Cairo University
Faculty of Engineering
Computer Department
4th Year Semester

Big Data Final Project

Up Votes Predictor

Reham Ali Mohamed
Amr Aboshama Ali
Mohamed Ramzy
Mai Mostafa Ibrahem

Problem Description:

Predict the number of upvotes on a question in a given forum or platform.

Business Value of this idea:

- content platforms have a constant need to identify the best content in time to appropriately promote and thereby improve the engagement at the website.
- It can also help to analyze which ads put on a certain question would reach more users.
- It will help us identify the best question authors in a short amount of time.

Project Pipeline:

- Brainstorming for ideas.
- Searching for datasets.
- Exploring the dataset.
- Visualize data.
- Preprocessing on data.
- Training models.
- Evaluating models.

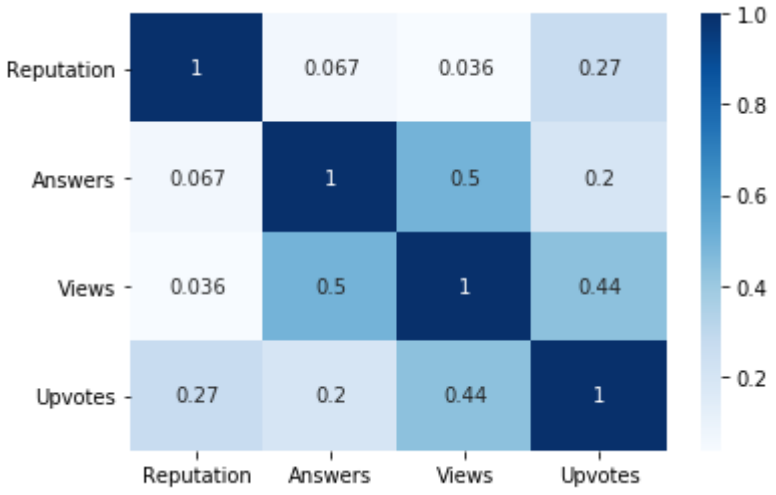
Analysis and Solution:

Data Preprocessing:

1. Removing unuseful columns from our dataset (ID & Username)
2. Converting the categorical variable ‘Tag’ to a numerical one so we can deal with it better.
3. Split data set into training data and validation data.
4. Perform standardization and scaling to the features’ values.
5. Generate polynomials and interactions between features.

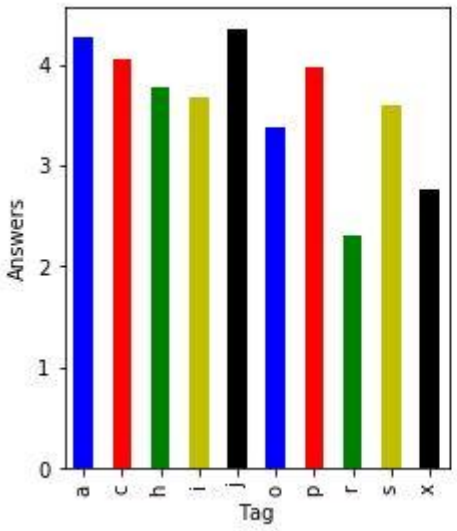
Data Visualization:

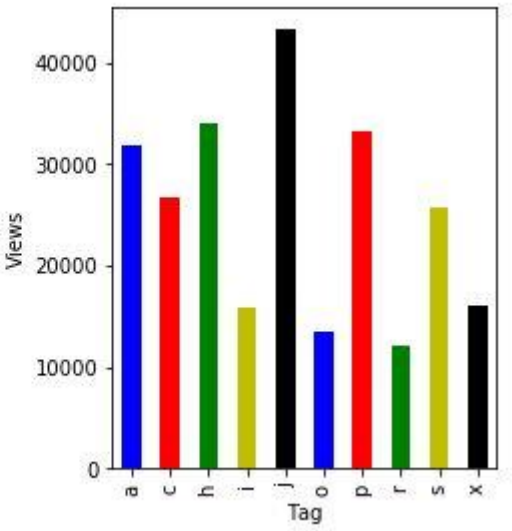
1. Correlation between features.



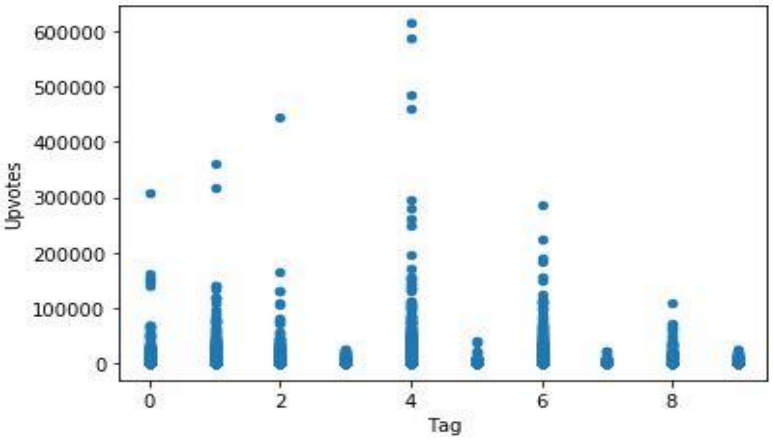
2. Relation between Tags with Answers & Tags with views

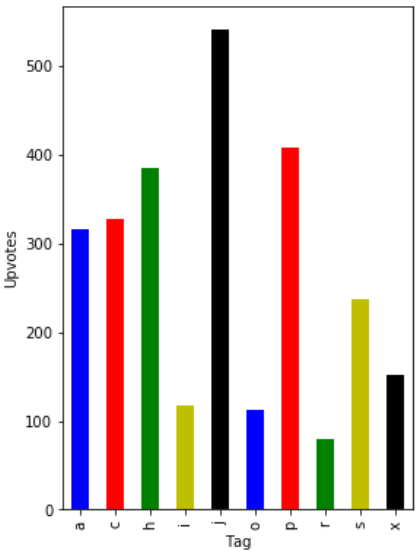
   Conclusion: Some Tags are more active than others, They receive more interaction whether its answers or views.





3. The relation between ‘Tag’ and ‘upvotes’





Insights from Data:

- No nulls or missing values.
- All features are numeric except “Tag” is categorial.
- Correlation between ‘Upvotes’ and ‘views’ is the highest correlation which indicates that ‘views’ affects the prediction.
- Some ‘Tag’ categories are more likely to be upvoted than others.

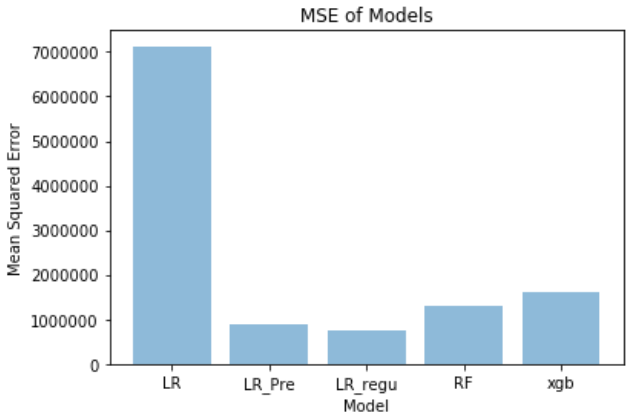
Model Training:

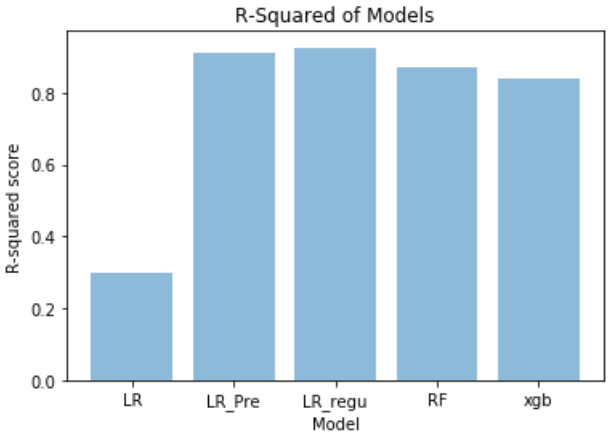
We tried using regression with three different models:

1. Linear Regression:
    a. Without Preprocessing on data.
    b. With standardization, scaling data and generating polynomials and interactions between features.
    c. With part b and regularization to decrease mean square error, avoid overfitting and reduce model complexity.
2. Random Forest Regressor.
3. XGBoost.

Results and Evaluation:

Model	MSE	R-Square
Linear Regression	7122238.31	0.300363
LR with preprocessing	884975.43	0.913066
LR with Pre & regularization	754112.53	0.925921
Random Forest Regression	1297485.36	0.872544
xgBoost Regression	1607732.73	0.842068





Conclusion:

- In our case, converting the categorical feature into many boolean features wasn't useful because it resulted in 9 new boolean features while the original features are only 4!
- It's better to convert the categorical feature into numerical codes instead of converting in many boolean features as it takes less time in training and processing and also results in better performance.

Unsuccessful Trials:

- Some trials in tuning the parameter ‘alpha’ while performing regularization on linear regression model till find **best** value: **alpha = 0.096**
- Trial to convert the categorical feature ‘Tag’ into n-1 of boolean features but it gave worse results and took more time in processing than encoding it into numerical values.

- We tried to train **xgBoost** and **Random Forest** on the data after standardization, scaling and generating polynomials but it took **too much** time and results were a little bit worse than before preprocessing.

Enhancements and future work:

- Better Tuning for parameters in Random Forest and xgBoost to achieve better performance.
- Trying other models**.**