Richter Earthquake Damage

By Jessica Daniella Girsang (U2221579J)
and Muhammad Aditya Alifadhilah (U2120345C)

CONTENTS

01

PROBLEM STATEMENT

Q: Which characteristics of buildings are most relevant in predicting whether it will withstand earthquakes?

Motivation

A destructive earthquake of 7.8 magnitude occurred in **Nepal** in **April 2015.**

This earthquake claimed almost 9,000 lives and around $10 billion in damages. Millions of people lost everything and became homeless in a few moments.

Objectives

Our objective is to see how each buildings' characteristics affects their damage grade in the case of an earthquake .

By finding the most important features, we can help developers during reconstruction so that they can apply corrections. especially on these characteristics, and minimize their risk of experiencing the same level of damage in the case of another earthquake.

The Dataset

The data is used in a competition hosted by **DRIVENDATA** "Richter's Predictor: Modeling Earthquake Damage" and contains information such as damage grade, building conditions, and variables involved

02

EDA

Exploratory Data Analysis

- Data Preparation
- Data Cleaning
- Numerical Data
- Categorical Data
- Correlations

DATA PREPARATION

Check Null and Duplicate Data
Encode Object Data to Categorical Data
Merge Numeric and Categorical Feature
Clear Outliers

NUMERICAL

```
Int64Index: 260601 entries, 802906 to 747594
Data columns (total 29 columns):
 #   Column                                  Non-Null Count   Dtype
---  ------                                  --------------   -----
 0   geo_level_1_id                          260601 non-null  int64
 1   geo_level_2_id                          260601 non-null  int64
 2   geo_level_3_id                          260601 non-null  int64
 3   count_floors_pre_eq                     260601 non-null  int64
 4   age                                     260601 non-null  int64
 5   area_percentage                         260601 non-null  int64
 6   height_percentage                       260601 non-null  int64
 7   has_superstructure_adobe_mud            260601 non-null  int64
 8   has_superstructure_mud_mortar_stone     260601 non-null  int64
 9   has_superstructure_stone_flag           260601 non-null  int64
 10  has_superstructure_cement_mortar_stone  260601 non-null  int64
 11  has_superstructure_mud_mortar_brick     260601 non-null  int64
 12  has_superstructure_cement_mortar_brick  260601 non-null  int64
 13  has_superstructure_timber               260601 non-null  int64
 14  has_superstructure_bamboo               260601 non-null  int64
 15  has_superstructure_rc_non_engineered    260601 non-null  int64
 16  has_superstructure_rc_engineered        260601 non-null  int64
 17  has_superstructure_other                260601 non-null  int64
 18  count_families                          260601 non-null  int64
 19  has_secondary_use_hotel                 260601 non-null  int64
 20  has_secondary_use_rental                260601 non-null  int64
 21  has_secondary_use_institution           260601 non-null  int64
 22  has_secondary_use_school                260601 non-null  int64
 23  has_secondary_use_industry              260601 non-null  int64
 24  has_secondary_use_health_post           260601 non-null  int64
 25  has_secondary_use_gov_office            260601 non-null  int64
 26  has_secondary_use_use_police            260601 non-null  int64
 27  has_secondary_use_other                 260601 non-null  int64
 28  damage_grade                            260601 non-null  int64
dtypes: int64(29)
```

29 Numerical Columns

CATEGORICAL

#	Column	Non-Null Count	Dtype
0	land_surface_condition_n	260601 non-null	uint8
1	land_surface_condition_o	260601 non-null	uint8
2	land_surface_condition_t	260601 non-null	uint8
3	foundation_type_h	260601 non-null	uint8
4	foundation_type_i	260601 non-null	uint8
5	foundation_type_r	260601 non-null	uint8
6	foundation_type_u	260601 non-null	uint8
7	foundation_type_w	260601 non-null	uint8
8	roof_type_n	260601 non-null	uint8
9	roof_type_q	260601 non-null	uint8
10	roof_type_x	260601 non-null	uint8
11	ground_floor_type_f	260601 non-null	uint8
12	ground_floor_type_m	260601 non-null	uint8
13	ground_floor_type_v	260601 non-null	uint8
14	ground_floor_type_x	260601 non-null	uint8
15	ground_floor_type_z	260601 non-null	uint8
16	other_floor_type_j	260601 non-null	uint8
17	other_floor_type_q	260601 non-null	uint8
18	other_floor_type_s	260601 non-null	uint8
19	other_floor_type_x	260601 non-null	uint8
20	position_j	260601 non-null	uint8
21	position_o	260601 non-null	uint8
22	position_s	260601 non-null	uint8
23	position_t	260601 non-null	uint8
24	plan_configuration_a	260601 non-null	uint8
25	plan_configuration_c	260601 non-null	uint8
26	plan_configuration_d	260601 non-null	uint8
27	plan_configuration_f	260601 non-null	uint8
28	plan_configuration_m	260601 non-null	uint8
29	plan_configuration_n	260601 non-null	uint8
30	plan_configuration_o	260601 non-null	uint8
31	plan_configuration_q	260601 non-null	uint8
32	plan_configuration_s	260601 non-null	uint8
33	plan_configuration_u	260601 non-null	uint8
34	legal_ownership_status_a	260601 non-null	uint8
35	legal_ownership_status_r	260601 non-null	uint8
36	legal_ownership_status_v	260601 non-null	uint8
37	legal_ownership_status_w	260601 non-null	uint8

dtypes: uint8(38)

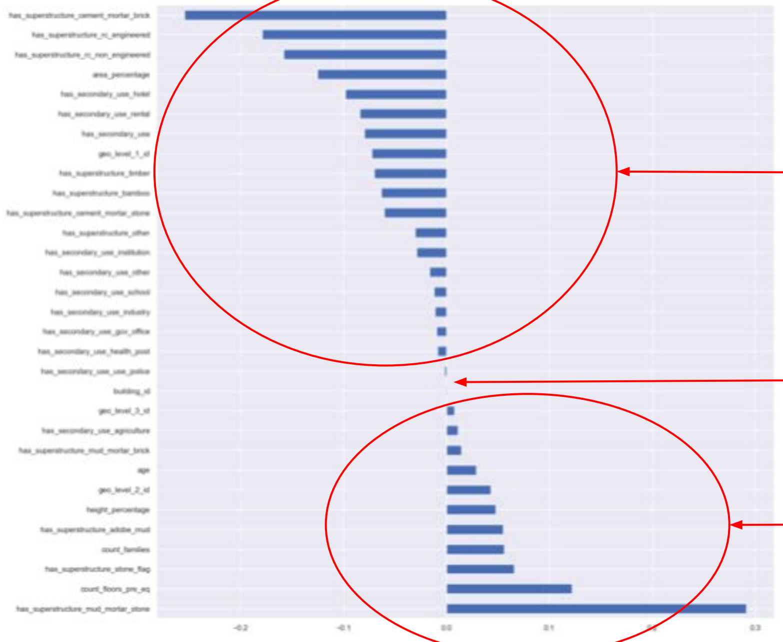
38 Categories

CORRELATIONS

POSITIVE AND NEGATIVE

In [33]:
```
plt.figure(figsize=(16,16))

corr_matrix['damage_grade'].drop('damage_grade').sort_values(ascend
plt.show()
```

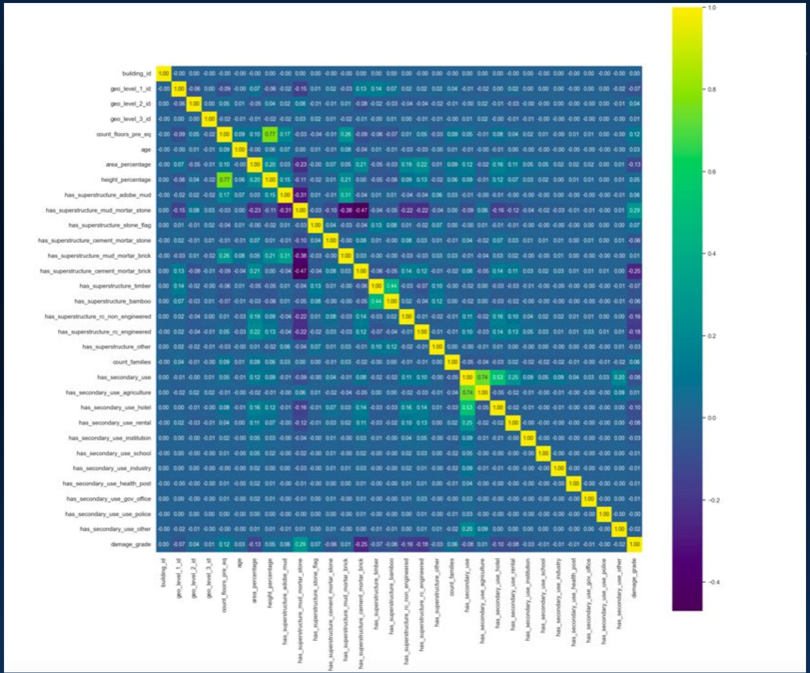


NEGATIVE CORRELATIONS

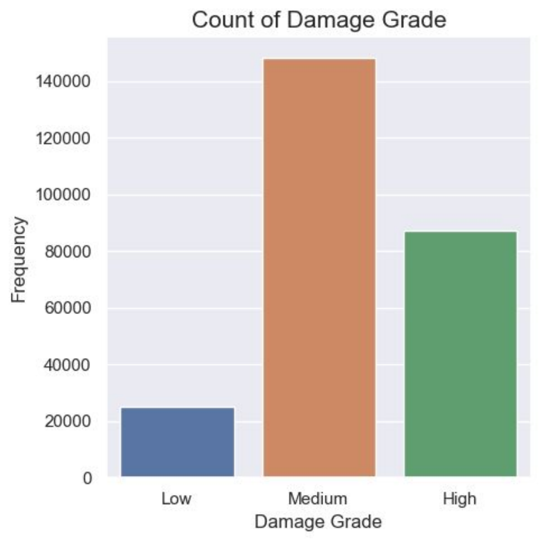
NO CORRELATIONS

POSITIVE CORRELATIONS

CORRELATION MATRIX



DAMAGE GRADES SUMMARY



Summary

```
Number of building with Low Damage level: 25124
Number of building with Medium Damage level: 148259
Number of building with High Damage level: 87218

Percentage of building with Low Damage Level: 9.64%
Percentage of building with Medium Damage Level: 56.89%
Percentage of building with High Damage Level: 33.47%
```

We make a new dataframe with y as Damage Grade and X as the other features.

DETERMINING MOST RELEVANT CATEGORIES

Using SelectKBest and Chi 2 score to get the Top 10 Categorical Variables

```
#We use Chi2score>100 and k=38
category_rank_feature = SelectKBest(score_func=chi2, k=38)
```

Feature	
ground_floor_type_v	32465.421066
roof_type_x	28048.595012
foundation_type_i	27929.304672
other_floor_type_s	18549.408221
foundation_type_w	8315.794578
other_floor_type_j	7422.919931
foundation_type_r	6391.952318
foundation_type_u	5494.248443
other_floor_type_q	5108.461280
ground_floor_type_f	3684.892346

03

MACHINE LEARNING

4 Machine Learning Models:

- Decision Tree
- Logistic Regression
- Random Forest
- Extreme Gradient Boost

Machine Learning Models

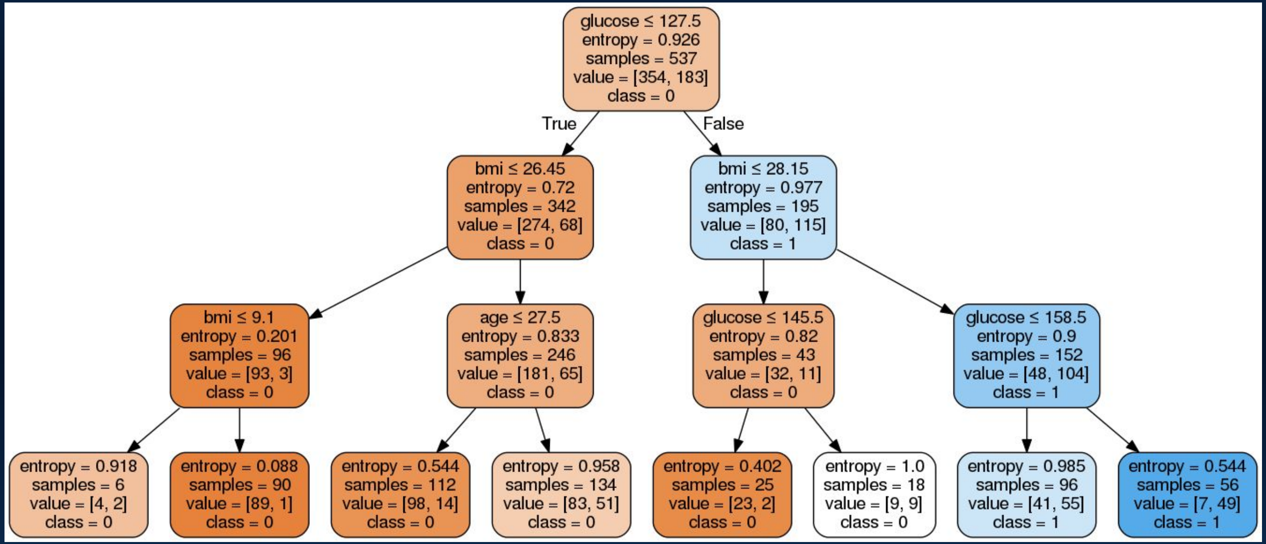
Decision Tree

Logistic Regression

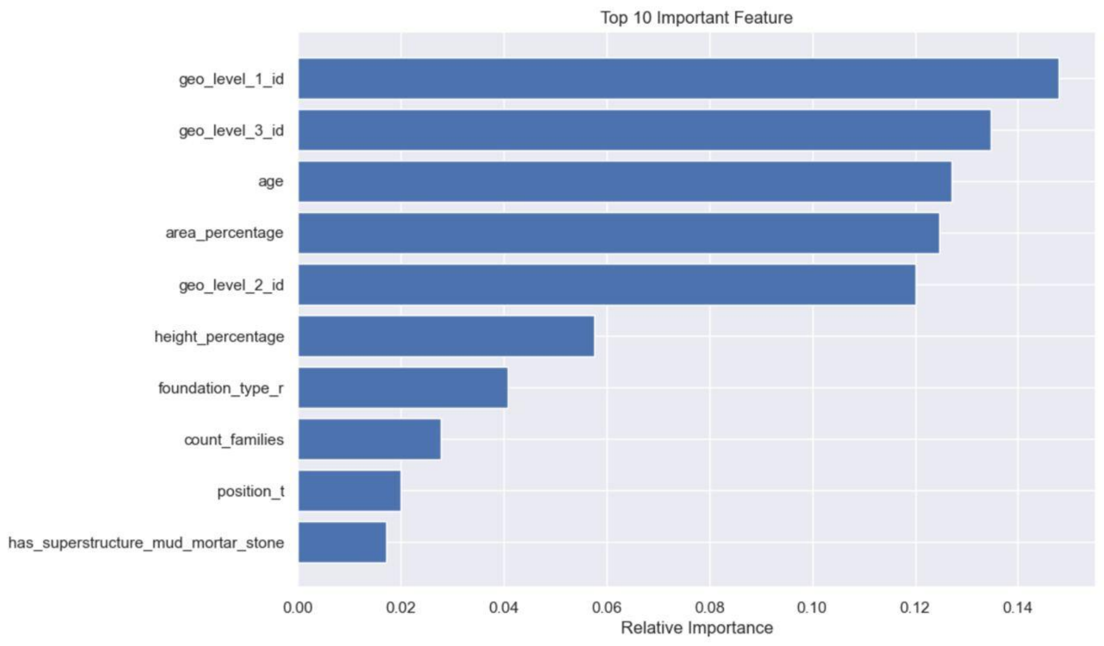
Random Forest

XGB

Decision Tree



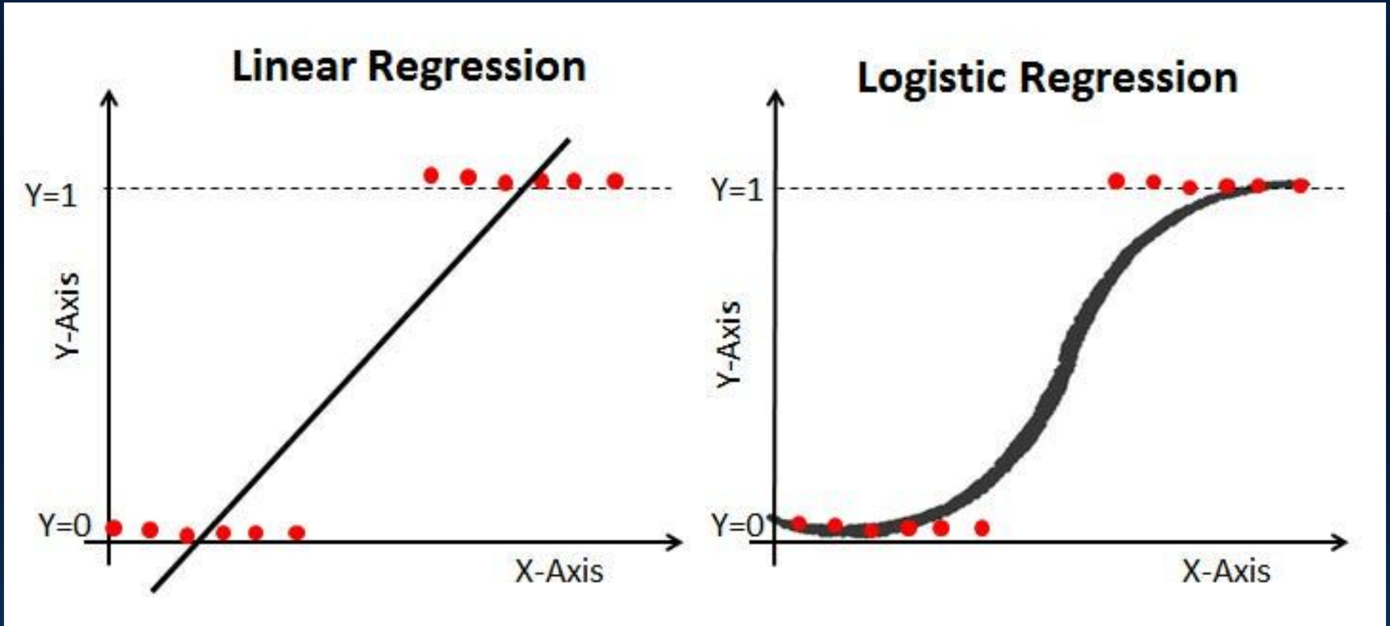
Decision Tree



Machine Learning Models

- Decision Tree
- Logistic Regression
- Random Forest
- XGB

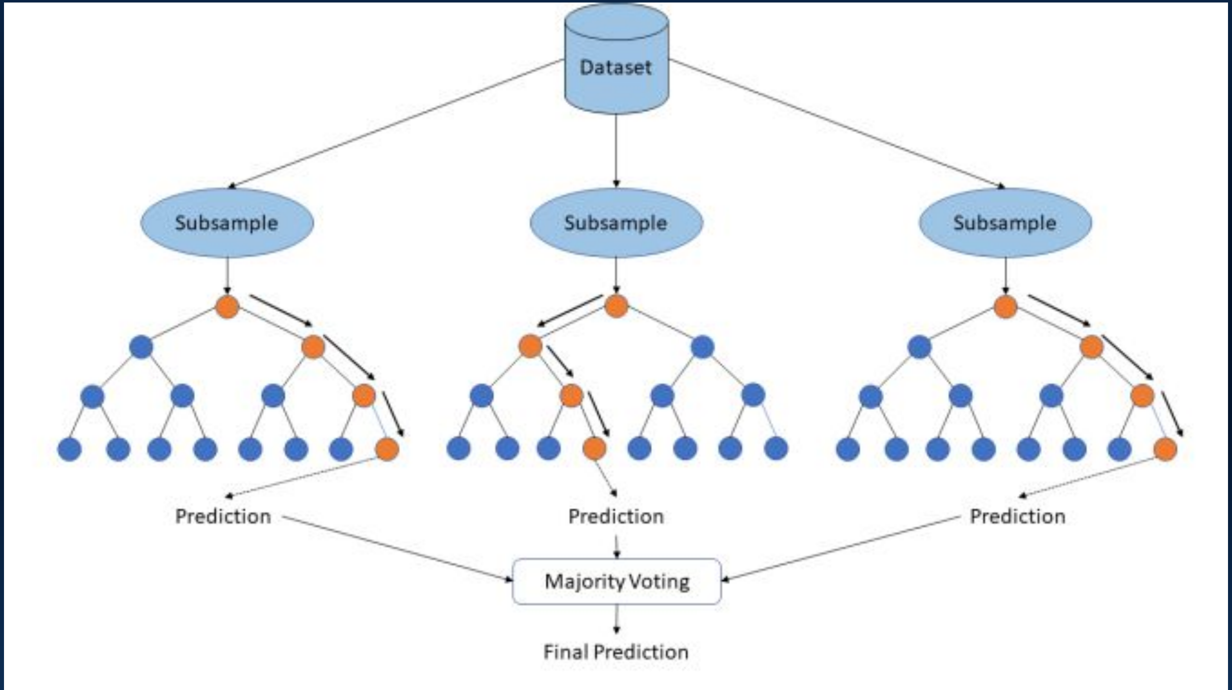
Logistic Regression



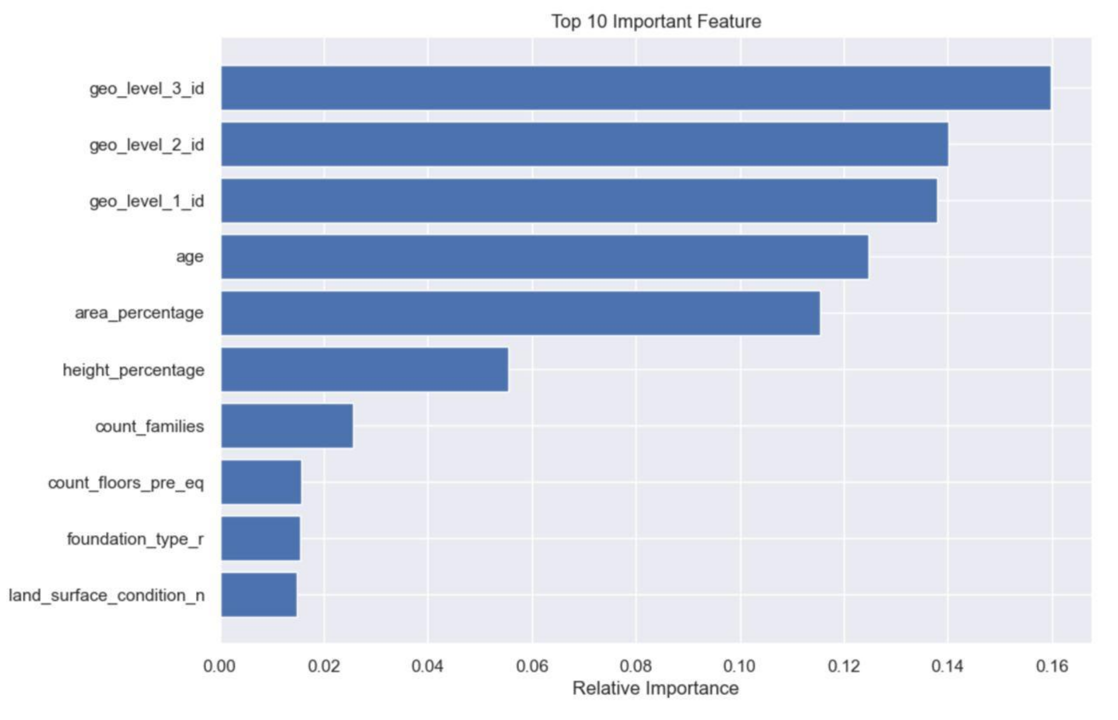
Machine Learning Models

- Decision Tree
- Logistic Regression
- Random Forest
- XGB

Random Forest



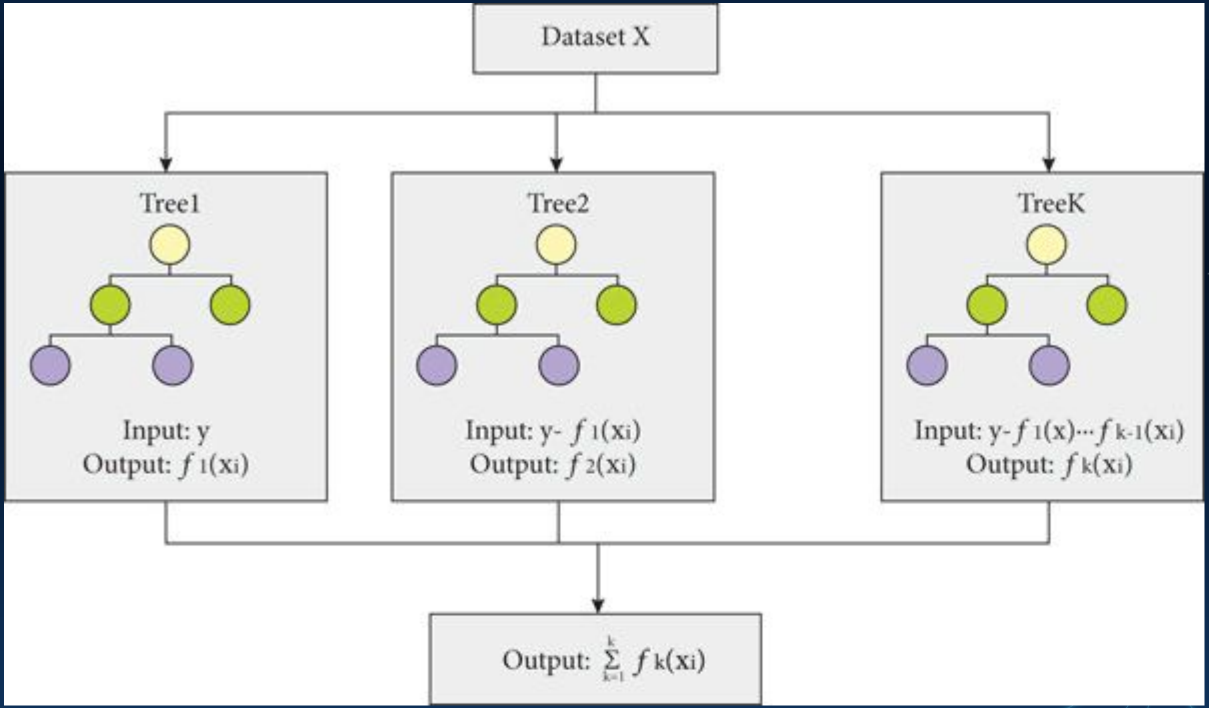
Random Forest



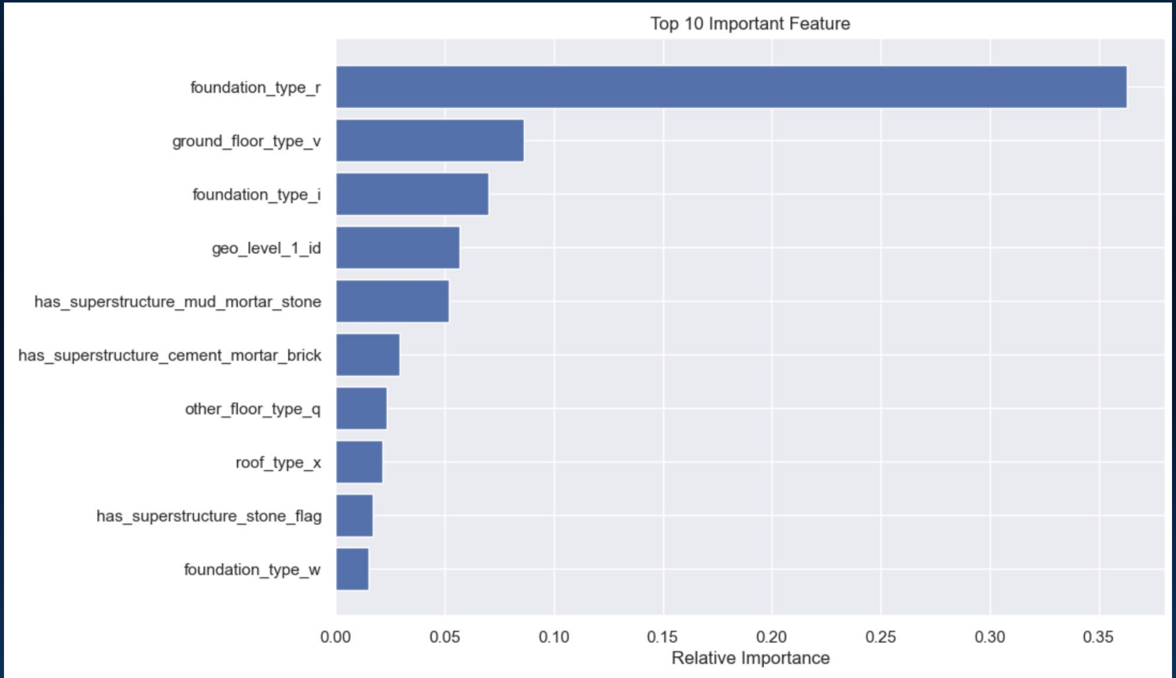
Machine Learning Models

- Decision Tree
- Logistic Regression
- Random Forest
- XGB

Extreme Gradient Boosting



Extreme Gradient Boosting



MODEL ANALYSIS - COMPARISON

PERFORMANCE METRICS

F1 SCORE MICRO

Calculated using precision and recall of the test.

	Algorithm	CV F1 Score
0	Logistic Regression	0.56938
1	Decision Tree	0.6544
2	Random Forest	0.71302
3	Extreme Gradient Boosting	0.72639

Scores

04

CONCLUSION

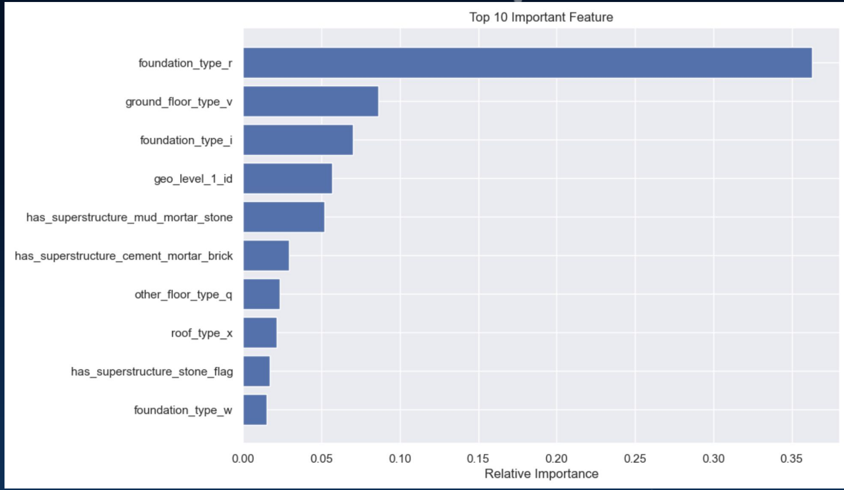
Performance comparison and evaluation, Best model, and Conclusion

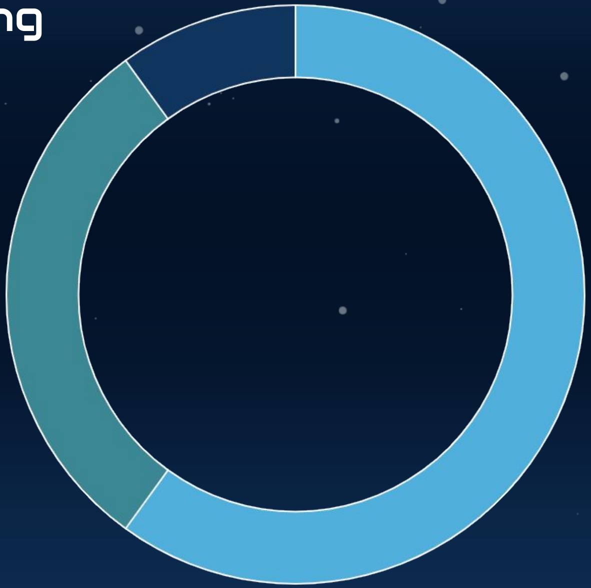
Best Model Performance

1

Extreme Gradient Boosting

F1 Score: 0.7264





CONCLUSION

1

From Machine Learning models, we found that **foundation and ground floor type** are the most important features in predicting damage grade.

2

From EDA, we found that only 9.64% building that has low Damage level, the rest is 56.89% building has Medium Damage level, and 33.47% building has High Damage level

3

We found that these models may help hotel developers to reconstruct after the earthquake and know the **most important factors** which lead to earthquake stability.

CONTRIBUTIONS

Jan	Feb	Mar	Apr	May	Jun

Task	Description		Status
JESSICA	Extreme Gradient Boosting , Random Forest	Performance Metrics (F1-Score)	Completed
ALI	Logistic Regression, Decision Tree,	Data extraction	Completed

Thank You!

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon** and infographics & images by **Freepik**

References

- https://www.drivendata.org/competitions/57/nepal-earthquake/
- https://www.analyticssteps.com/blogs/how-does-linear-and-logistic-regression-work-machine-learning
- https://dataaspirant.com/catboost-algorithm/#t-1609567161998
- https://analyticsindiamag.com/7-types-classification-algorithms/
- https://dataaspirant.com/catboost-algorithm/#t-1609567161998
- https://medium.com/analytics-vidhya/richters-predictor-modeling-earthquake-damage-b44e3dbdaef