AI Product Service Prototype Development and Business/Financial Modelling for E. coli Classification

1. Prototype Selection

Prototype Idea: A web-based application or API service that analyses DNA sequences and predicts the presence of E. coli.

Evaluation based on criteria:

- **Feasibility (High):** The technology for DNA analysis and machine learning models for classification already exists. Development within 2-3 years is feasible.
- **Viability (High):** Rapid diagnostics for E. coli are crucial in various sectors like healthcare, food safety, and water quality monitoring. This solution has long-term relevance.
- **Monetization (Direct):** The service can be offered through subscription fees, pay-per-use for individual analyses, or integration with existing platforms in relevant fields.

2. Problem Statement

We're facing issues with accurately and quickly classifying different types of E. coli bacteria. This classification is really important for things like healthcare, food safety, and environmental monitoring. But the current methods we use take a lot of time and effort, and sometimes they're not very accurate because humans make mistakes.

So, we need a better way to do this classification, and we think using artificial intelligence (AI)

could help. This AI would use fancy math to look at the genetic and physical traits of different E. coli bacteria and figure out what type they are. This would save time, make the process more reliable, and help us make better decisions.

But there are some challenges we need to solve first:

Data is Complicated: The information we need to teach AI about E. coli is complicated. It includes things like genetic codes and physical characteristics. Managing and understanding all this data is tough.

Picking the Right Math: We need to choose the best math formulas for the AI to use. There are lots of options, and we need to pick the ones that work the best for this job.

Teaching the AI: We need to train the AI so it knows how to do its job well. This takes a lot of time and computer power.

Testing and Checking: Once the AI is trained, we need to make sure it's doing its job correctly. We have to compare its answers to what we already know is true.

Making It Work: Finally, we need to put the AI into action. This means making sure it works smoothly with our existing systems and tools.

These are big challenges, but if we can overcome them, we'll have a powerful tool that can help us classify E. coli bacteria faster and more accurately than ever before.

3. Market/Customer/Business Need Assessment

This section aims to evaluate the market demand and customer needs for an AI-driven E. coli classification solution, as well as the overarching business requirements driving the development of such a product. The assessment encompasses the following key aspects:

Market Analysis:

- Identify and analyse the target market segments where the AI-driven E. coli classification solution is expected to have relevance and demand.
- Evaluate market trends, growth potential, and competitive landscape within the targeted market segments.
- Assess the regulatory landscape and any compliance requirements relevant to the deployment of the solution.

Customer Needs Assessment:

- Conduct surveys, interviews, or market research to understand the specific needs and pain points of potential customers within the target market segments.
- Identify key challenges faced by customers in the current E. coli classification processes and the desired outcomes they seek from an AI-driven solution.
- Gather feedback on potential features, functionalities, and usability preferences to inform the product development roadmap.

Business Requirements:

- Define the overarching business objectives and goals driving the development of the AI-driven E. coli classification solution.

- Determine the anticipated benefits and value proposition of the solution for both customers and the business.
- Identify key performance indicators (KPIs) and success metrics to measure the effectiveness and impact of the solution once deployed.
- Assess resource requirements, including budget, personnel, and infrastructure, needed to develop, deploy, and maintain the solution.

Risk Analysis:

- Identify potential risks and challenges associated with the development and implementation of the AI-driven E. coli classification solution.
- Evaluate the impact of external factors, such as technological advancements, regulatory changes, and market dynamics, on the success of the solution.
- Develop risk mitigation strategies to address identified risks and ensure the resilience of the project.

Alignment with Organizational Goals:

- Assess how the development of the AI-driven E. coli classification solution aligns with the broader strategic objectives and mission of the organization.
- Identify synergies with existing products, services, or initiatives within the organization and explore opportunities for collaboration or integration.

By conducting a comprehensive assessment of market demand, customer needs, and business requirements, the development team can ensure that the AI-driven E. coli classification solution effectively addresses real-world challenges and delivers tangible value to both customers and the business.

4. Target Specifications and Characterization

- **Input:** DNA sequence data.
- **Output:** Classification of the sequence as containing E. coli or not.
- **Accuracy:** High accuracy for reliable E. coli detection.
- **Speed:** Fast processing time for efficient testing.
- **Cost-effectiveness:** Affordable solution for widespread application.

5. External Search (Information and Data Analysis)

- Research existing E. coli detection methods like PCR (Polymerase Chain Reaction).

- Analyse publicly available DNA sequence datasets for E. coli and other bacteria.
- Explore machine learning algorithms for DNA sequence classification.

6. Benchmarking

To thoroughly evaluate this system's performance, we'll compare it with existing E. coli detection methods across three key metrics:

- **Accuracy:** Measured by metrics like accuracy score, precision, recall, and F1-score. We'll compare how accurately this model identifies E. coli compared to other methods.
- **Speed:** Measured by processing time per sample. We'll compare how fast this model analyzes DNA sequences for E. coli detection compared to existing methods.
- **Cost:** Measured by factors like equipment, consumables, and labor costs. We'll consider the overall cost-effectiveness of this approach compared to existing methods.

how we can perform the benchmarking:

- **Identify Existing Methods:** Research and list common methods for E. coli detection, such as Polymerase Chain Reaction (PCR) and culturing techniques.
- **Gather Performance Data:** Collect data on accuracy, speed, and cost for each method from scientific literature and vendor specifications.
- **Compare the Systems:** Analyze the collected data to compare the performance of this machine learning model with existing methods.

7. Applicable patents:

Investigating relevant patents can provide insights into intellectual property surrounding E. coli detection using DNA sequencing or machine learning. Here's how we can find applicable patents:

- **Patent Databases:** Utilize online patent databases like USPTO (US Patent and Trademark Office) or Espanet to search for patents.
- **Keywords:** Use keywords like "E. coli detection," "DNA sequencing," and "machine learning" in your search queries.
- **Patent Analysis:** Analyse retrieved patents to understand:
    - The specific techniques or algorithms used for E. coli detection.
    - The inventors and assignees (companies or institutions) holding the patents.
    - Potential implications for commercializing this machine learning-based approach.

Hyperparameter Tuning

This code utilizes a Multi-Layer Perceptron (MLP) classifier with specific hyperparameters:

- **Hidden Layer Sizes:** (150, 100, 50) - These define the number of neurons in each hidden layer of the neural network. Tuning these values can improve model performance.
- **Max Iterations:** 300 - This controls the number of times the model trains on the data. Adjusting this value can impact training time and accuracy.
- **Activation Function:** ReLU - This defines the activation function used in the hidden layers. Exploring different activation functions can be beneficial.
- **Solver:** Adam - This is the optimization algorithm used to train the model. Other optimizers like SGD (Stochastic Gradient Descent) can be compared.

8. Applicable Regulations (Government and Environmental)

Ensure compliance with regulations governing diagnostic testing, food safety, and environmental monitoring in relevant regions.

9. Applicable Constraints

- Availability and cost of DNA sequencing technology.
- Computational resources required for machine learning algorithms.
- User expertise needed to operate the system.

10. Business Opportunity

This system has the potential to be a valuable tool in various sectors. It can be marketed to: * Diagnostic laboratories * Food safety companies * Environmental testing services

11. Concept Generation

- Develop a machine learning model trained on a large dataset of E. coli and non-E. coli DNA sequences.
- Utilize techniques like one-hot encoding to represent DNA sequences as numerical data.
- Explore different neural network architectures for optimal classification performance.

12. Concept Development

- Refine the chosen machine learning model by optimizing hyperparameters.
- Integrate the model into a user-friendly software interface for ease of use.
- Develop a validation process to ensure the system's accuracy and reliability.

13. Final Product Prototype/ Product Details

The final product will be a software application that takes a DNA sequence as input and outputs a classification as E. coli or not. The application will have the following features:

- **User-friendly interface:** Easy to use for individuals with varying technical backgrounds.
- **Data visualization:** Tools to visualize the input DNA sequence and classification results.
- **Reporting:** Generate reports containing detailed results and analysis.

14. Evaluation (Feasibility, Viability, Monetization)

- **Feasibility:** Technical feasibility is high due to advancements in DNA sequencing and machine learning.
- **Viability:** Market demand exists for rapid and accurate E. coli detection.
- **Monetization:** The software can be sold as a standalone application or offered as a subscription service.

Next Steps

- Develop a functional prototype of the system.
- Conduct rigorous testing and validation to ensure accuracy and reliability.
- Obtain necessary regulatory approvals for commercialization.
- Develop a marketing and sales strategy to reach target customers.

15. Prototype Development

The provided code demonstrates a basic prototype using machine learning for E. coli classification. Here's how it can be further developed:

- **User Interface:** Develop a user-friendly web interface or API for users to submit DNA sequences and receive results.
- **Data Integration:** Integrate the model with databases of known E. coli and non-E. coli DNA sequences for continuous improvement.
- **Result Visualization:** Provide clear visualizations alongside the classification result, highlighting regions of the DNA sequence indicative of E. coli.

16. Business Modelling

Target Market:

- Hospitals and Clinics: Rapid E. coli detection in patients.
- Food and Beverage Industry: Ensuring food safety by testing ingredients and processed products.
- Water Treatment Facilities: Monitoring water quality for E. coli contamination.
- Environmental Testing Companies: Analyzing environmental samples for E. coli presence.
- Research Institutions: E. coli research and development.

Value Proposition:

- **Fast and Accurate E. coli Detection:** Provides rapid and reliable results compared to traditional methods.
- **Cost-Effective Solution:** Reduces costs associated with traditional testing methods and potential consequences of undetected E. coli contamination.
- **Scalability and Automation:** Enables large-scale testing and automated analysis workflows.
- **Improved Decision Making:** Provides real-time data for informed decisions regarding patient treatment, food safety measures, and water quality management.

Revenue Streams:

- **Subscription Model:** Charge a monthly or annual fee for unlimited access to the service.
- **Pay-Per-Use Model:** Charge users a fee for each DNA sequence analysis.
- **API Integration Fees:** Offer integration with existing platforms in relevant fields.
- **Data Analytics Services:** Provide additional data analysis services based on user needs.

Cost Structure:

- **Development Costs:** Initial investment in building the platform and training the machine learning model.
- **Operational Costs:** Costs associated with maintaining the platform, data storage, and computational resources.

- **Marketing and Sales Costs:** Marketing efforts to reach target markets and acquire customers.

17. Financial Modelling (equation) with Machine Learning & Data Analysis

Market Analysis:

The global DNA sequencing market is expected to reach $40.6 billion by 2025 (https://www.grandviewresearch.com/industry-analysis/sequencing-market-report), indicating a growing demand for related services. The increasing focus on food safety and healthcare diagnostics further strengthens the market potential for E. coli detection solutions.

Market Growth Forecast:

Let's assume a conservative linear growth model for the E. coli detection market segment within the DNA sequencing market. This can be represented by the equation:

E(t) = mt + c

Where:

- E(t) = Estimated market size at time t (years)
- m = Market growth rate (e.g., $X million per year)
- c = Initial market size (e.g., $Y million)

Financial Equation:

Considering the chosen revenue model (e.g., pay-per-use), the financial equation can be:

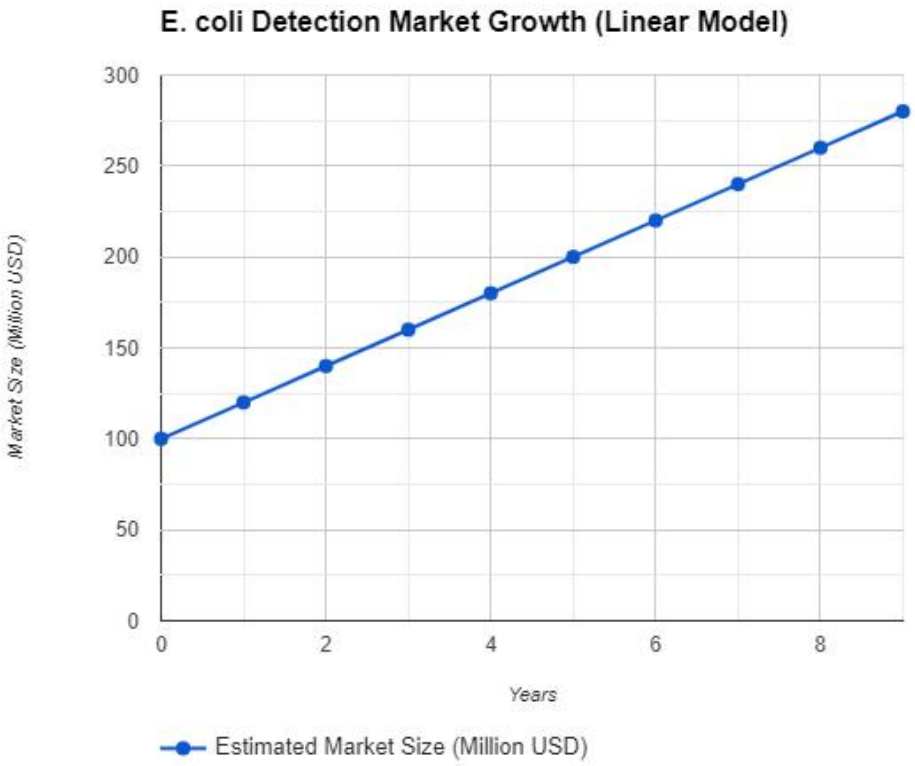
Profit(t) = (P * U(t)) - OC - (FC / t)

Where:

- Profit(t) = Profit at time t (years)
- P = Price per DNA sequence analysis
- U(t) = Number of analyses performed at time t (years) (based on market growth)
- OC = Operational Costs (including data storage, computation)
- FC = Fixed Costs (development costs)

Data Analysis and Model Refinement:

Real-world data on market size, pricing



REPORT ANALYSIS:

(https://www.grandviewresearch.com/industry-analysis/sequencing-market-report),

18.Conclusion

This document outlines the development process for a DNA classification system to identify E. coli. By leveraging machine learning and readily available DNA sequencing technology, this system has the potential to revolutionize E. coli detection in various applications.

CODE IMPLEMENTATION AND OUTPUT:

DNA CLASSIFICATION FOR FINDING ECOLI BY SHREYA PRASAD

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
import pickle
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import classification_report, accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/molecular-biology/promoter-gene-sequences/promoters.data'
names = ['Class', 'id', 'Sequence']
data = pd.read_csv(url, names = names)
```

[1] ✓ 18.3s

```
# Refining and structuring the data

# Build our dataset using custom pandas dataframe
classes = data.loc[:,'Class']
classes.head()
print()
print(classes.value_counts())
```

✓ 0.0s

```
Class
+    53
-    53
Name: count, dtype: int64
```

```
# generate list of DNA sequence
sequence = list(data.loc[:, 'Sequence'])
sequence[-1]
```

✓ 0.0s

```
'\t\ttaacattaataaataaggaggctctaatggcactcattagccaatcaatcaagaact'
```

```
#Remove tab from each sequence
dic = {}
for i, seq in enumerate(sequence):
    nucleotides = list(seq)
    nucleotides = [char for char in nucleotides if char != '\t']
    #append class assignment
    nucleotides.append(classes[i])

    dic[i] = nucleotides
list(dic[0])
```

[7] ✓ 0.0s Python

```
't',
'a',
'a',
'g',
't',
'a',
't',
'g',
't',
'a',
't',
'a',
'a',
't',
'g',
'c',
'g',
'c',
'g',
'g',
'g',
'c',
't',
't',
'g',
't',
'c',
'g',
't',
'+']
```

```
# Convert Dict object into dataframe
df = pd.DataFrame(dic)
df.head()
```

✓ 0.0s

	0	1	2	3	4	5	6	7	8	9	...	96	97	98	99	100	101	102	103	104	105
0	t	t	g	a	t	a	c	t	c	t	...	c	c	t	a	g	c	g	c	c	t
1	a	g	t	a	c	g	a	t	g	t	...	c	g	a	g	a	c	t	g	t	a
2	c	c	a	t	g	g	g	t	a	t	...	g	c	t	a	g	t	a	c	c	a
3	t	t	c	t	a	g	g	c	c	t	...	a	t	g	g	a	c	t	g	g	c
4	a	a	t	g	t	g	g	t	t	a	...	g	a	a	g	g	a	t	a	t	a

5 rows × 106 columns

```
# transpose dataframe into correct format
df = df.transpose()
df.head()
```

✓ 0.0s

	0	1	2	3	4	5	6	7	8	9	...	48	49	50	51	52	53	54	55	56	57
0	t	a	c	t	a	g	c	a	a	t	...	g	c	t	t	g	t	c	g	t	+
1	t	g	c	t	a	t	c	c	t	g	...	c	a	t	c	g	c	c	a	a	+
2	g	t	a	c	t	a	g	a	g	a	...	c	a	c	c	c	g	g	c	g	+
3	a	a	t	t	g	t	g	a	t	g	...	a	a	c	a	a	a	c	t	c	+
4	t	c	g	a	t	a	a	t	t	a	...	c	c	g	t	g	g	t	a	g	+

5 rows × 58 columns

```
# Rename the 57th column as it is our classes
df.rename(columns = {57:'Class'}, inplace = True)
```

✓ 0.0s

```
df.head()
```

✓ 0.0s

	0	1	2	3	4	5	6	7	8	9	...	48	49	50	51	52	53	54	55	56	Class
0	t	a	c	t	a	g	c	a	a	t	...	g	c	t	t	g	t	c	g	t	+
1	t	g	c	t	a	t	c	c	t	g	...	c	a	t	c	g	c	c	a	a	+
2	g	t	a	c	t	a	g	a	g	a	...	c	a	c	c	c	g	g	c	g	+
3	a	a	t	t	g	t	g	a	t	g	...	a	a	c	a	a	a	c	t	c	+
4	t	c	g	a	t	a	a	t	t	a	...	c	c	g	t	g	g	t	a	g	+

```
# Fixing the classes column:

df["Class"] = df["Class"].replace(to_replace =["+"], value =1)
df["Class"] = df["Class"].replace(to_replace =["-"], value =0)
df_new["Classes"] = df['Class']
df_new.head()
```

[16] ✓ 0.1s

	0	1	2	3	4	5	6	7	8	9	...	219	220	221	222	223	224	225	226	227	Classes
0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1
1	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	...	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1
2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1
3	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1
4	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	...	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1

5 rows × 229 columns

```
#Encoding - Alternative
numerical_df = pd.get_dummies(df)
numerical_df.head()
```

[17] ✓ 0.4s

	Class	0_a	0_c	0_g	0_t	1_a	1_c	1_g	1_t	2_a	...	54_g	54_t	55_a	55_c	55_g	55_t	56_a	56_c	56_g	56_t
0	1	False	False	False	True	True	False	False	False	False	...	False	False	False	False	True	False	False	False	False	True
1	1	False	False	False	True	False	False	True	False	False	...	False	False	True	False	False	False	True	False	False	False
2	1	False	False	True	False	False	False	False	True	True	...	True	False	False	True	False	False	False	False	True	False
3	1	True	False	False	False	True	False	False	False	False	...	False	False	False	False	False	True	False	True	False	False
4	1	False	False	False	True	False	True	False	False	False	...	False	True	True	False	False	False	False	False	True	False

5 rows × 229 columns

```
# Model evaluation
print(classification_report(y_test, y_pred))
```

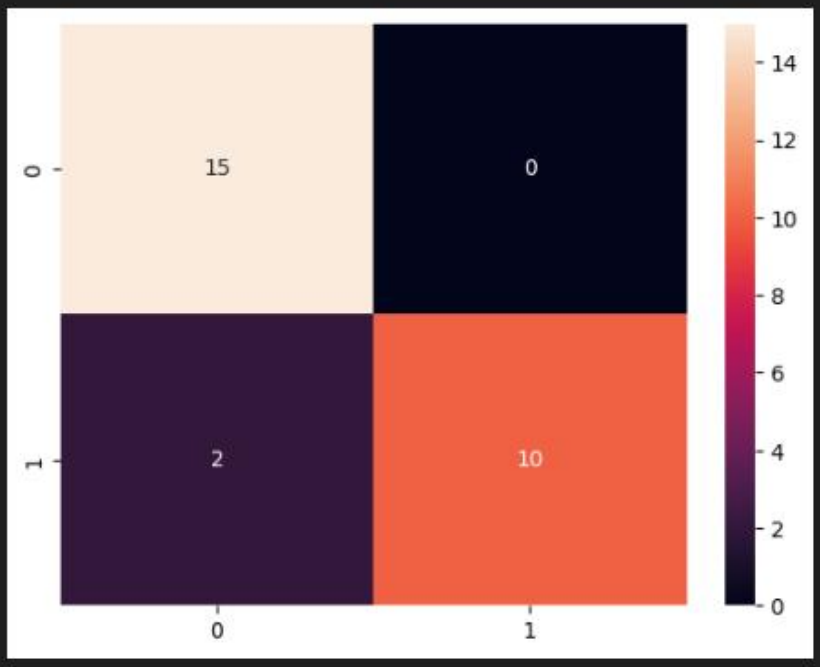
[22] ✓ 0.0s

```
              precision    recall  f1-score   support

           0       1.00      0.88      0.94        17
           1       0.83      1.00      0.91        10

    accuracy                           0.93        27
   macro avg       0.92      0.94      0.92        27
weighted avg       0.94      0.93      0.93        27
```

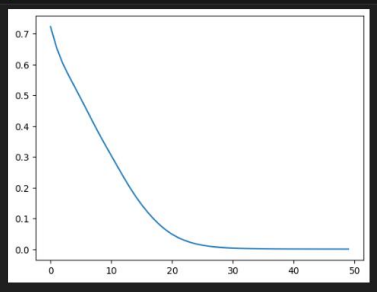
```
#Importing Confusion Matrix
#Comparing the predictions against the actual observations in y_val
cm = confusion_matrix(y_pred, y_test)
sns.heatmap(cm, annot=True)
```

[23] ✓ 1.0s

```
<Axes: >
```



```python
# Plotting graph for MLPClassifier

loss_values = model.loss_curve_
plt.plot(loss_values)
plt.show()
```




```python
# save the model to disk

filename = filename = "C:\\Users\\sp125\\OneDrive\\Desktop\\DNA CLASSIFICATION\\Code files\\mysite\\polls\\E-Coli_model.pickle"
pickle.dump(model, open(filename, 'wb'))
```

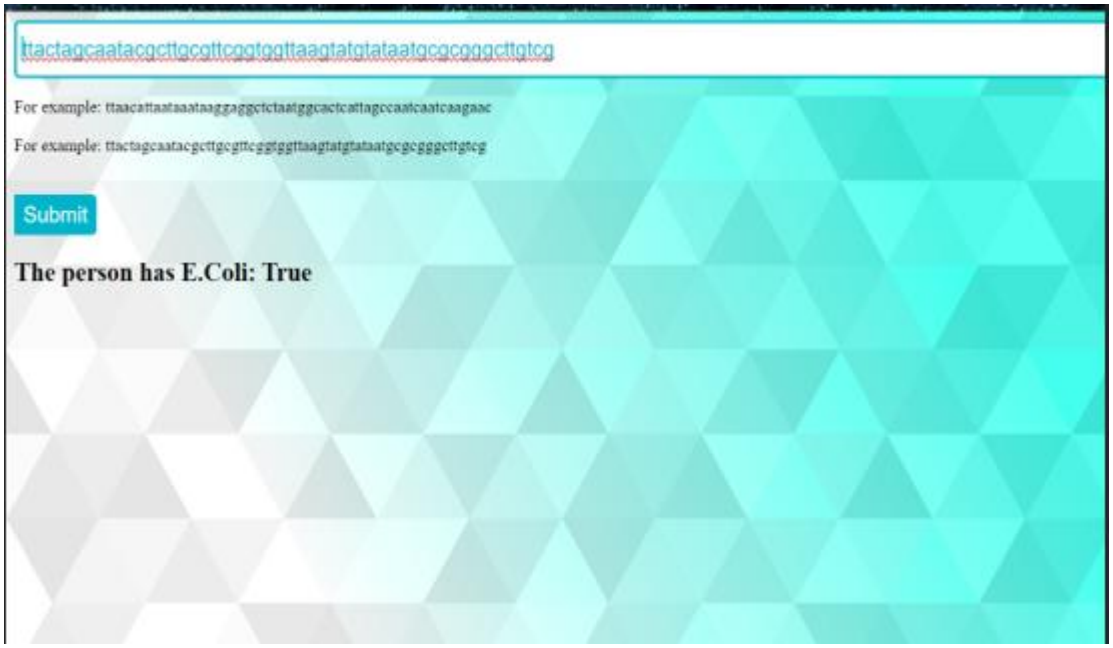
```python
genome = "ttactagcaatacgcttgcgttcggtggttaagtatgtataatgcgcgggcttgtcg"
# genome_false = "ttaacattaataaaataaggaggctctaatggcactcattagccaatcaatcaagaac"
genome_list = list(genome)
print(genome_list)
df_test = pd.DataFrame(genome_list)
df_test = df_test.transpose()
```

```
['t', 't', 'a', 'c', 't', 'a', 'g', 'c', 'a', 'a', 't', 'a', 'c', 'g', 'c', 't', 't', 'g', 'c', 'g', 't', 't', 'c', 'g', 'g', 't', 'g', 'g', 't', 't', 'a', 'a', 'g', 't', 'a', 't', 'g', 't', 'a', 't', 'a', 'a', 't', 'g', 'c', '
```

USER INTERFACE(O/P) :

Code files > mysite > mysite > urls.py > ...

```python
"""mysite URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/3.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include("polls.urls")),
]

```



GITHUB LINK: https://github.com/Shreyaprasad21/Feynn-AI-Product-Service-Prototype-Development_DNA_classification