Introduction to Data Mining
Graduate Admission Prediction system

Done by:
Noor Mamlook,
Sanuj Kumar,
Shreya Banwaskar

Outline

1. Introduction
2. ML model
3. User interface
4. Demo

Introduction

Introduction

- **Problem Statement**
  - Number of students attending graduate school increases
  - Acceptance requirements in universities become more challenging
  - Students submit their applications to several universities
  - Wait until one of them replies

- **Our Solution:**

  Develop a system which predicts the probability of a student being accepted in a specific university

Why graduate school ?

1. The necessary of graduate certificate for the persons professional field

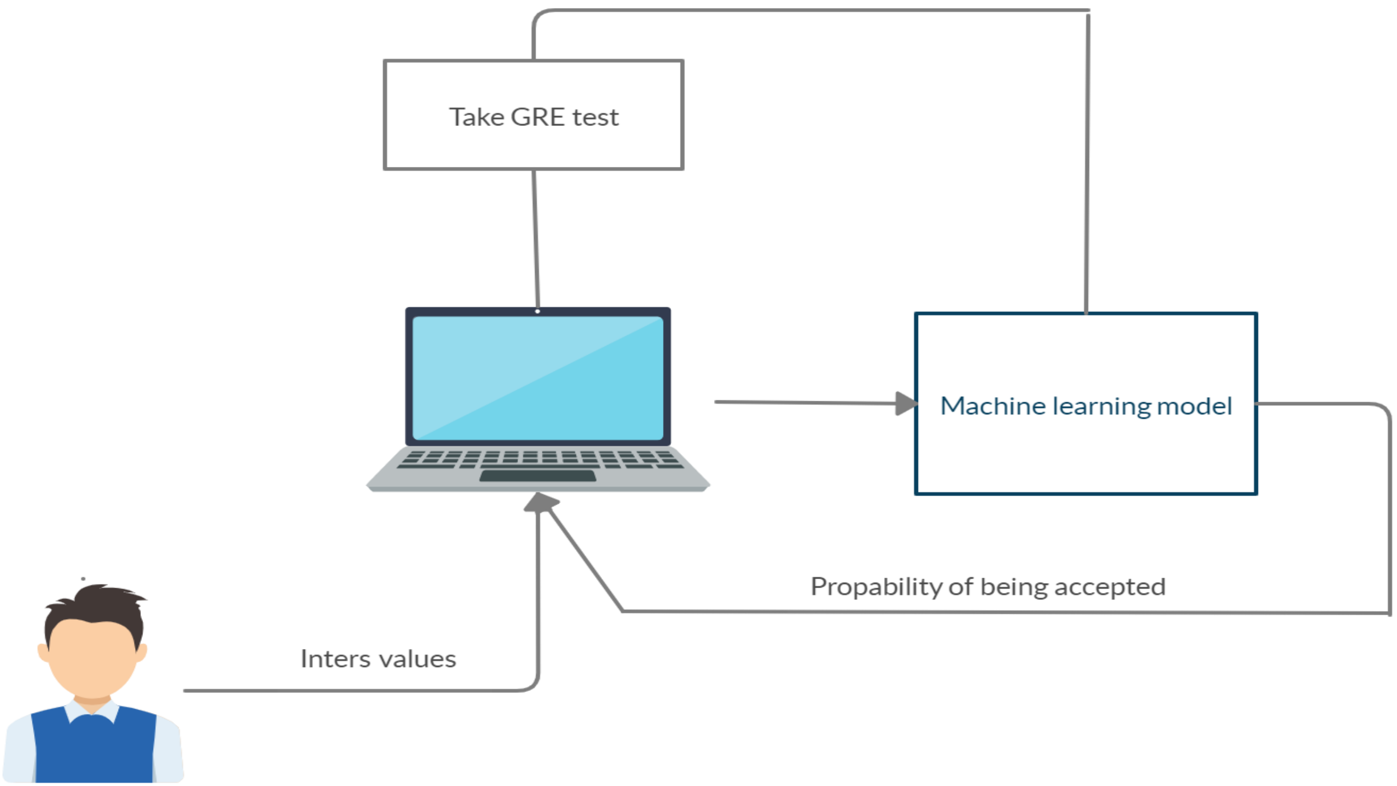
2. Advance the persons own career

3. In touch with professionals and classmates who share the same interests

Graduate school admission requirements

1. Transcripts, GPA
2. Test scores
   A. GRE
   B. TOEFL
3. The letter of recommendation (LOR)
4. Statement of purpose (SOP)

Objective of work



Machine learning model

Data

- Records of grad students regarding the graduate admission

- The data is spread across a table

- Download from Kaggle

	A	B	C	D	E	F	G	H	I
1	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
2	1	337	118	4	4.5	4.5	9.65	1	0.92
3	2	324	107	4	4	4.5	8.87	1	0.76
4	3	316	104	3	3	3.5	8	1	0.72
5	4	322	110	3	3.5	2.5	8.67	1	0.8
6	5	314	103	2	2	3	8.21	0	0.65
7	6	330	115	5	4.5	3	9.34	1	0.9

Data

- There are 8 attributes:
    - GRE score
    - TOFEL score
    - University rating
    - SOP
    - LOR
    - CGPA
- 400 instances
- No missing values.
- All attributes are numerical values

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
Serial No.           400 non-null int64
GRE Score            400 non-null int64
TOEFL Score          400 non-null int64
University Rating    400 non-null int64
SOP                  400 non-null float64
LOR                  400 non-null float64
CGPA                 400 non-null float64
Research             400 non-null int64
Chance of Admit      400 non-null float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
None
```

Data

- CGPA, GRE, TOEFL, University rating, SOP, LOR and research have a correlation close to 1
- Meaning that there is a strong positive correlation
- The change of admittance tends to increase as those attributes increase

```
Chance of Admit    1.000000
CGPA               0.883593
GRE                0.812884
TOEFL              0.797465
u_rating           0.727587
SOP                0.681096
LOR                0.672106
Research           0.562526
SN                 0.050298
Name: Chance of Admit, dtype: float64
```

Preparing the data for machine learning algorithms

- No missing values so no need to clean the data

- No categorical attributes with text values to handle

- The numerical attributes have different scales we scaled the data to range from 0 to 1

Data Mining Tasks

Classification Tasks:

- Decision Tree Classifier

	precision	recall	f1-score	support
0	1.00	0.99	0.99	67
1	0.93	1.00	0.96	13
accuracy			0.99	80
macro avg	0.96	0.99	0.98	80
weighted avg	0.99	0.99	0.99	80

- K Nearest Neighbors

	precision	recall	f1-score	support
0	0.88	1.00	0.94	67
1	1.00	0.31	0.47	13
accuracy			0.89	80
macro avg	0.94	0.65	0.70	80
weighted avg	0.90	0.89	0.86	80

- Support Vector Classifier (SVC)

	precision	recall	f1-score	support
0	0.84	1.00	0.91	67
1	0.00	0.00	0.00	13
accuracy			0.84	80
macro avg	0.42	0.50	0.46	80
weighted avg	0.70	0.84	0.76	80

Regression Tasks:

- Decision Tree Regressor

	precision	recall	f1-score	support
0	1.00	0.99	0.99	67
1	0.93	1.00	0.96	13
accuracy			0.99	80
macro avg	0.96	0.99	0.98	80
weighted avg	0.99	0.99	0.99	80

- Linear Regression

	precision	recall	f1-score	support
0	0.96	1.00	0.98	67
1	1.00	0.77	0.87	13
accuracy			0.96	80
macro avg	0.98	0.88	0.92	80
weighted avg	0.96	0.96	0.96	80

- Logistic Regression

	precision	recall	f1-score	support
0	0.93	0.94	0.93	67
1	0.67	0.62	0.64	13
accuracy			0.89	80
macro avg	0.80	0.78	0.79	80
weighted avg	0.88	0.89	0.89	80

- Support Vector Regressor (SVR)

	precision	recall	f1-score	support
0	0.84	1.00	0.91	67
1	0.00	0.00	0.00	13
accuracy			0.84	80
macro avg	0.42	0.50	0.46	80
weighted avg	0.70	0.84	0.76	80

Best Model

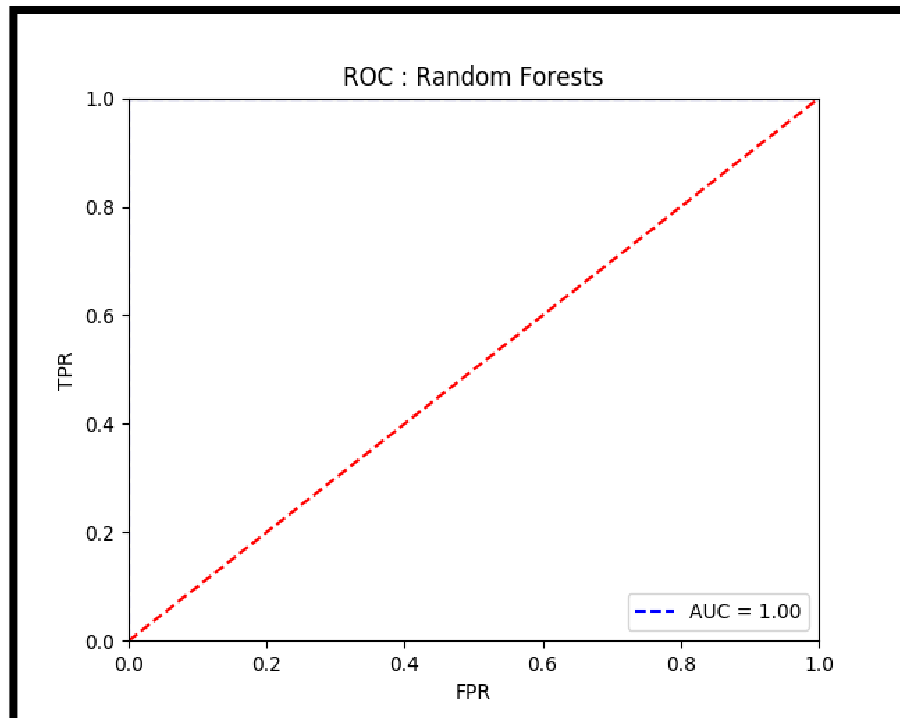
	Accuracy	ROC(AUC)	Runtime	F1-Score
Decision Tree	98.75%	0.99	0.0053s	0.99
Decision Tree Regressor	98.75%	-	0.0046s	0.99
KNN	88.75%	0.93	0.0136s	0.94
Linear Regression	96.25%	-	0.0044s	0.98
Logistic Regression	88.75%	0.95	0.0084s	0.93
SVR	83.75%	-	0.0114s	0.91
SVC	83.75%	-	0.0165s	0.91
Random Forest	100%	1.00	0.0207s	1.00

Random Forests

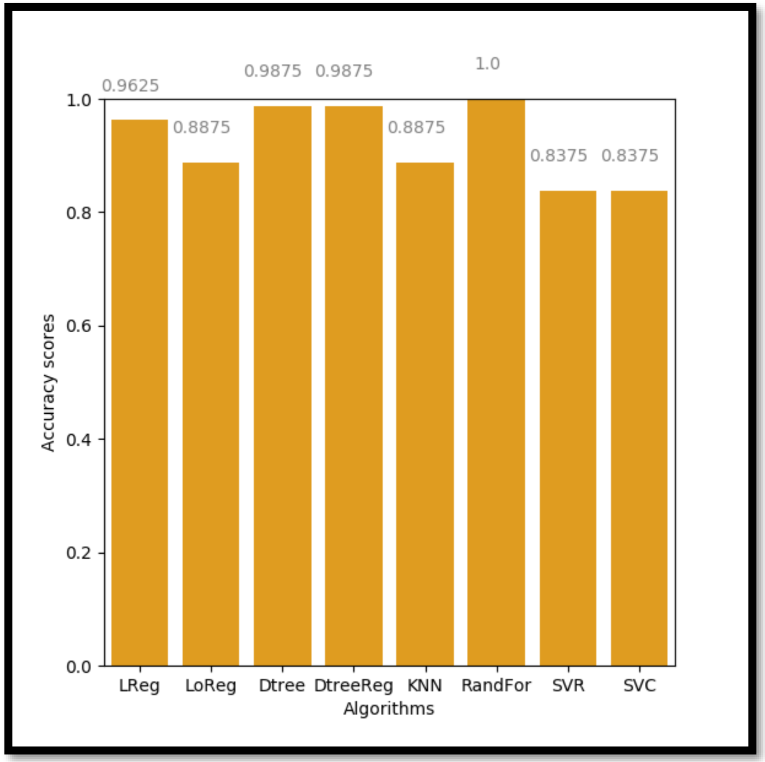
- We took an ensemble of 60 trees(n_estimators)
- Max Depth : 17
- Max Features : 2

- The fine tuned parametric values were found using GridSerachCV

	precision	recall	f1-score	support
0	1.00	1.00	1.00	67
1	1.00	1.00	1.00	13
accuracy			1.00	80
macro avg	1.00	1.00	1.00	80
weighted avg	1.00	1.00	1.00	80



Comparison



- This is the plot showing the accuracy scores all the different models we used in this project so far..

- The Random forest has the highest accuracy as discussed earlier.
- The Support Vector regressor and classifier were not as good in terms of accuracy as dimensionality of data was not very high

Computing weighted average (Ensemble):

We also ensembled the different models used in the project so far, into a single model so as computed the weighted average accuracy score

The accuracy of the ensembled model turned out to be 100% which is the same as that of Random Forest.

So, we can conclude that both the ensemble methods have the best performance which is justifiable by the theory behind it.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	67
1	1.00	1.00	1.00	13
accuracy			1.00	80
macro avg	1.00	1.00	1.00	80
weighted avg	1.00	1.00	1.00	80

Significance Testing

In order to test if a classification score is significative a technique in repeating the classification procedure after randomizing, permuting, the labels. The p-value is then given by the percentage of runs for which the score obtained is greater than the classification score obtained in the first place.

- If the p-value is sufficiently small, we reject the null hypothesis, H0 and say that the result is statistically significant.
- If the p-value is not sufficiently small, we say that we fail to reject the null hypothesis

For our data sample we are using the **permutation_test_score** sklearn for evaluating the significance of 5 cross-validated scores with 100 permutations. The model used for the evaluation is the Random Forest classifier as it has the best accuracy score among all the models.

The p-value for our sample turned out to be nearly 0.01.
We achieved a classification score of 96.56%

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        67
           1       1.00      1.00      1.00        13

    accuracy                           1.00        80
   macro avg       1.00      1.00      1.00        80
weighted avg       1.00      1.00      1.00        80

Classification score 0.9656173687423688 (pvalue : 0.009900990099009901)

Perm Scores[0.81256105 0.81251145 0.80953373 0.81280601 0.80010913 0.81261065
 0.80958181 0.82193758 0.79986264 0.81876297 0.81256105 0.80621184
 0.81891026 0.81881258 0.80328068 0.82496642 0.81265873 0.81876297
 0.80288996 0.81583333 0.800058   0.80342643 0.81251145 0.81251145
 0.79996032 0.82193758 0.80606456 0.7969826  0.81265873 0.80943605
 0.81251145 0.80020681 0.81890873 0.81246337 0.81256105 0.81568605
 0.8155899  0.81563797 0.80313645 0.82193758 0.81568605 0.8032326
 0.80630952 0.81886065 0.80342796 0.82511065 0.80655296 0.81886065
 0.80948565 0.81558837 0.800058   0.81265873 0.81563797 0.81588141
 0.81578373 0.80943452 0.79365919 0.80933684 0.82833639 0.79981303
 0.80630952 0.80943605 0.81554029 0.81260913 0.80943452 0.81255952
 0.81866529 0.81886065 0.80000992 0.78736111 0.79366224 0.81568605
 0.81265873 0.81886065 0.79385607 0.82506258 0.80010913 0.8250641
 0.80308684 0.79996184 0.80328068 0.80958181 0.803183   0.79390415
 0.81241377 0.81251297 0.81573565 0.81573565 0.81260913 0.80645833
 0.81236416 0.800058   0.81886065 0.79400488 0.79678571 0.80958181
 0.8064072  0.81881258 0.81890873 0.8187149 ]
```

Testing in UI

- The final step is to validate the model on the test set.
- The process is as follows: for each input sequence (question and paragraph), embedding vector is calculated.
- The embedding vectors are feed into the encoder.
- The model calculates the start vector and end vector and the maximum probability of both vectors.

Challenges

- Data Collection (SOP, LOR) : Due to confidentiality [Limited publicly available responses]
- Written responses for NLP task { Requirement 1000, Got ~100}
- Prediction may not be so accurate for universities with specific requirements.
- Dataset – The datasets used here has just 500 records which is not very big

Future scope

- Run NLP MODEL (BERT) on TPU (Tensor Processing Unit) Instead of CPU or GPU
- Huge Data Set, At least More than 500
- Collecting SOP,s and LOR's from students with permission
- Getting appropriate accuracy for every university

Conclusion

- Data Preprocessing - Collecting data.
- Creating test sets - We used the scikit-learn function (StratifiedShuffleSplit) to create the test set by selecting 20% of the dataset.
- The scikit-learn function ensures that the test set will remain consistent across multiple runs.
- Data Mining Tasks – Classification and Regression
- Motivation – As we are international students we have faced issues with the Admission process back in our country, So we decided to make something that will help other students.
- NLP tasks – BERT
- The reason for building BERT base is to compare results with OpenAI GPT.

Demo

THANK YOU…

We hope you liked this ☺

Any Questions??