

PROJECT REPORT

GENETIC CLASSIFICATION OF AN INDIVIDUAL USING MACHINE LEARNING

TEAM CODE: 593068

TEAM MEMBERS:

**MOHAMMED UZAYR BOMBAYWALA
NITHILLEN JAYASEELAN
ATUL B PILLAI
PRANAV ANAND LEELARAM**

CAMPUS: VELLORE CAMPUS

COURSE: ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

TABLE OF CONTENTS:

1. INTRODUCTION
 2. LITERATURE SURVEY
 3. IDEATION AND PROPOSED SOLUTION
 4. REQUIREMENT ANALYSIS
 5. PROJECT DESIGN
 6. PROJECT PLANNING AND SCHEDULING
 7. CODING AND SOLUTIONING
 8. PERFORMANCE TESTING
 9. RESULTS
 10. ADVANTAGES AND DISADVANTAGES
 11. CONCLUSION
 12. FUTURE SCOPE
 13. APPENDIX
-

INTRODUCTION

The aim of this project is to develop a machine learning model that can classify human genetic variants based on their clinical significance. The model will use data from ClinVar, a public resource that contains annotations about human genetic variants. ClinVar classifies variants on a categorical spectrum ranging from benign, likely benign, uncertain significance, likely pathogenic, and pathogenic. However, some variants have conflicting classifications from different laboratories, which can cause confusion and uncertainty for clinicians and researchers. Therefore, this project will try to resolve these conflicts and provide a consistent and reliable classification for each variant.

The purpose of this project is to:

- Explore the ClinVar data and understand the factors that contribute to conflicting classifications.
- Apply machine learning techniques to preprocess, analyze, and model the data.
- Evaluate the performance and accuracy of the model using appropriate metrics and validation methods.
- Compare the model's predictions with the existing classifications and identify the variants that have the most disagreement or uncertainty.
- Provide insights and recommendations for improving the quality and consistency of ClinVar data and annotations.

LITERATURE SURVEY:

The existing problem is that conflicting classifications of genetic variants can lead to misinterpretation of the variant's impact on the disease of a given patient. This can result in incorrect diagnosis, treatment, and prognosis, which can have serious consequences for the patient's health and well-being.

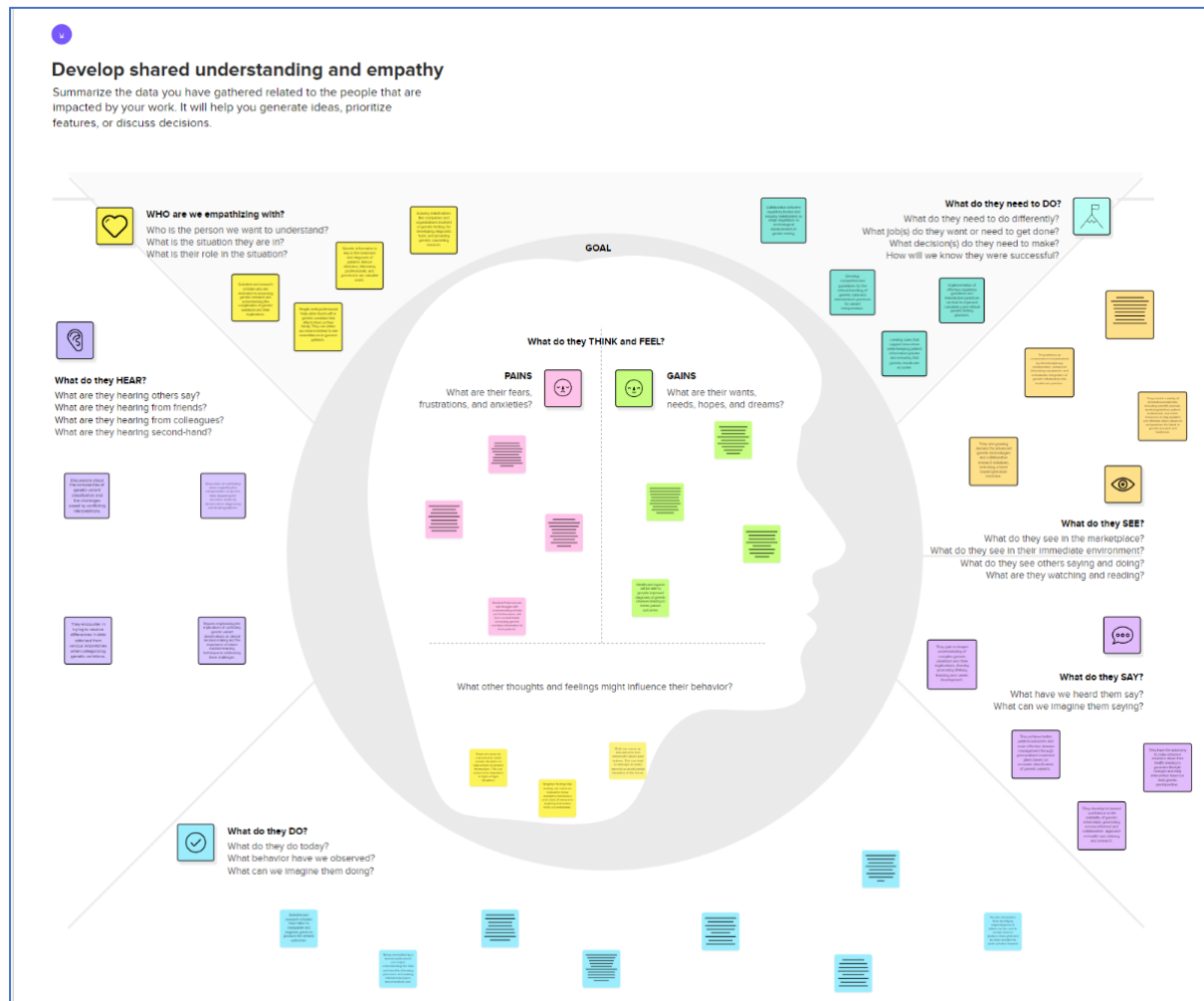
Here are some references that you might find useful:

- <https://www.ncbi.nlm.nih.gov/clinvar/docs/clinsig/>
- [GitHub - arvkevi/clinvar-kaggle: Scripts used to generate the ClinVar conflicting classifications dataset on Kaggle](#)

The problem statement is to develop a machine learning model that can resolve the conflicting classifications of genetic variants in ClinVar and provide a consistent and reliable classification for each variant. The model should be able to analyze the factors that contribute to conflicting classifications and use appropriate machine learning techniques to preprocess, analyze, and model the data. The model's performance and accuracy should be evaluated using appropriate metrics and validation methods. The model's predictions should be compared with the existing classifications and identify the variants that have the most disagreement or uncertainty. Finally, the project should provide insights and recommendations for improving the quality and consistency of ClinVar data and annotations.

IDEATION AND PROPOSED SOLUTION:

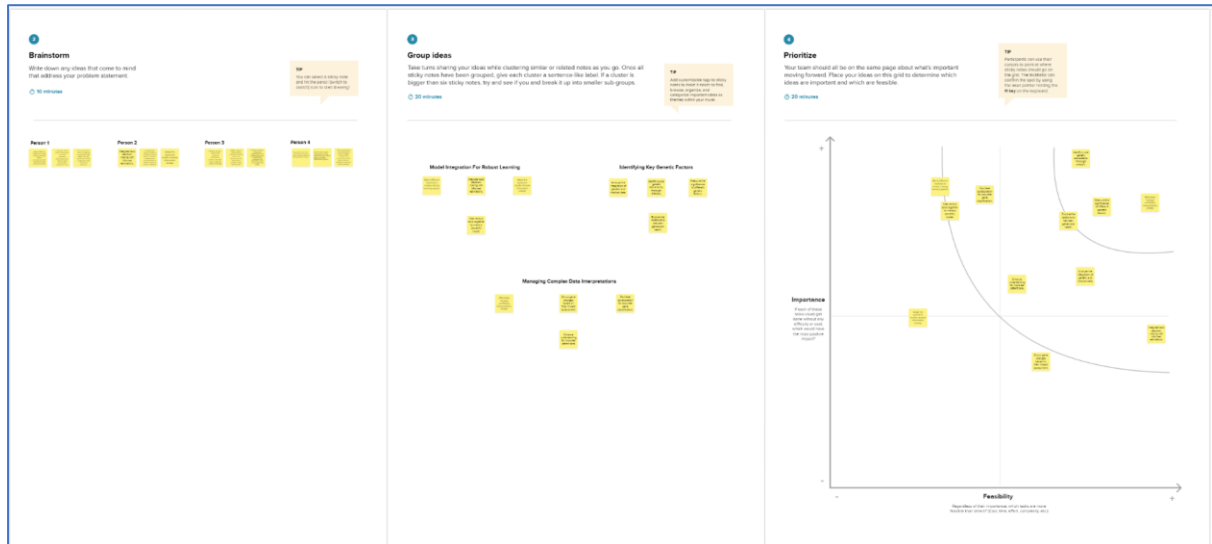
EMPATHY MAP:



MURAL LINK :

<https://app.mural.co/t/humangenomeproject8710/m/humangenomeproject8710/1697450124365/488957b6e38133e5fdb27e2b2be20d735a46a57d?sender=ud8b36b7ede0a72e96f054578>

BRAINSTORMING MAP:



MURAL LINK :

<https://app.mural.co/t/humangenomeproject8710/m/humangenomeproject8710/1697561135615/d235f7d3293000661c606f740f8d642cbb9f2550?sender=ud8b36b7ede0a72e96f054578>

REQUIREMENT ANALYSIS:

Functional Requirements:

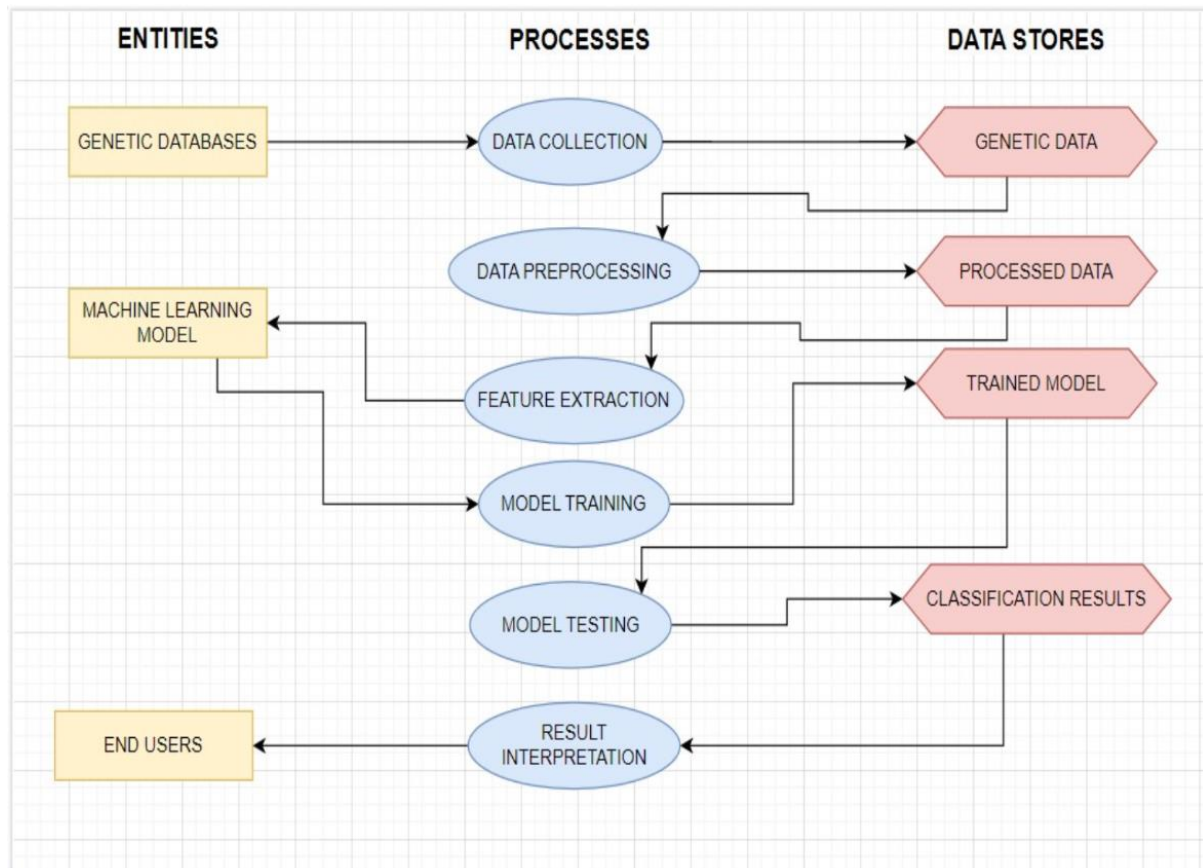
- **Pre processing/cleaning of dataset**
- **Feature Extraction, Model training and Testing**
- **Interpreting the final result.**

Non – functional Requirements:

- **.csv file of raw dataset. This is to be converted into a cleaned .csv file that will be used later.**
 - **Google colab – Pre processing and model training and testing using various classifiers.**
 - **VS Code – html and flask source code for web application.**
-

PROJECT DESIGN:

Data Flow Diagram:

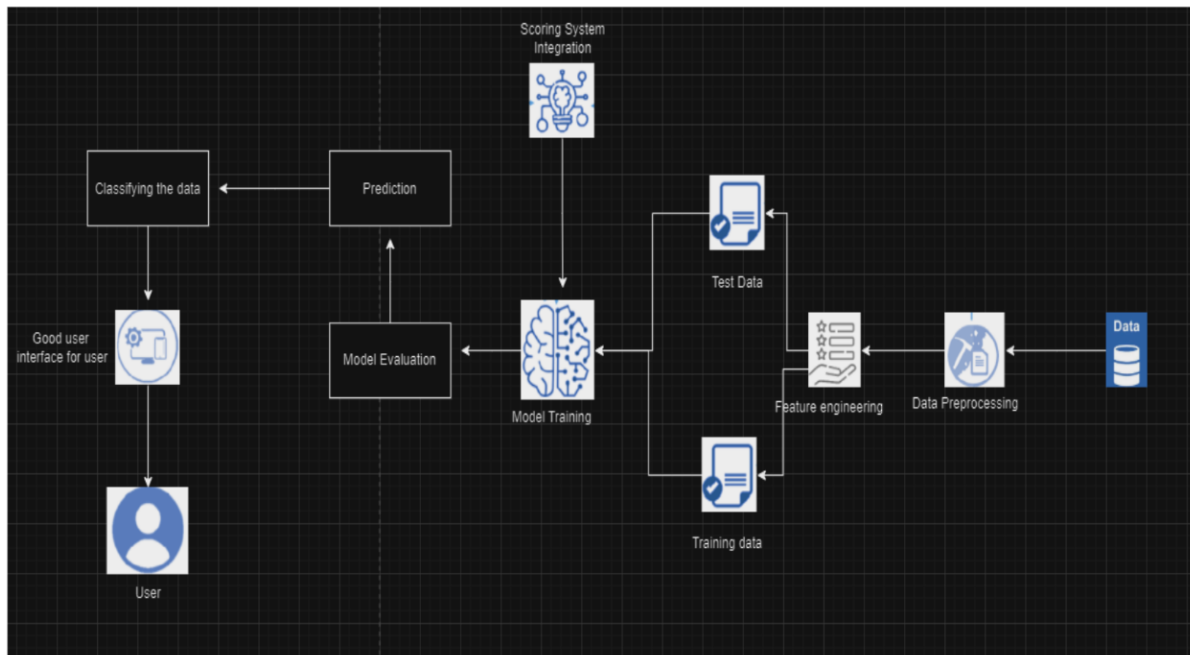


User stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story/Task	Acceptance Criteria	Priority	Release
Clinical Laboratories	Setup & Infrastructure	USN-1	Configure the development environment with necessary libraries and frameworks for genetic variant classification.	Development environment is successfully set up with all required tools and frameworks.	High	Sprint 1
Genetic Researchers	Data Collection	USN-2	Gather a diverse dataset of genetic variants, including associated clinical data, for training and testing the machine learning models.	A comprehensive dataset comprising a range of genetic variants and relevant clinical data is obtained.	High	Sprint 1
General Public	User Interface Development	USN-3	Design an intuitive and user-friendly interface that allows the general public to access basic information about genetic variants and their potential health implications.	The user interface is visually appealing, easy to navigate, and provides understandable information about genetic variants.	Medium	Sprint 2
Industry Stakeholders	Product Integration	USN-4	Integrate the genetic variant classification tool into existing	The tool is successfully integrated, and it seamlessly	Medium	Sprint 3

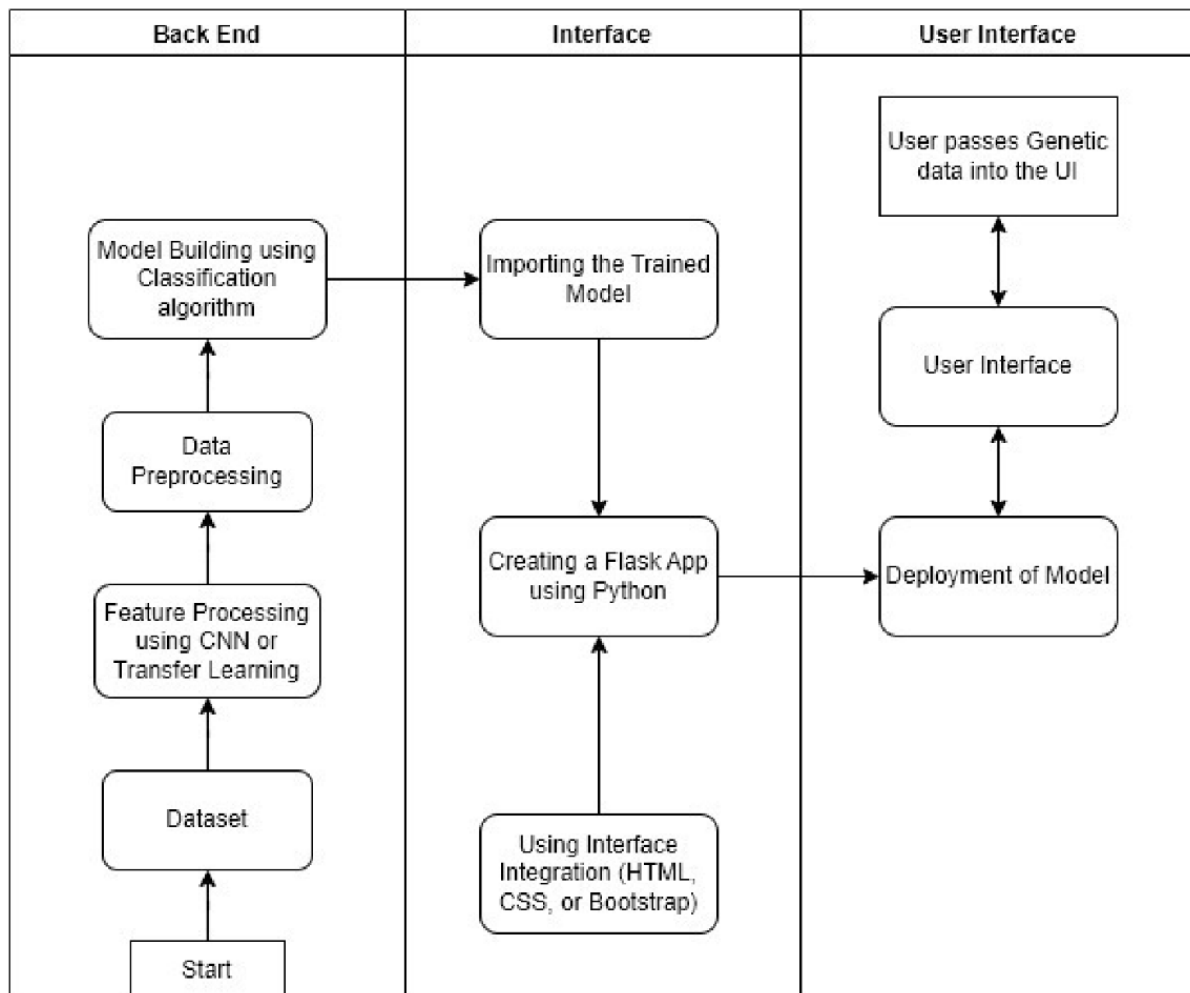
			industry systems, ensuring compatibility and seamless operation.	interacts with the industry's existing infrastructure.		
--	--	--	--	--	--	--

Solution Architecture:



PROJECT PLANNING AND SCHEDULING:

Technical Architecture:



Components and Technologies:

S. No	Component	Description	Technology
1.	User Interface	How user interacts with application	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, MongoDB
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	23andMe API, etc.
9.	External API-2	Purpose of External API used in the application	AncestryDNA API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Scikit-learn, TensorFlow, PyTorch
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes, etc.

CODING AND SOLUTIONING:

CLEANING DATASET:

The dataset is cleaned using various pre-processing techniques such as dropping, dichotomizing, encoding, splitting via recognized character, classification of minority elements as a singular class, and so on.

This is done so that it will be easier and possible to implement the required machine learning algorithms for the next phase.

TRAINING AND TESTING THE MODEL:

The data to be trained and tested can be decided by the programmer by means of the train test split method. Here we split the dataset into two parts.

Accuracy and precision for various classifiers are also done

```
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier,
AdaBoostClassifier, GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, precision_score,
recall_score, f1_score
accuracy_results.sort(key=lambda x: x[1], reverse=True)
precision_results.sort(key=lambda x: x[1], reverse=True)
recall_results.sort(key=lambda x: x[1], reverse=True)
f1_score_results.sort(key=lambda x: x[1], reverse=True)

labels, values = zip(*accuracy_results)
plt.figure(figsize=(12, 6))
plt.subplot(2, 2, 1)
plt.barh(labels, values, color='skyblue')
plt.xlabel('Accuracy')
plt.ylabel('Classifier')
plt.title('Accuracy Comparison of Different Classifiers')
```

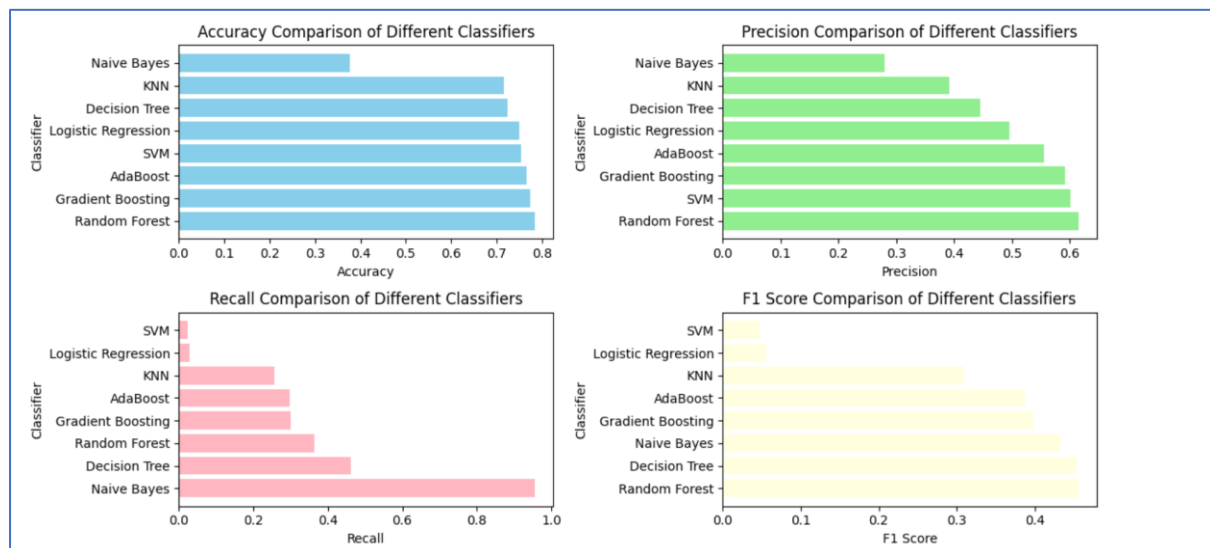
```

labels, values = zip(*precision_results)
plt.subplot(2, 2, 2)
plt.barh(labels, values, color='lightgreen')
plt.xlabel('Precision')
plt.ylabel('Classifier')
plt.title('Precision Comparison of Different Classifiers')

labels, values = zip(*recall_results)
plt.subplot(2, 2, 3)
plt.barh(labels, values, color='lightpink')
plt.xlabel('Recall')
plt.ylabel('Classifier')
plt.title('Recall Comparison of Different Classifiers')

labels, values = zip(*f1_score_results)
plt.subplot(2, 2, 4)
plt.barh(labels, values, color='lightyellow')
plt.xlabel('F1 Score')
plt.ylabel('Classifier')
plt.title('F1 Score Comparison of Different Classifiers')

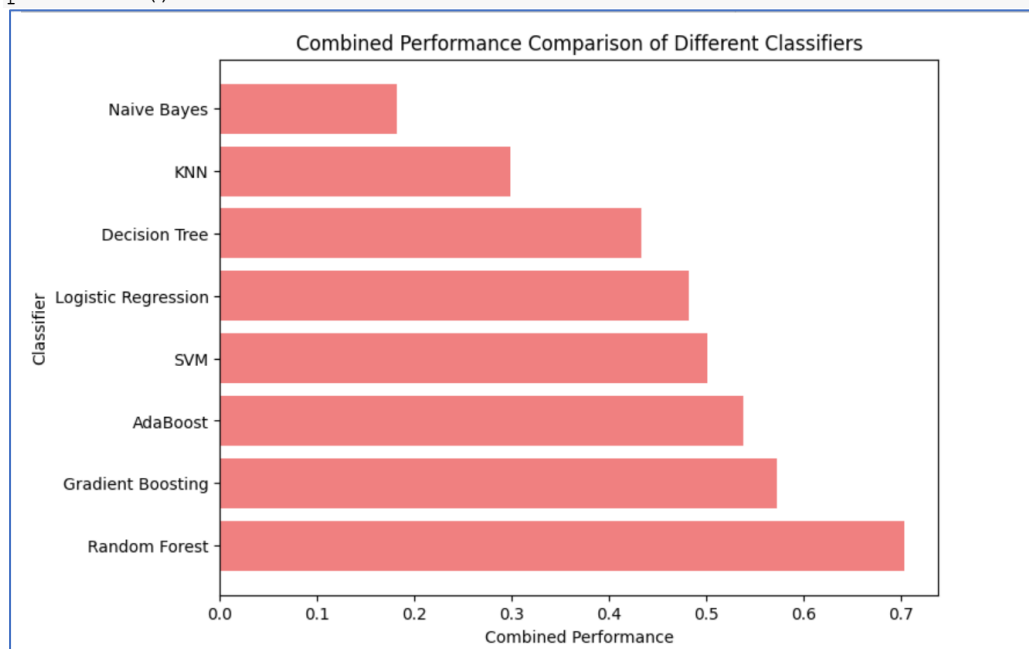
```



Then we compare the overall performance of each classifier to choose the ones that are satisfying our performance criteria.

```
combined_results = [(label, (accuracy + precision + recall + f1) / 4)
for (label, accuracy), (_, precision), (_, recall), (_, f1) in
zip(accuracy_results, precision_results, recall_results,
f1_score_results)]
combined_results.sort(key=lambda x: x[1], reverse=True)

labels, values = zip(*combined_results)
plt.figure(figsize=(8, 6))
plt.barh(labels, values, color='lightcoral')
plt.xlabel('Combined Performance')
plt.ylabel('Classifier')
plt.title('Combined Performance Comparison of Different Classifiers')
plt.show()
```



Here, we use 'randomforestclassifier' as it has the best performance.

We then build the model as follows:

```
rfc = RandomForestClassifier(n_estimators=100)
rfc.fit(X_train, y_train)
y_pred = rfc.predict(X_test)
print(y_pred)
```

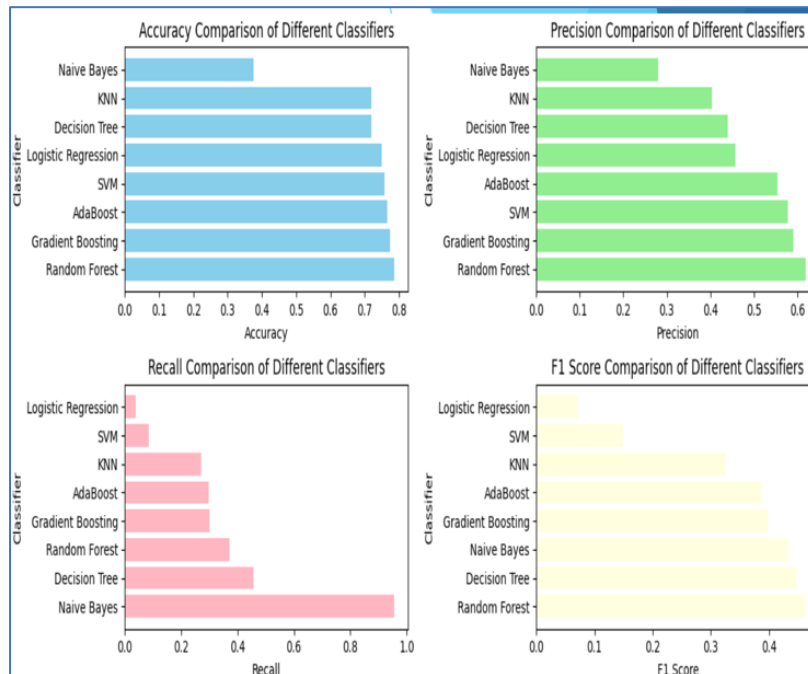
We then dump the model in a .pkl file and download the file.

For the web application, we make use of html for the frontend and python flask for the backend.

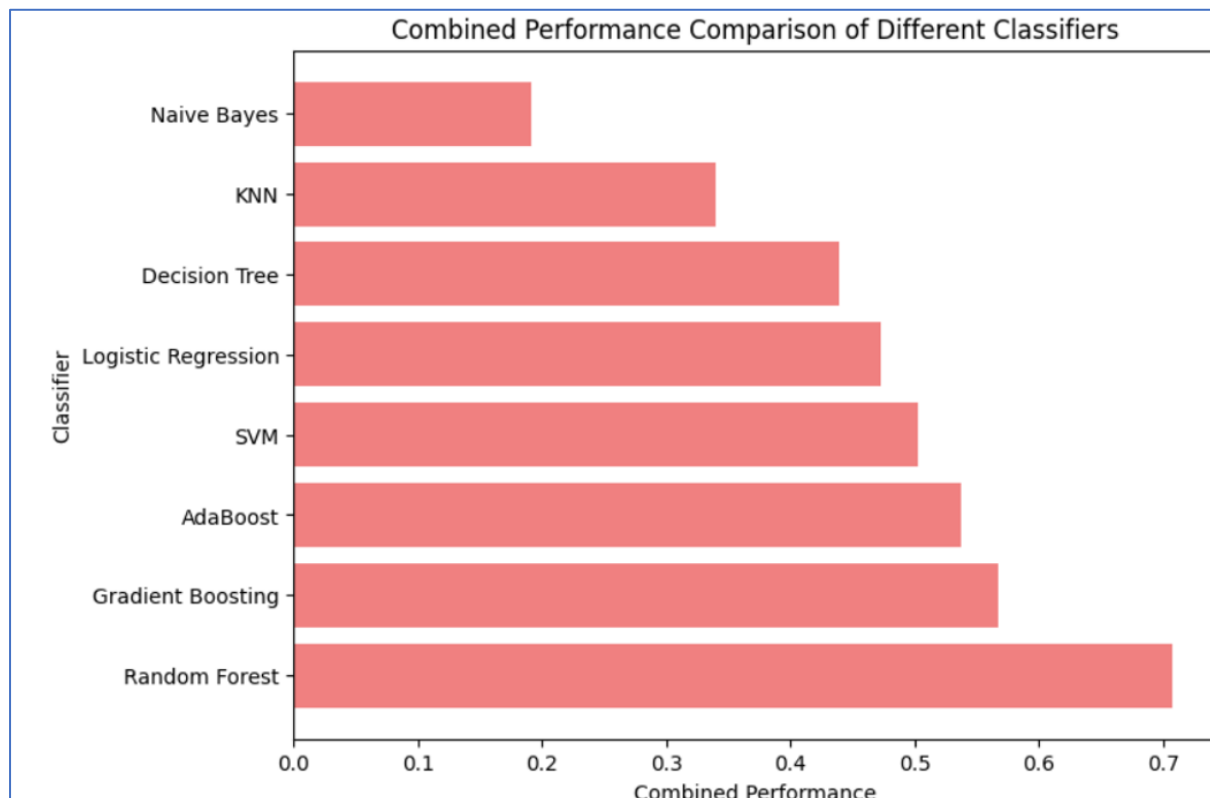
We make the user input all the required values corresponding to the uncleaned dataset, process the same, and then run our model to give us the result.

PERFORMANCE TESTING:

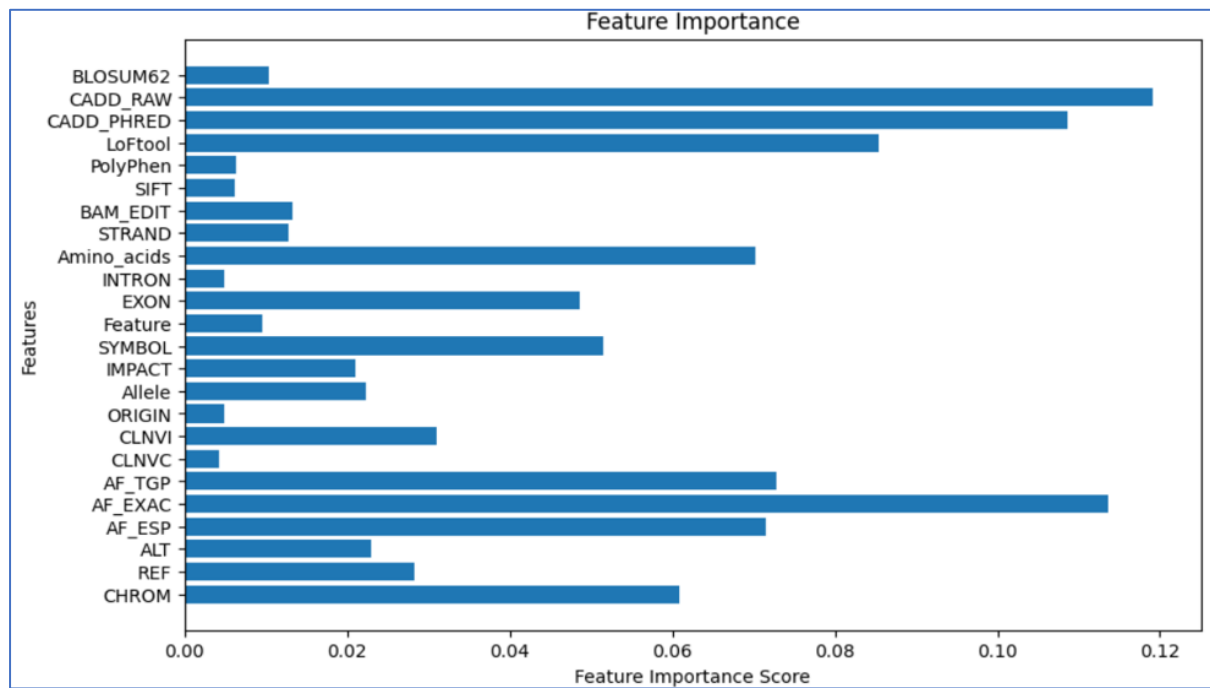
Accuracy comparison of classifiers:



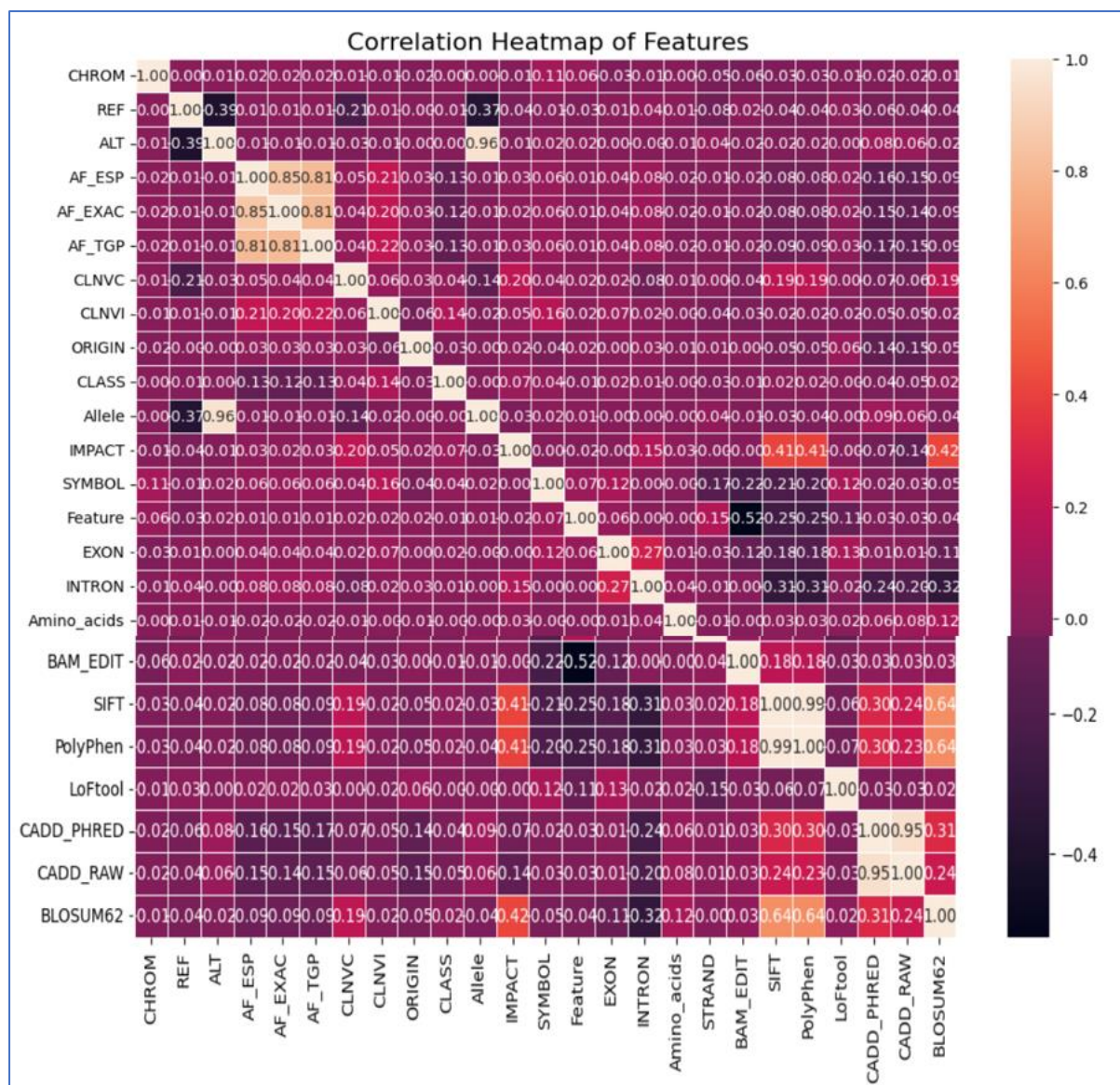
Performance comparison of classifiers:



Feature Importance:



Heatmap:

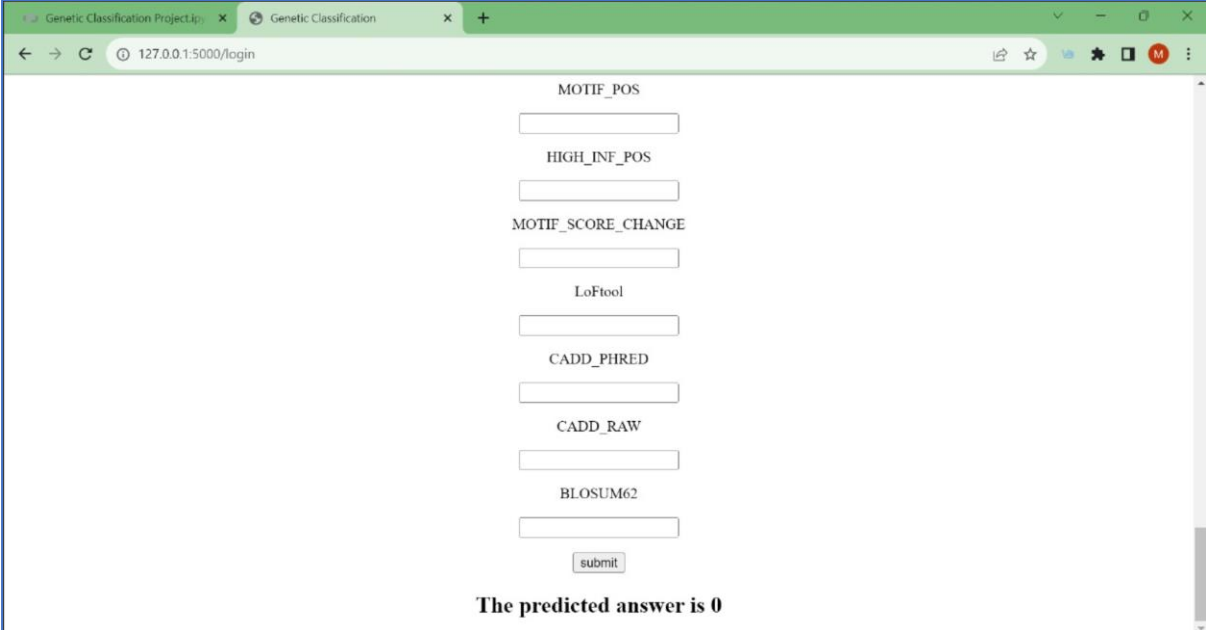


RESULT:

We get the output as either 0 or 1 where they represent the predicted value.

0 – not conflicting

1 – Conflicting



The screenshot shows a web browser window with two tabs: "Genetic Classification Project.ipynb" and "Genetic Classification". The address bar displays "127.0.0.1:5000/login". The main content area contains a vertical list of input fields, each with a label above it: "MOTIF_POS", "HIGH_INF_POS", "MOTIF_SCORE_CHANGE", "LoFtool", "CADD_PHRED", "CADD_RAW", and "BLOSUM62". Below these fields is a "submit" button. At the bottom of the form, the text "The predicted answer is 0" is displayed.

Field Label	Input Value
MOTIF_POS	
HIGH_INF_POS	
MOTIF_SCORE_CHANGE	
LoFtool	
CADD_PHRED	
CADD_RAW	
BLOSUM62	

submit

The predicted answer is 0

ADVANTAGES AND DISADVANTAGES:

Advantages:

- It can help identify the genetic basis of diseases and disorders, which can lead to better diagnosis, treatment, and prevention.
- It can help identify individuals who are at risk of developing certain diseases or disorders, which can lead to early intervention and prevention.
- It can help identify carriers of genetic mutations that can cause diseases or disorders, which can help prevent the transmission of these mutations to future generations.
- It can help identify genetic variations that are associated with drug response, which can lead to personalized medicine and better treatment outcomes.
- It can help identify genetic variations that are associated with physical traits, which can be useful in forensic investigations and ancestry testing.

Disadvantages:

- It can raise ethical, legal, and social issues related to privacy, discrimination, and stigmatization.
- It can lead to psychological distress and anxiety for individuals who receive unfavourable results or who are at risk of developing certain diseases or disorders.
- It can be expensive and time-consuming to perform genetic testing and analysis, which can limit its accessibility and affordability.
- It can produce results that are difficult to interpret or that have uncertain clinical significance, which can lead to confusion and uncertainty for clinicians and researchers.
- It can raise concerns about the accuracy, reliability, and reproducibility of genetic testing and analysis, which can affect the quality and validity of the results.

CONCLUSION AND FUTURE SCOPE:

To conclude, we have successfully managed to obtain a model that will tell us whether the genetic data is conflicting or not based on the categories they are suited to be classified into by the used one, 'randomforestclassifier'.

Here are some of the potential areas of growth and opportunity in genetic classification:

- **Precision medicine:** The ability to use genetic information to tailor medical treatments to an individual's unique genetic makeup, leading to more effective and personalized treatments.
 - **Genetic counseling:** The use of genetic information to provide guidance and support to individuals and families who are at risk of developing certain diseases or disorders.
 - **Pharmacogenomics:** The study of how genetic variations affect drug response, which can lead to personalized medicine and better treatment outcomes.
 - **Forensic genetics:** The use of genetic information to identify individuals and establish relationships in forensic investigations.
 - **Ancestry testing:** The use of genetic information to trace an individual's ancestry and genealogy.
-

APPENDIX:

Source code:

https://colab.research.google.com/drive/1QO1lmeoeKm1TaQq3VYX9b3k9WcIki_Eg?usp=chrome_ntp#scrollTo=7eTcVOkT8mR

Github Repo:

<https://github.com/smartinternz02/SI-GuidedProject-596722-1697547501>

Project demo Link:

<https://drive.google.com/drive/folders/14RTmJRzZ-XLYKHJ9XeCc2o0RTV60eX63>
