Team: Data Freaks

Team Members: Ravi Teja Reddy Dodda, Harika Satti, Tanuja Reddy Maligireddy

Project Name: **Priv-Rank for Social Media**

Data Science Approaches and Algorithms:

1. **Computations:** For Priv-Rank and its analysis, we have used libraries of machine learning.

   1.1 At first, we have used Fast API library to compute performance of flask web frameworks which is helpful for asynchronous code for declaring endpoints.

   1.2 Then we have used random library to compute random similiarity matrix for preprocessed priv-rank dataset

2. **Prediction Model:** To build the classification model, we have used several approaches and algorithms.

   2.1 Multiclass Classification Algorithms: First of all, we have used several machine learning algorithms to build classification/prediction models. Following are the used algorithms with their accuracy.

No.	Algorithms	Accuracy
1	Logistic Regression	0.893236
2	Stochastic Gradient Decent	0.881183
3	Random Forest	0.872281
4	CatBoost	0.869907
5	Naive Bayes	0.769763
6	XGBoost	0.724382

2.2 Deep Learning Approach (Transfer learning in fastai):

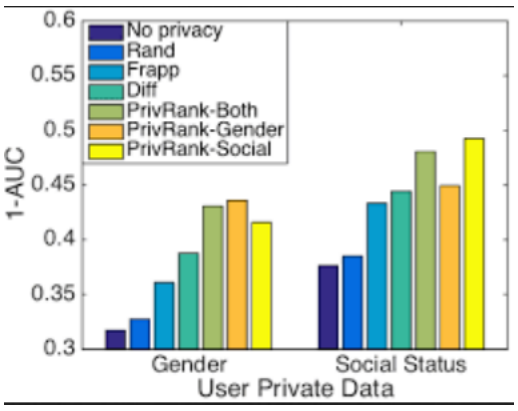
We have two kaggle datasets. Firstly, we normalize the data using Standard Scalar. Then used the PCA (Principle Component Analysis) for dimensionality reduction from 3 to 1. And then we created the similarity matrix or we can say correlation matrix... Our intention is to explore the advanced side of data science. We have used the fastAI library here. Following are the steps taken:

a. We currently have a dataset of users containing 4 things. UserID, gender, age and posts they liked.

b. Firstly, we normalize the data using Standard Scalar. Then used the PCA (Principle Component Analysis) for dimensionality reduction from 3 to 1. And then we created the similarity matrix or we can say correlation matrix.

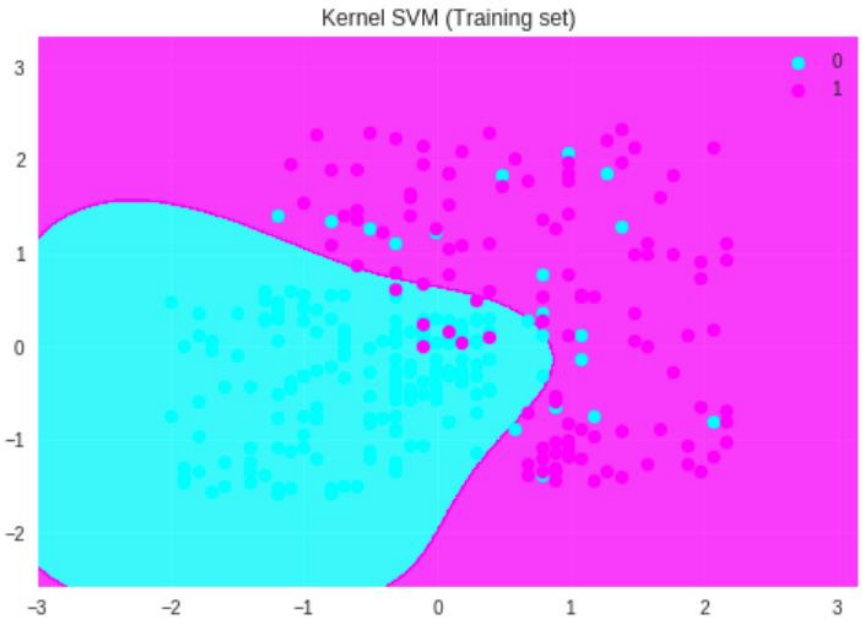
c. Finally, we obfuscate data 10 - 20% like converting the likes to dislikes and vice-versa. So, we could not easily backtrack to the user from their choices. This process was done on the user's end.

Features Used



The main features we have explored and worked on are:

1. Profile based leaks
2. Person prediction
3. Gender based leaks



Features Derived:

1. User Activity
2. Applications access
3. Polarity
4. Priv score

Profile based Leaks

From our labelled dataset, we obtain the users meta data and profile activity.

Person Prediction

We actually divide all the individual and hashtags activities to obtain more in-depth results.

Gender based Leaks

Generally, people had different tastes and different opinions on topics in social media. So a gender based activity narrows a little to find which activities cause more damage to data leakage and of what gender.

User Activity

Now that we have users that are most similar to us, we need to find the likeability of each post not seen by our current user by using a customized formula.

Applications Access

Different applications access your social media that causes privacy issues.

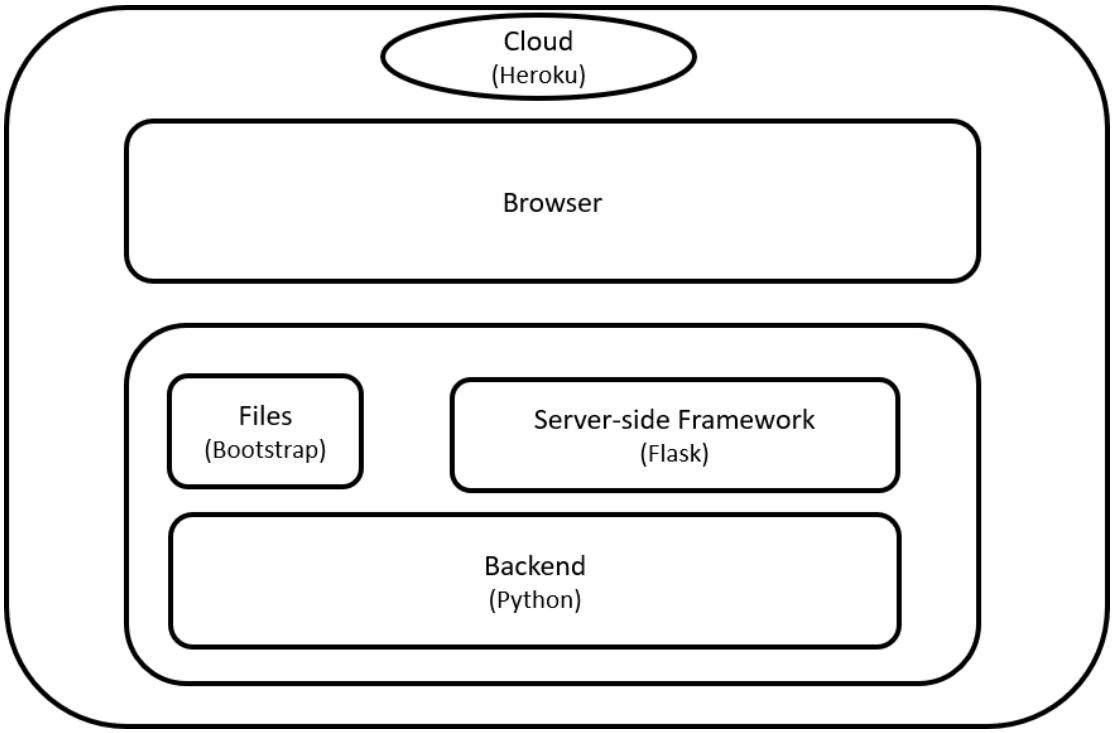
Polarity

We obtain the polarity of the profile for identifying if the profile is at risk or not.

Priv Score

We derive the priv score from the data itself to analyze the inclination of the profile.

Client Side Design



Cloud

Application deployed into the cloud is the shell of entire client centric development which makes it as a seamless interaction between user and application

Browser

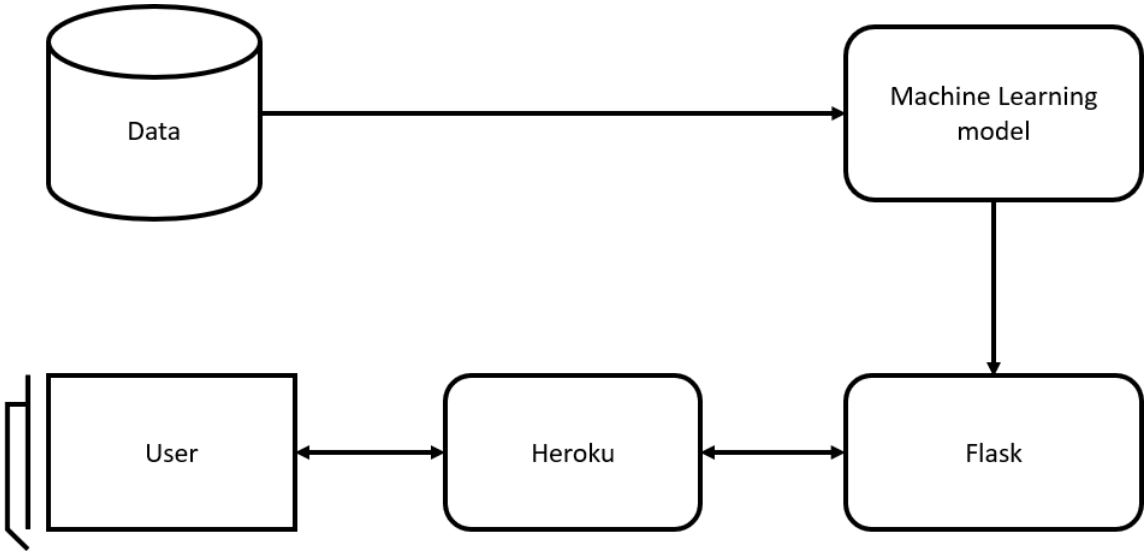
Most common and feasible component for a web application and Edge, Opera, Firefox, Chrome, Safari, Safari mac, Firefox mac, Chrome mac are the major browsers while writing this document. As you can see, trying to build and test everything is difficult. Each browser has its own subtle nuances different in browser security, default font sizes, borders etc. All these issues can be overcome with the changes required in the programming.

Programming

Programming with required components whether it can be frontend or backend captivates the user's experience.

To make it a more user friendly and interactive interface it always moves in a frontend's direction. Frontend is a key that always captures customer's satisfaction to yield more income. Backend is always a hidden gem that performs it's actions to strengthen the application and makes it more efficient.

Model Deployment

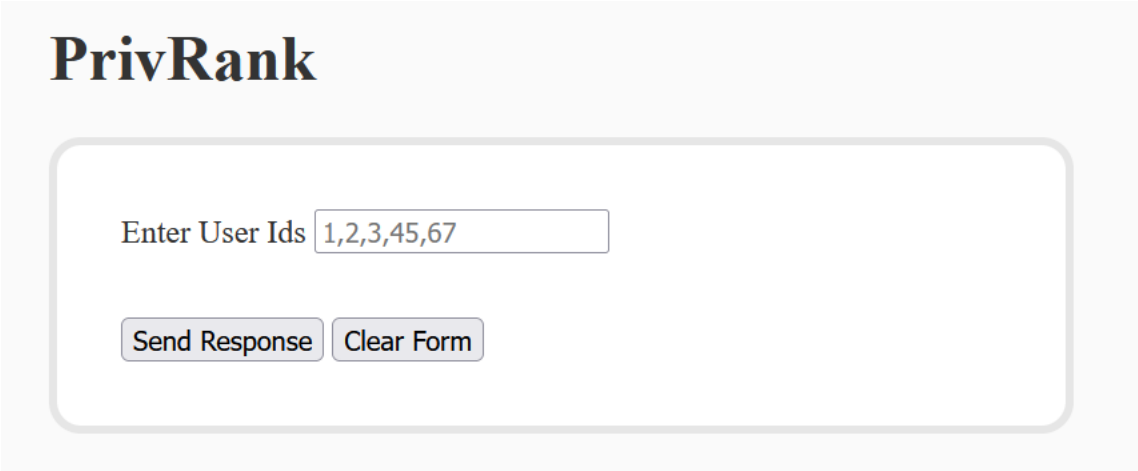


Fetching Training data ----------> training model ---------- > Evaluating model ---------- > Model Endpoint

Fetching and training data from the trusted sources and then building a model to attain a classifier and using that pickle file/ providing an interface by making it a supervised approach takes it further by gathering inputs from users and to make a prediction with an accuracy of 88.5% as high as possible for a multiclass classifier in this spectrum looks an efficient build. Deploying the same into the cloud by compressing slug size without any compromises is an added advantage. We have used this model to make COVID-19 vaccine related tweet sentiment prediction.

Client-Side Application: Web App

To make Interface more interactive, scalable and enjoyable we have picked Flask and Bootstrap as our designers. As Flask is a micro-framework i.e with little to no dependencies to external libraries is the reason we picked it over Django and it is light, there are little dependency to update and watch for security bugs



We have crafted it to be simple yet elegant to use. The straight forward interface which has response analyzer on the home screen and redirects to the result page with a simple click.