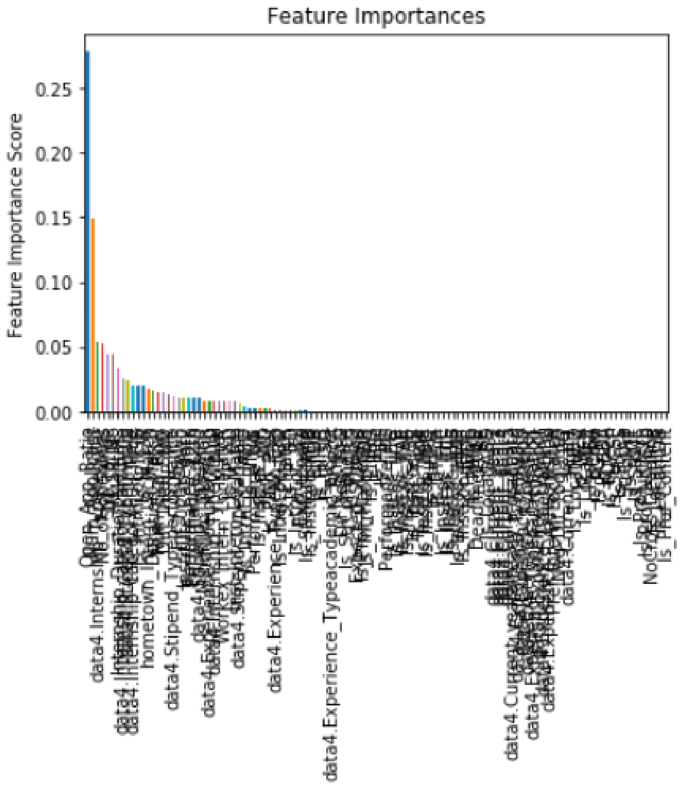
```
df_train.describe()
```

	Internship_ID	Student_ID	Minimum_Duration	Is_Part_Time	Is_Shortlisted
count	192582.000000	1.925820e+05	192582.000000	192582.000000	192582.000000
mean	7910.562919	2.161640e+07	3.790043	0.343012	0.127629
std	2006.863160	2.819268e+07	2.702877	0.474717	0.333677
min	4568.000000	7.654321e+06	1.000000	0.000000	0.000000
25%	6111.000000	7.672068e+06	2.000000	0.000000	0.000000
50%	8072.000000	7.690870e+06	3.000000	0.000000	0.000000
75%	9649.000000	7.708160e+06	6.000000	1.000000	0.000000
max	11334.000000	7.866825e+07	12.000000	1.000000	1.000000

3. Model fitting

```
]: modelfit(gbm0, train, test, predictors)
```

```
Model Report
Accuracy : 0.8758
AUC Score (Train): 0.750221
CV Score : Mean - 0.6623082 | Std - 0.01868213 | Min - 0.6375912 | Max - 0.6816427
```




```
In [30]: #  With all tuned lets try reducing the learning rate and proportionally increasing the number of estimators to get
         #  more robust results:

         predictors = predictors1
         gbm_tuned_1 = GradientBoostingClassifier(learning_rate=0.05, n_estimators=120,max_depth=4, min_samples_split=150,min_samples_lea
         modelfit(gbm_tuned_1, train, test, predictors)
```

```
Model Report
Accuracy : 0.8734
AUC Score (Train): 0.739402
CV Score : Mean - 0.6790757 | Std - 0.01970682 | Min - 0.6599959 | Max - 0.7168823
```

4. Improvement in the model

```
In [20]: #gsearch1.grid_scores_
         gsearch1.cv_results_
         gsearch1.best_params_
         gsearch1.best_score_

Out[20]: 0.6750645216043039
```

```
In [24]: #gsearch3.grid_scores_
         gsearch3.cv_results_
         gsearch3.best_estimator_
         gsearch3.best_score_

Out[24]: 0.6830675608165294
```

Chapter 7

Appendices

Code:

Data Cleaning

```
interns <- read.csv("trainfiles/Internship/Internship.csv")
student   <- read.csv("trainfiles/Student/Student.csv")
train <- read.csv("trainfiles/traincsv/train.csv")
test  <- read.csv("test-date-your-data/test.csv")

#install.packages("sqldf")

library(sqldf)
student1 <- student
student1$S_Date <- student1$Start.Date
student1$E_Date <- student1$End.Date
student1$Num_Exp <- 1
student2 <- sqldf("select Student_ID, Institute_Category, Institute_location ,hometown ,Degree,
        Stream, Current_year, Year_of_graduation, Performance_PG, PG_scale,
        Performance_UG, UG_Scale, Performance_12th, Performance_10th, Experience_Type,
        Profile, Location, S_Date, E_Date, SUM(Num_Exp) as Num_Exp_Row From student1 Group BY Student_ID")

# Converting S_Date, E_Date to date class
S_Date <- as.Date(student2$S_Date, "%d-%m-%Y")
E_Date <- as.Date(student2$E_Date, "%d-%m-%Y")


student2$S_Date <- S_Date
student2$E_Date <- E_Date


# tagging train and test data
train1 <- train
train1$tag <- "train"
test1 <- test
test1$tag <- "test"

#Combining train and test
test1$Is_Shortlisted <- 0
```

```
data <- rbind(train1,test1)

#combining data and student2

data1 <- merge(data,student2,by="Student_ID",all.x=TRUE)
interns1 <- interns[,c(1:13)]
data2 <- merge(data1,interns1, by="Internship_ID", all.x=TRUE)

## modification of Earliest_Start_Date

ESD <- data2$Earliest_Start_Date
ESD1 <- gsub('/','-',ESD)
ESD2 <- as.Date(ESD1, "%d-%m-%Y")
data2$Earliest_Start_Date <- ESD2

## Converting "Start_Date"  to Date class
Start_Date <- data2$Start_Date
Start_Date <- as.Date(Start_Date,"%d-%m-%Y")
data2$Start_Date <- Start_Date

## Class balance
table(train$Is_Shortlisted)
#    0      1
#168003  24579

## Converting to factor variables Degree ,Stream , Profile
data2$Degree <- as.factor(data2$Degree)
data2$Stream <- as.factor(data2$Stream)
data2$Profile <-as.factor(data2$Profile)
data3 <- data2

# missing value treatment of data3$Preferred_location
# Lets tag it as No_Pref and create a feature to tag it
data3$Preferred_location <- as.character(data3$Preferred_location)
data3$Preferred_location <- ifelse(data3$Preferred_location=="","No_Pref",data3$Preferred_location)
data3$Preferred_location <- as.factor(data3$Preferred_location)


# substituting NA values of Degree with most common category
data3$Degree <- as.character(data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Stream=="Management", "MBA",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Stream=="Fashion Lifestyle Business Management", "MBA",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Stream=="Commence", "B.Com",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Stream=="Commerce", "B.Com",data3$Degree)
```

```
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Design", "Designing",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Social Media Marketing", "Digital Marketing",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Graphic Design", "Graphic Design",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Digital Marketing", "Digital Marketing",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Illustration", "B.A.(Hons) Journalism",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Google Ad Word Management", "MBA",data3$Degree)
data3$Degree <- ifelse(is.na(data3$Degree) & data3$Internship_Profile=="Operations- Quality Analyst", "Global Business Operations (GBO)",data3$Degree)

data3$Degree <- as.factor(data3$Degree)

# substituting NA values of Stream
data3$Stream <- as.character(data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Designing", "Accessory Designing",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="MCA", "Computer Application",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Post Graduate Dimploma in Management", "Marketing",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="MBA", "Marketing",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.Com (Hons.)", "Accountancy And Finance",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Graphic Design", "Visual Comm",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Bachelor of Business Admininstration", "Management",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Digital Marketing", "Commerce",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.M.M.", "Arts",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="BCA", "Computer Application",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Global Business Operations (GBO)", "Finance",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.A.LL.B. (Hons.)", "Law",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Under", "Under",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.A. Programme", "Arts",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.Sc (Hons.) Computer Science", "Science",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.S. & M.S. (Dual)", "Mathematics and Computing",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="Undecided", "Undecided",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Degree=="B.A.(Hons) Journalism", "Arts",data3$Stream)
data3$Stream <- ifelse(is.na(data3$Stream) & data3$Internship_Profile=="Editorial(Law)", "Law",data3$Stream)
data3$Stream <- as.factor(data3$Stream)


# Replacing NULL in Experience_Type , Profile with No_Exp
summary(data3$Experience_Type)
summary(data3$Profile)
data3$Profile <- as.character(data3$Profile)
data3$Experience_Type <- as.character(data3$Experience_Type)


table(as.factor(data3$Experience_Type))
data3$Profile[data3$Experience_Type!="NULL" & data3$Profile=="NULL"]<- "Intern"
data3$Profile[is.na(data3$Profile)] <- "Intern"
```

```
data3$Experience_Type[data3$Experience_Type=="NULL"] <- "No_Exp"
data3$Profile[data3$Profile=="NULL"] <- "No_Exp"

table(data3$Experience_Type)
sort(table(as.factor(data3$Profile)),decreasing=TRUE)[1:50]

data3$Profile <- as.factor(data3$Profile)
data3$Experience_Type <- as.factor(data3$Experience_Type)

# NAs in S_Date , E_Date

data3$S_Date <- as.character(data3$S_Date)
data3$E_Date <- as.character(data3$E_Date)

data3$S_Date[is.na(data3$S_Date) & data3$Experience_Type=="No_Exp"] <- "2015-02-21"
data3$E_Date[is.na(data3$E_Date) & data3$Experience_Type=="No_Exp"] <- "2015-02-21"
data3$S_Date <- as.Date(data3$S_Date,"%Y-%m-%d")
data3$E_Date <- as.Date(data3$E_Date,"%Y-%m-%d")

data3$E_Date[is.na(data3$E_Date)] <- as.Date("21-02-2015", "%d-%m-%Y")
max(data3$E_Date)



#NULL values of Stipend1 (2859 NULL values)
data3$Stipend1 <- as.character(data3$Stipend1)
data3$Stipend1 <- as.numeric(data3$Stipend1)
sum(is.na(data3$Stipend1))
sum(is.na(data3$Stipend1[data3$Stipend_Type=="unpaid"]))

## Stipend_Type == "unpaid" are NA or NULL in Stipend1; can replace them as 0
data3$Stipend1 <- ifelse(is.na(data3$Stipend1),0,data3$Stipend1)
table(data3$Stipend1[data3$Stipend_Type=="unpaid"])
# (7+5) obs in data3$Stipend1 has values otherthan 0 when Stipend_Type=="unpaid"
# Converting them to 0
#data3$Stipend1 <- ifelse(data3$Stipend_Type=="unpaid",0,data3$Stipend1)
data3$Stipend1 <- as.numeric(as.character(data3$Stipend1))
data3$Stipend1[data3$Stipend_Type=="unpaid"] <- 0

#NULL values of stipend2 (151897 NULL values) replaced  by median
data3$Stipend2 <- as.numeric(as.character(data3$Stipend2))
data3$Stipend2[data3$Stipend_Type=="unpaid"] <- 0

## NA values replaced by median
```

```
data3$Stipend2[is.na(data3$Stipend2)] <- 10000


## Capping outliersin data3$Stipend1
table(data3$Stipend1)
data3$Stipend2[data3$Stipend1==30000]

data3$Stipend1[data3$Stipend1==50000] <- 5000
data3$Stipend1[data3$Stipend1==40000] <- 4000
data3$Stipend1[data3$Stipend1==35000] <- 3500
data3$Stipend1[data3$Stipend1==30000 & data3$Stipend2==10000]<- 3000


## Capping outliersin data3$Stipend2
sort(data3$Stipend2,decreasing=TRUE)
table(data3$Stipend2)
data3$Stipend1[data3$Stipend2==150000] ##showing 8000 and 10000 . Must be wrong entry
data3$Stipend2[data3$Stipend2==150000]<- 15000

data3$Stipend1[data3$Stipend2==75000]
data3$Stipend1[data3$Stipend2==50000]


# Outliers in data2$Internship_Duration.Months.
summary(data3$Internship_Duration.Months.)
table(data3$Internship_Duration.Months.)
table(data3$Start_Date[data3$Internship_Duration.Months.==2016]) # 2014-12-15
#replacing by 24
data3$Internship_Duration.Months.<- ifelse(data3$Internship_Duration.Months.==2016,24,data3$Internship_Duration.Months.)

table(data3$Start_Date[data3$Internship_Duration.Months.==10000])
data3$Internship_Duration.Months.<- ifelse(data3$Internship_Duration.Months.==10000,10,data3$Internship_Duration.Months.)

table(data3$Start_Date[data3$Internship_Duration.Months.==20160201])
data3$Internship_Duration.Months.<- ifelse(data3$Internship_Duration.Months.==20160201,12,data3$Internship_Duration.Months.)

table(data3$Start_Date[data3$Internship_Duration.Months.==20160331])
data3$Internship_Duration.Months.<- ifelse(data3$Internship_Duration.Months.==20160331,15,data3$Internship_Duration.Months.)


# why min=0 in summary(data3$Performance_PG),summary(data3$Performance_UG),summary(data3$Performance_12th)
# summary(data3$Performance_10th)
table(data3$Performance_10th)

Performance_10th <- ifelse(data3$Performance_10th <= 10, (data3$Performance_10th*10),data3$Performance_10th)
```

```
Performance_10th <- ifelse(Performance_10th == 8.5, (Performance_10th*10),Performance_10th)
Performance_10th <- ifelse(Performance_10th < 40 , 40,Performance_10th)
data3$Performance_10th <- Performance_10th




table(data3$Performance_12th)
Performance_12th <- ifelse(data3$Performance_12th <= 10, (data3$Performance_12th*10),data3$Performance_12th)
Performance_12th <- ifelse(Performance_12th <= 10, (Performance_12th*10),Performance_12th)
Performance_12th <- ifelse(Performance_12th < 40 , 40,Performance_12th)
data3$Performance_12th <- Performance_12th

## Since UG_Scale is there, lets convert to ratio. Degree awarded student must have passed UG
table(data3$Performance_UG)
table(data3$UG_Scale[data3$Performance_UG==0.6])

Per_UG <- (data3$Performance_UG/data3$UG_Scale)*100
Per_UG <- ifelse(Per_UG <= 10, (Per_UG*10), Per_UG)
Per_UG[substr(data3$Degree,1,1)=="B" & Per_UG < 40 | substr(data3$Degree,1,1)=="M" & Per_UG < 40] <- 40

data3$Performance_UG <- Per_UG

## Since PG_Scale is there, lets convert to ratio
table(data3$Performance_PG)
Per_PG <- (data3$Performance_PG/data3$PG_scale)*100
Per_PG <- ifelse(Per_PG < 10, Per_PG*10, Per_PG)     # Per_PG=0 may be who are not PG yet
table(Per_PG)

Per_PG[substr(data3$Degree,1,1)=="M" & Per_PG < 40]<- 40
data3$Performance_PG <- Per_PG





# Skills_required NULL
data3$Skills_required <- as.character(data3$Skills_required)
data3$Skills_required[data3$Skills_required=="NULL"] <-"No_Skill"
data3$Skills_required <- as.factor(data3$Skills_required)

#Feature Engineering
# Exp_tenure

data3$Exp_tenure <- 0
data3$Exp_tenure <- data3$E_Date - data3$S_Date
data3$Exp_tenure <- as.numeric(as.character(data3$Exp_tenure))
```

```
summary(data3$Exp_tenure)

data3$Exp_tenure[data3$Exp_tenure < 0]<- 0
table(data3$Exp_tenure)

data3$S_Date[data3$Exp_tenure==1][1:10]
data3$E_Date[data3$Exp_tenure==1][1:10]
data3$Exp_tenure[data3$Exp_tenure < 30] <- 0


####################
## Tagging on Preferred_location

data4 <- data3
sort(table(data3$Preferred_location),decreasing=TRUE)
data4$Preferred_location <- as.character(data4$Preferred_location)
data4$Is_PlNo_Pref <- ifelse(data4$Preferred_location=="No_Pref",1,0)
data4$Is_PlIHFG    <- ifelse(data4$Preferred_location=="IHFG",1,0)
data4$Is_PlIHJB <- ifelse(data4$Preferred_location=="IHJB",1,0)
data4$Is_PlIIBD <- ifelse(data4$Preferred_location=="IIBD",1,0)
data4$Is_PlIIDB <- ifelse(data4$Preferred_location=="IIDB",1,0)
data4$Is_PlIJBG <- ifelse(data4$Preferred_location=="IJBG",1,0)
data4$Is_PlIJCE <- ifelse(data4$Preferred_location=="IJCE",1,0)
data4$Is_PlIJJI <- ifelse(data4$Preferred_location=="IJJI",1,0)
data4$Is_PlJABD <- ifelse(data4$Preferred_location=="JABD",1,0)
data4$Is_PlJBDB <- ifelse(data4$Preferred_location=="JBDB",1,0)

## Institute_location
sort(table(data4$Institute_location),decreasing=TRUE)
data4$Institute_location  <- as.character(data4$Institute_location)
data4$Is_InstLoc_IHHF <- ifelse(data4$Institute_location=="IHHF",1,0)
data4$Is_InstLoc_IHHH <- ifelse(data4$Institute_location=="IHHH",1,0)
data4$Is_InstLoc_IHJB <- ifelse(data4$Institute_location=="IHJB",1,0)
data4$Is_InstLoc_IJCE <- ifelse(data4$Institute_location=="IJCE",1,0)
data4$Is_InstLoc_IHJC <- ifelse(data4$Institute_location=="IHJC",1,0)
data4$Is_InstLoc_IIBD <- ifelse(data4$Institute_location=="IIBD",1,0)
data4$Is_InstLoc_IIDB <- ifelse(data4$Institute_location=="IIDB",1,0)
data4$Is_InstLoc_IIGE <- ifelse(data4$Institute_location=="IIGE",1,0)
data4$Is_InstLoc_IIIF <- ifelse(data4$Institute_location=="IIIF",1,0)
data4$Is_InstLoc_IIJJ <- ifelse(data4$Institute_location=="IIJJ",1,0)
data4$Is_InstLoc_IJAB <- ifelse(data4$Institute_location=="IJAB",1,0)
data4$Is_InstLoc_IJAE <- ifelse(data4$Institute_location=="IJAE",1,0)
data4$Is_InstLoc_IJGB <- ifelse(data4$Institute_location=="IJGB",1,0)
data4$Is_InstLoc_IJBG <- ifelse(data4$Institute_location=="IJBG",1,0)
```

```
## hometown

sort(table(data4$hometown),decreasing=TRUE)

data4$hometown  <- as.character(data4$hometown)
data4$Inf_hometown <- ifelse(data4$hometown %in%
c("IHGI","IHHH","IHJB","IHJC","IIAI","IIBD","IIDB","IIGA","IIIF","IJAB","IJAE","IJBG","IJCE","IJHA","IJIG",
        "IJJI","JAAJ","JABD","JADD","JADH","JAGD","JAHG","JBBE","JBDB","JBEB","JBEI","JBID","JCBC",
        "JCDD","JCHJ","JDAE","JDFA","JECD","JEEH","JEHI"),1,0)


## Degree

sort(table(data4$Degree),decreasing=TRUE)[1:10]
data4$Degree <- as.character(data4$Degree)
data4$Is_BTech <- ifelse(data4$Degree=="B.Tech",1,0)
data4$Is_BE <- ifelse(data4$Degree=="B.E",1,0)
data4$Is_MCA <- ifelse(data4$Degree=="MCA",1,0)
data4$Is_MBA <- ifelse(data4$Degree=="MBA",1,0)
data4$Is_BCom <- ifelse(data4$Degree=="B.Com" | data4$Degree=="B.Com (Hons.)",1,0)
data4$Is_PGDM <- ifelse(data4$Degree=="Post Graduate Dimploma in Management",1,0)
data4$Is_BSc <- ifelse(data4$Degree=="B.Sc",1,0)
data4$Is_BBA <- ifelse(data4$Degree=="Bachelor of Business Admininstration",1,0)
data4$Is_MTech <- ifelse(data4$Degree=="M.Tech",1,0)



## Stream
sort(table(data4$Stream),decreasing=TRUE)[1:10]
data4$Stream <- as.character(data4$Stream)

data4$Is_StrCSE<- ifelse(data4$Stream=="Computer Science & Engineering",1,0)
data4$Is_StrCS<- ifelse(data4$Stream=="Computer  Science",1,0)
data4$Is_StrECE<- ifelse(data4$Stream=="Electronics and Communication Engineering",1,0)
data4$Is_StrCoAp<- ifelse(data4$Stream=="Computer Application",1,0)
data4$Is_StrCommerce<- ifelse(data4$Stream=="Commerce",1,0)
data4$Is_StrIT<- ifelse(data4$Stream=="Information Technology",1,0)
data4$Is_StrME<- ifelse(data4$Stream=="Mechanical Engineering",1,0)
data4$Is_StrMarketing<- ifelse(data4$Stream=="Marketing",1,0)


##  Profile
sort(table(data4$Profile),decreasing=TRUE)[1:10]
data4$Profile <- as.character(data4$Profile)
```

```
data4$Is_Prof_intern <- ifelse(data4$Profile=="Intern",1,0)
data4$Is_Prof_No_Exp <- ifelse(data4$Profile=="No_Exp",1,0)
data4$Is_Prof_Marketing <- ifelse(data4$Profile=="Content Writing & Social Media Marketing" | data4$Profile=="Marketing",1,0)
data4$Is_Prof_Content <- ifelse(data4$Profile=="Content Writer" |data4$Profile=="Content Development" ,1,0)

##  Location

sort(table(data4$Location),decreasing=TRUE)[1:10]
data4$Location <- as.character(data4$Location)
data4$Is_LocatIIGB <- ifelse(data4$Location=="IIGB",1,0)
data4$Is_LocatIIDB <- ifelse(data4$Location=="IIDB",1,0)
data4$Is_LocatJEJJ <- ifelse(data4$Location=="JEJJ",1,0)
data4$Is_LocatIIBD <- ifelse(data4$Location=="IIBD",1,0)
data4$Is_LocatJABD <- ifelse(data4$Location=="JABD",1,0)


## Internship_Profile
sort(table(data4$Internship_Profile),decreasing=TRUE)[1:10]
data4$Internship_Profile <- as.character(data4$Internship_Profile)

data4$Is_IP_WD <- ifelse(data4$Internship_Profile=="Web Development",1,0)
data4$Is_IP_SD <- ifelse(data4$Internship_Profile=="Software Development",1,0)
data4$Is_IP_CW <- ifelse(data4$Internship_Profile=="Content Writing",1,0)
data4$Is_IP_AD <- ifelse(data4$Internship_Profile=="Android App Development",1,0)
data4$Is_IP_MK <- ifelse(data4$Internship_Profile=="Marketing",1,0)
data4$Is_IP_BD <- ifelse(data4$Internship_Profile=="Business Development",1,0)


## Skills_required
sort(table(data4$Skills_required),decreasing=TRUE)[1:10]
data4$Skills_required <- as.character(data4$Skills_required)
data4$Is_SR_No <- ifelse(data4$Skills_required=="No_Skill",1,0)

## Internship_Location

sort(table(data4$Internship_Location),decreasing=TRUE)[1:10]

data4$Internship_Location <- as.character(data4$Internship_Location)
data4$Is_IntrnLoc_IIDB <- ifelse(data4$Internship_Location =="IIDB",1,0)
data4$Is_IntrnLoc_IIBD <- ifelse(data4$Internship_Location =="IIBD",1,0)
data4$Is_IntrnLoc_IIGB <- ifelse(data4$Internship_Location =="IIGB",1,0)
data4$Is_IntrnLoc_JABD <- ifelse(data4$Internship_Location =="JABD",1,0)
data4$Is_IntrnLoc_JEJJ <- ifelse(data4$Internship_Location =="JEJJ",1,0)

# converting Internship_deadline to factor
```

```
data4$Internship_deadline <- as.character(data4$Internship_deadline)
data4$Internship_deadline <- as.Date(data4$Internship_deadline, "%d-%m-%Y")


# creating dummy variables of Current_year ,Experience_Type etc
#install.packages("dummies")
library(dummies)
dummy.data.frame
ss <- data.frame(data4$Current_year,data4$Experience_Type,data4$Internship_Type,data4$Internship_category,data4$Stipend_Type)
ss1<- dummy.data.frame(ss)
data4 <- cbind(data4,ss1)

#Dropping irrelevant variables
data4$Current_year <- NULL
data4$Experience_Type <- NULL
data4$Internship_Type <- NULL
data4$Internship_category <- NULL
data4$Stipend_Type <- NULL


# Match/ Distance between Preferred_location and Internship_Location
data4$Pref_Intern_LocMatch <- 0
data4$Pref_Intern_LocMatch[as.character(data4$Preferred_location) == as.character(data4$Internship_Location) |
as.character(data4$Preferred_location)=="No_Pref"] <- 1


# Expected_Stipend (expected by student) Stipend1(min offered) Stipend2(max offered)
# Substituting Middle value of Expected_Stipend
table(data4$Expected_Stipend)

data4$Expected_Stipend <- as.character(data4$Expected_Stipend)
data4$Expected_Stipend[data4$Expected_Stipend=="10K+"] <- 10000
data4$Expected_Stipend[data4$Expected_Stipend=="2-5K"] <- 3500
data4$Expected_Stipend[data4$Expected_Stipend=="5-10K"] <- 7500
data4$Expected_Stipend[data4$Expected_Stipend=="No Expectations"] <- 0
data4$Expected_Stipend <- as.numeric(data4$Expected_Stipend)

# creating  Feature whether Expected_Stipend < Stipend1
data4$St_EMatch <- ifelse(data4$Expected_Stipend < data4$Stipend1,1,0)


# Creating Feature about range of Stipend Offered
#Stipend2 - Stipend1

data4$Stip_range <- abs(data4$Stipend2 - data4$Stipend1)
```

```
# Creating feature Minimum_Duration is less than Internship_Duration.Months.
summary(data4$Internship_Duration.Months.)
summary(data4$Minimum_Duration)

data4$Duration_Match <- 0
data4$Duration_Match <- ifelse(data4$Minimum_Duration >= data4$Internship_Duration.Months.,1,0)


#Creating Feature whether there is a match between Institute_location and Internship_Location
data4$Inst_Intern_LocMatch <- 0
data4$Inst_Intern_LocMatch[as.character(data4$Institute_location) == as.character(data4$Internship_Location) ] <- 1


#Creating Feature whether there is a match between hometown and Internship_Location
data4$hometown_Intern_LocMatch <- 0
data4$hometown_Intern_LocMatch[as.character(data4$hometown) == as.character(data4$Internship_Location) ] <- 1



# Creating feature difference between Year_of_graduation and year of Internship_deadline
#install.packages("lubridate")
library(lubridate)
data4$Dif_Yog_IntD <- 0
data4$Dif_Yog_IntD <- data4$Year_of_graduation - year(data4$Internship_deadline)

data4$Neg_Dif_Yog_IntD <- ifelse(data4$Dif_Yog_IntD > 0, 1,0)

# tagging whether a candidate  is PG
data4$Is_PG <- 0

data4$Is_PG <- ifelse(substr(data4$Degree,1,1)=="M" | substr(data4$Degree,1,1)=="P" ,1,0)
data4$Is_PG[grep("B.E. & MBA",data4$Degree)]<- 1
data4$Is_PG[grep("B.Tech and M.Tech",data4$Degree)]<- 1
data4$Is_PG[grep("Integrated",data4$Degree)]<- 1


# tagging whether a candidate have Prof degree
data4$Is_Prof <- 0
data4$Is_Prof[grep("Tech",data4$Degree)]<- 1
data4$Is_Prof[grep("B.E",data4$Degree)]<- 1
data4$Is_Prof[grep("MCA",data4$Degree)]<- 1
data4$Is_Prof[grep("MBA",data4$Degree)]<- 1
data4$Is_Prof[grep("Management",data4$Degree)]<- 1
data4$Is_Prof[grep("Admininstration",data4$Degree)]<- 1
```

```
data4$Is_Prof[grep("Technology",data4$Degree)]<- 1
data4$Is_Prof[grep("Computer",data4$Degree)]<- 1

##Creating Feature whether there is a match between Location (Location of work experience) and Internship_Location
data4$Workex_Intern_LocMatch <- 0
data4$Workex_Intern_LocMatch[as.character(data4$Location) == as.character(data4$Internship_Location)] <- 1



# No_of_openings
# group by Internship_ID the train file to check how many applicants
# ratio of applicant to opening
RATO <- data.frame(Internship_ID = data4$Internship_ID)
RATO$Num <- 1
library(sqldf)
RATO1 <- sqldf("select Internship_ID, SUM(Num) as Num_Applicant From RATO Group BY Internship_ID")
data4 <- merge(data4,RATO1, by="Internship_ID", all.x=TRUE)

data4$Open_App_Ratio <- data4$No_of_openings/data4$Num_Applicant
#removing data4$Num_Applicant
#data4$Num_Applicant <- NULL

## any relation between Internship_deadline,Earliest_Start_Date
data4$Internship_deadline <- as.Date(data4$Internship_deadline, "%d-%m-%Y")[1:10]
data4$Diff_Intdl_StrD <- as.numeric(as.character(data4$Internship_deadline - data4$Earliest_Start_Date))

## If applicant available before internship deadline

data4$NoCross_Deadline <- ifelse(data4$Diff_Intdl_StrD > 0 ,1,0)



## Internship_deadline < 2015-01-14

data4$Internship_deadline[data4$Internship_deadline < "2015-01-13"]

table(data4$Is_Shortlisted, data4$Internship_deadline > "2015-01-13")
data4$Deadline2015 <- ifelse(data4$Internship_deadline >"2015-01-13", 1,0)



#Institute_Category
data4$Institute_Category <- as.character(data4$Institute_Category)
data4$Institute_Category <- ifelse(data4$Institute_Category=="Y",1,0)



## Dropping irrelevant variables
```

```
data5 <- data4[,c(1:2,4:5,7,10,16,18,20,21,26,30,35:139,8,9)]
names(data5) <- make.names(names(data5))

## Splitting to Train and Test

Train <- data5[data5$tag=="train",]
Test  <- data5[data5$tag=="test",]
Train$tag <- NULL
Test$tag  <- NULL
Test$Is_Shortlisted  <- NULL

write.csv(Train,"TrainD.csv",row.names=FALSE)
write.csv(Test,"TestD.csv",row.names=FALSE)
```

Data Modeling & Prediction

```
# ##  Data Understanding

# In[1]:

#import packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
get_ipython().run_line_magic('matplotlib', 'inline')

# In[2]:

# read train files in trainfiles,i.e., internships, students, shortlisted
df_internship = pd.read_csv("trainfiles/Internship/Internship.csv")
df_student = pd.read_csv("trainfiles/Student/Student.csv")
df_train = pd.read_csv("trainfiles/traincsv/train.csv")

# In[3]:

# INTERNSHIPS LISTED
df_internship.shape

# In[4]:
```

```
# STUDENT APPLICATIONS RECEIVED
df_student.shape


# In[5]:


# SHORTLISTED STUDENTS DATA for Training ML model
df_train.shape


# In[6]:


df_internship.iloc[:,:13].describe()


# In[7]:


df_student.describe()


# In[8]:


df_train.describe()


# In[9]:


# View Internships
df_internship.head(2)


# In[10]:


# VIew Students Applicants
df_student.head(2)


# In[11]:


# View Training Data
df_train.head(2)


# # Modeling


# ### The csv file saved in R environment is imported in python environment for further processing
#
```

```
# In[1]:

# Load the Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 12, 4

# In[2]:

train = pd.read_csv('../input/TrainD.csv')
test = pd.read_csv('../input/TestD.csv')

# In[3]:

train.shape,test.shape

# In[4]:

train.dtypes

# In[5]:

from sklearn.ensemble import GradientBoostingClassifier
from sklearn.model_selection import cross_val_score, GridSearchCV
from sklearn import metrics

# In[6]:

target='Is_Shortlisted'
Internship_ID = 'Internship_ID'
Student_ID = 'Student_ID'

# In[7]:

train['Is_Shortlisted'].value_counts()

# In[8]:
```

```
def modelfit(alg, dtrain, dtest, predictors, performCV=True, printFeatureImportance=True, cv_folds=5):
    #Fit the algorithm on the data
    alg.fit(dtrain[predictors], dtrain['Is_Shortlisted'])

    #Predict training set:
    dtrain_predictions = alg.predict(dtrain[predictors])
    dtrain_predprob = alg.predict_proba(dtrain[predictors])[:,1]

    #Perform cross-validation:
    if performCV:
        cv_score = cross_val_score(alg, dtrain[predictors], dtrain['Is_Shortlisted'], cv=cv_folds, scoring='roc_auc')

    #Print model report:
    print ("\nModel Report")
    print ("Accuracy : %.4g" % metrics.accuracy_score(dtrain['Is_Shortlisted'].values, dtrain_predictions))
    print ("AUC Score (Train): %f" % metrics.roc_auc_score(dtrain['Is_Shortlisted'], dtrain_predprob))

    if performCV:
        print ("CV Score : Mean - %.7g | Std - %.7g | Min - %.7g | Max - %.7g" %
(np.mean(cv_score),np.std(cv_score),np.min(cv_score),np.max(cv_score)))

    #Print Feature Importance:
    if printFeatureImportance:
        feat_imp = pd.Series(alg.feature_importances_, predictors).sort_values(ascending=False)
        feat_imp.plot(kind='bar', title='Feature Importances')
        plt.ylabel('Feature Importance Score')

# ### Baseline Model
# * Since here the criteria is AUC, simply predicting the most prominent class would give an AUC of 0.5 always.
# * Another way of getting a baseline model is to use the algorithm without tuning, i.e. with default parameters.
#

# In[9]:

#Choose all predictors except target & IDcols
predictors = [x for x in train.columns if x not in [target, Internship_ID,Student_ID]]
gbm0 = GradientBoostingClassifier(random_state=10)

# In[10]:
```

```
# to check if there is any NaN
np.any(np.isnan(train)), np.all(np.isfinite(train))

# In[11]:

train.fillna(0,inplace=True) # fill 0 inpalce of NaN
# to check after filling NaN
np.any(np.isnan(train)), np.all(np.isfinite(train))

# In[12]:

modelfit(gbm0, train, test, predictors)

# In[13]:

pd.Series(gbm0.feature_importances_, predictors).sort_values(ascending=False)[1:30]

# In[14]:

#Taking important features as predictors
predictors1=
['Stip_range','Num_Applicant','No_of_openings','Internship_Duration.Months.','Is_SR_No','Diff_Intdl_StrD',
'Minimum_Duration','Performance_10th','data4.Internship_categoryPart.time','Num_Exp_Row','Duration_Match',
'hometown_Intern_LocMatch','data4.Internship_Typeregular','data4.Stipend_Typeunpaid','Is_IP_MK','Inf_hometow
n',
'Is_IP_CW','Institute_Category','data4.Stipend_Typeperformance','data4.Internship_Typevirtual','data4.Experience_
Typeinternship','Inst_Intern_LocMatch','Is_Prof','Performance_12th','Is_IntrnLoc_JABD','data4.Stipend_Typefixed',
'Workex_Intern_LocMatch','Is_IP_AD','Performance_UG','Is_PlIJCE',
'data4.Internship_categoryFull.Time','data4.Stipend_Typevariable','Is_IntrnLoc_IIGB','Is_InstLoc_IIIF','Is_IP_BD','
NoCross_Deadline','Is_PlNo_Pref','Is_PlIHJB','Is_StrMarketing','Is_IntrnLoc_IIDB','Is_IP_WD','Is_IntrnLoc_IIBD'
,'Is_IntrnLoc_JEJJ','Is_IP_SD','Is_InstLoc_IIDB','Is_StrCommerce','Exp_tenure','Is_Part_Time','St_EMatch','Dif_Y
og_IntD','data4.Experience_Typeacademic_project','data4.Current_year2','data4.Experience_TypeNo_Exp','Expecte
d_Stipend','Is_Prof_Marketing','Is_MTech','Is_PlIIDB']

# In[15]:

#Choose important predictors and excepting target & IDcols
predictors = predictors1
param_test1 = {'n_estimators':range(20,81,10)}
```

```
gsearch1 = GridSearchCV(estimator = GradientBoostingClassifier(learning_rate=0.1, min_samples_split=500,
min_samples_leaf=50,max_depth=8,max_features='sqrt', subsample=0.8,random_state=10), param_grid =
param_test1, scoring='roc_auc',n_jobs=4,iid=False, cv=5)
gsearch1.fit(train[predictors],train[target])

# In[20]:

#gsearch1.grid_scores_
gsearch1.cv_results_
gsearch1.best_params_
gsearch1.best_score_

# In[21]:

#Grid seach on subsample and max_features
predictors = predictors1
param_test2 = {'max_depth':range(2,7,2), 'min_samples_split':range(100,400,100)}
gsearch2 = GridSearchCV(estimator = GradientBoostingClassifier(learning_rate=0.1, n_estimators=70,
max_features='sqrt', subsample=0.8, random_state=10), param_grid = param_test2,
scoring='roc_auc',n_jobs=4,iid=False, cv=5)
gsearch2.fit(train[predictors],train[target])

# In[22]:

#gsearch2.grid_scores_
gsearch2.cv_results_
gsearch2.best_params_
gsearch2.best_score_

# In[23]:

#Grid seach on subsample and max_features
predictors = predictors1
param_test3 = {'min_samples_split':range(50,200,50), 'min_samples_leaf':range(30,71,10)}
gsearch3 = GridSearchCV(estimator = GradientBoostingClassifier(learning_rate=0.1,
n_estimators=70,max_depth=4, max_features='sqrt', subsample=0.8, random_state=10), param_grid = param_test3,
scoring='roc_auc',n_jobs=4,iid=False, cv=5)
gsearch3.fit(train[predictors],train[target])

# In[24]:
```

```
#gsearch3.grid_scores_
gsearch3.cv_results_
gsearch3.best_estimator_
gsearch3.best_score_

# In[25]:

modelfit(gsearch3.best_estimator_, train, test, predictors)

# In[26]:

#Tune max_features:
#Grid seach on subsample and max_features
param_test4 = {'max_features':range(5,20,2)}
gsearch4 = GridSearchCV(estimator = GradientBoostingClassifier(learning_rate=0.1,
n_estimators=70,max_depth=4, min_samples_split=150, min_samples_leaf=70, subsample=0.8, random_state=10),
param_grid = param_test4, scoring='roc_auc',n_jobs=4,iid=False, cv=5)
gsearch4.fit(train[predictors],train[target])

# In[27]:

#gsearch4.grid_scores_
gsearch4.cv_results_
gsearch4.best_params_
gsearch4.best_score_

# In[28]:

### Step3- Tune Subsample and Lower Learning Rate
#Grid seach on subsample and max_features
param_test5 = {'subsample':[0.7,0.75,0.8,0.85,0.9]}
gsearch5 = GridSearchCV(estimator = GradientBoostingClassifier(learning_rate=0.1,
n_estimators=70,max_depth=4, min_samples_split=150, min_samples_leaf=70, random_state=10,
max_features=7), param_grid = param_test5, scoring='roc_auc',n_jobs=4,iid=False, cv=5)
gsearch5.fit(train[predictors],train[target])

# In[29]:

#gsearch5.grid_scores_
```

```
gsearch5.cv_results_
gsearch5.best_params_
gsearch5.best_score_

# In[30]:

#  With all tuned lets try reducing the learning rate and proportionally increasing the number of estimators to get
#  more robust results:
#Choose all predictors except target & IDcols
predictors = predictors1
gbm_tuned_1 = GradientBoostingClassifier(learning_rate=0.05, n_estimators=120,max_depth=4,
min_samples_split=150,min_samples_leaf=70, subsample=0.7, random_state=10, max_features=5)
modelfit(gbm_tuned_1, train, test, predictors)

# In[31]:

est = GradientBoostingClassifier(learning_rate=0.05, n_estimators=120,max_depth=4, min_samples_split=150,
min_samples_leaf=70, subsample=0.7, random_state=10, max_features=5)

# In[32]:



est.fit(train[predictors],train[target])

# In[33]:



test.fillna(0,inplace=True)

# predict probabilities
prob = est.predict_proba(test[predictors])[:,1]

# In[34]:

test1=test
test1['Is_Shortlisted']=prob[:]
test1.to_csv('DYD_SEC1.csv', columns=['Internship_ID','Student_ID','Is_Shortlisted'],index=False)
```