(name of your school)

C A P S T O N E P R O J E C T

Project Documentation

“A GRIEVANCE DESK THAT READS”

SDG 16: Peace, Justice and Strong Institutions · SDG 11: Sustainable Cities and Communities

Class XII

Artificial Intelligence — Subject Code 843
Part C: Capstone Project

Session _____

Submitted by:

Group members listed overleaf

Under the guidance of:

_____

Teacher (Artificial Intelligence)

GROUP MEMBERS

The Capstone Project is carried out by a group of three to five students.

S. NO.	NAME	ROLL NO.	ROLE IN THE PROJECT	SIGNATURE
1				
2				
3				
4				
5				

Certificate

This is to certify that the students named above, of Class XII, have satisfactorily completed the Capstone Project entitled **“A Grievance Desk that Reads”** during the academic session _____, as prescribed in Part C of Artificial Intelligence (Subject Code 843) by the Central Board of Secondary Education, and have submitted this documentation and the accompanying three-minute video.

Teacher-in-charge

External Examiner

Principal

ACKNOWLEDGEMENT

We would like to thank _____, our Artificial Intelligence teacher, for guiding this capstone through every step of the Data Science Methodology.

We are grateful to the Principal of _____ for the computer lab and the support that made the project possible.

Our thanks go to _____, who helped us understand the problem and how it is handled today, and to everyone who shared data with us or answered our questions.

Finally, we thank our families for their patience while the project was built, tested and filmed.

_____
(Group members)

INDEX

SL. NO.	TOPIC	PAGE NO.
1	Introduction	5
2	Business Understanding	6
3	Analytic Approach	7
4	Data Requirements	9
5	Data Collection	9
6	Data Understanding	11
7	Data Preparation	12
8	Modelling	14
9	Evaluation	14
10	Deployment	17
11	Feedback	17
12	Ethical Considerations	18
13	The Project in Orange	18
14	Technical Details	19
15	Source Code	20
16	Output of a Real Run	25
17	Testing	27
18	Team Roles and Timeline	28
19	Results and Future Scope	28
20	Conclusion	29
21	Video Script	29
22	Viva Preparation	29
23	Glossary	30
24	References	31
25	Appendix: The Data	32

INTRODUCTION

Citizens can file grievances online, through portals such as CPGRAMS at the national level and city portals and helplines locally. Every grievance must first be read by someone and forwarded to the department that can act on it.

When thousands arrive each month, that first step becomes a queue. A complaint about a sparking transformer can wait days before it even reaches the electricity department, and some are forwarded to the wrong department and bounce back.

This capstone builds a model that reads each grievance and routes it at once. Text is not numbers, so the project is a natural language processing problem: the words must be cleaned and turned into numbers before any model can learn from them. Following the Data Science Methodology, it also decides what the model should do when it is unsure.

Who would use it: The grievance cell of a municipal corporation, the department officers who resolve complaints, and the citizens who file them.

Alignment to the Sustainable Development Goals

SDG 16 — Peace, Justice and Strong Institutions; SDG 11 — Sustainable Cities and Communities. SDG target 16.6 asks for effective, accountable and transparent institutions at all levels, and target 16.7 for responsive decision-making. A grievance that waits a week in a pile before anyone forwards it is the opposite of responsive. Routing complaints to the right department on the day they arrive makes local government quicker to act on what citizens report — broken roads, dirty water, overflowing drains.

The Data Science Methodology

The project follows the ten-step Data Science Methodology of John Rollins (IBM), taught in Unit 2 of the Class XII handbook. The sections that follow take the steps in order.

MODULE	STEPS
1. From Problem to Approach	Business Understanding; Analytic Approach
2. From Requirements to Collection	Data Requirements; Data Collection
3. From Understanding to Preparation	Data Understanding; Data Preparation
4. From Modelling to Evaluation	Modelling; Evaluation
5. From Deployment to Feedback	Deployment; Feedback

BUSINESS UNDERSTANDING

What is the problem that you are trying to solve?

The problem was scoped with the 5W1H Problem Canvas.

Who? • Citizens who file grievances • The grievance cell that forwards them • Department officers who resolve them	**What?** • Grievances wait in a queue before reaching a department • Some are sent to the wrong department
Where? • A city's online grievance portal	**When?** • As each grievance arrives, every day
Why? • Quick routing means quicker action on broken services • Officers' time goes on solving, not sorting	**How?** • Learn from grievances already routed by clerks • Route the clear ones automatically; send unclear ones to an officer

Problem statement

> Our **the municipal grievance cell** have a problem of **forwarding every grievance to the right department quickly** when **thousands of short, informal complaints arrive each month, written in citizens' own words**. An ideal solution would **route most grievances correctly the moment they arrive, and pass the unclear ones to a person**.

Talking to the stakeholders

The handbook pairs the 5W1H canvas with the Design Thinking framework, which starts by empathising with the people who have the problem. These are the questions the group put to them, and what each answer changed in the project.

QUESTION ASKED	WHAT WE LEARNED, AND WHAT IT CHANGED
To a clerk in the grievance cell: how do you decide where a grievance goes?	By the words: 'tap', 'pipeline' means water; 'pothole' means roads. That is exactly what a text model can learn.
To the same clerk: are some grievances hard to route?	Yes — 'drain water flooding the road' could be roads or sanitation, and clerks disagree. So some labels are judgement calls, and no model will get them all right.
To a department officer: what happens when a grievance comes to you wrongly?	It is sent back and loses days. A wrong routing is costly, so it is better to send an unclear grievance to a person than to guess.
To a citizen: how do you write a complaint?	Short, in my own words, sometimes with Hindi words like 'bijli' or 'paani', and spelling mistakes. The data must include that, not neat sentences.
To the clerk: what new kinds of complaints have appeared?	Online fraud — people come to the city when it should go to the cyber cell. The model must not force a new kind of complaint into an old department.
To the cell's head: what would you trust?	Something that tells me how sure it is. That led to the confidence threshold.

How the problem is handled today

Manual forwarding. A clerk reads and forwards each grievance. Accurate when there is time, but slow when thousands arrive.

Category drop-downs. Portals ask citizens to pick a category. Many pick the wrong one or 'Others', and the clerk must still read the text.

Keyword rules. 'If it says water, send to Water Supply.' Simple, but fails on 'dirty water in the school tap' and on misspellings like 'watr'.

Objectives

1. To frame grievance routing with the 5W1H canvas and interviews
2. To clean informal text — case, punctuation, stop words — and turn it into numbers with TF-IDF
3. To train Naive Bayes and Logistic Regression text classifiers and compare them with stratified 5-fold cross validation, using macro F1
4. To find which departments and which kinds of grievances the model confuses
5. To deploy routing with a confidence threshold that sends unclear grievances to a human officer
6. To test it on a new month, including a kind of complaint never seen before

ANALYTIC APPROACH

How can we use data to answer the question?

Type of analytics: Predictive.

Which department should handle a grievance is a prediction from its text — predictive analytics, with classification into six classes.

The input is text, so the approach includes natural language processing: text normalisation (lowercase, removing punctuation and stop words) and turning text into numbers with TF-IDF — the same ideas as the bag-of-words and TF-IDF met in the NLP unit.

Approaches considered

CANDIDATE	WHY IT WAS CONSIDERED
Multinomial Naive Bayes	A classic, fast text classifier. It learns how likely each word is in each department's grievances.
Logistic Regression	Learns a weight for every word for every department, and gives a probability for each department — used for the confidence threshold.
TF-IDF	Not a model but the step that makes text into numbers: words frequent in one grievance but rare across all of them get the largest values.

DATA REQUIREMENTS

What are the data requirements?

The data must be grievances in citizens' own words, each with the department that finally handled it. The department is the label the model learns from.

It must be messy in the ways real grievances are — short, misspelt, with Hindi words, sometimes about two departments — or the model will look better than it is.

DATA NEEDED	TYPE	FORMAT	SOURCE
Grievance ID	Text	e.g. G10001	Portal
Grievance text	Text	as the citizen typed it	Portal
Department (label)	Category	one of six departments	The clerk's final routing

DATA COLLECTION

What occurs during data collection?

A real project would export past grievances and their final departments from the city's grievance portal, with personal details such as names, phone numbers and addresses removed before use. Interviews with the clerk, officers and citizens are primary data and shaped the approach.

The sample file, grievances.csv, holds 1752 grievances across six departments, written to behave like real ones — including misspellings, Hindi words, grievances touching two departments and vague ones. new_month.csv holds 293 grievances from a later month, including a kind never seen before, and is used only for feedback.

> **About this dataset.** The data is a LambdaLab sample made by a program to behave like the real thing, so the project could be run and tested. It is not real data. A submitted capstone must use data the group collected or a credited public dataset.

The columns of the dataset

COLUMN	TYPE	WHAT IT HOLDS	ROLE
`id`	text	Grievance ID	used to find duplicates
`grievance`	text	The complaint as typed; sometimes blank	cleaned, then TF-IDF features
`department`	category	Water Supply, Electricity, Roads, Sanitation, Health or Education; typed by hand	target
`clean`	text	Made by the program: lowercase, no punctuation, no stop words	model input

grievances.csv — Past grievances and the department each was routed to. 1,752 rows, 131.3 KB.

new_month.csv — A later month, used only for feedback. 293 rows, 21.7 KB.

The first eight rows of grievances.csv

ID	GRIEVANCE	DEPARTMENT
G10001	Dustbin overflowing near gandhi nagar. kindly take action!!	Sanitation
G10002	pipeline burst and water is wasted near gandhi nagar	Water Supply
G10003	Water tanker did not come to nehru park area. please help	Water Supply
G10004	dran pani floodig the road in our colony	Roads
G10005	muddy water coming from the tap in ward 12. kindly take action	Water Supply
G10006	Dengue mosquitoes from blocked naali in ward 7. this is very urgent	Sanitation
G10007	work not done properly in gandhi nagar	Education
G10008	footpath broken and people fall in ward 12. please help!!	Roads

The first four rows of new_month.csv

ID	GRIEVANCE	DEPARTMENT
G20001	meter reading wrong in my electricity bill. this is very urgent	Electricity
G20002	no electricity in gandhi nagar for ten days. please help	Electricity
G20003	Bijli bill is too high this month. kindly take action	Electricity
G20004	Water meter not working at my house in our lane!!	Water Supply

DATA UNDERSTANDING

Is the data collected representative of the problem to be solved?

Counting the department column as typed shows a problem at once: the same department appears as 'Roads', 'roads' and 'ROADS', which a computer treats as three different classes.

Chart 1 shows the departments are not equally busy: Water Supply, Sanitation and Electricity receive about twice as many grievances as Education. With uneven classes, overall accuracy can hide a department that is badly served — so macro F1, the average of each department's F1, is used.

Chart 2 shows how short grievances are: about 5.2 words after cleaning. Chart 3 shows the commonest words per department — 'water', 'pipeline' and 'tap'; 'pothole' and 'repair'; 'school' and 'students'. It also shows 'ward' near the top of almost every department: a place word that says nothing about the department. TF-IDF gives such words low weight because they appear everywhere.

Figure 1: Grievances received by each department

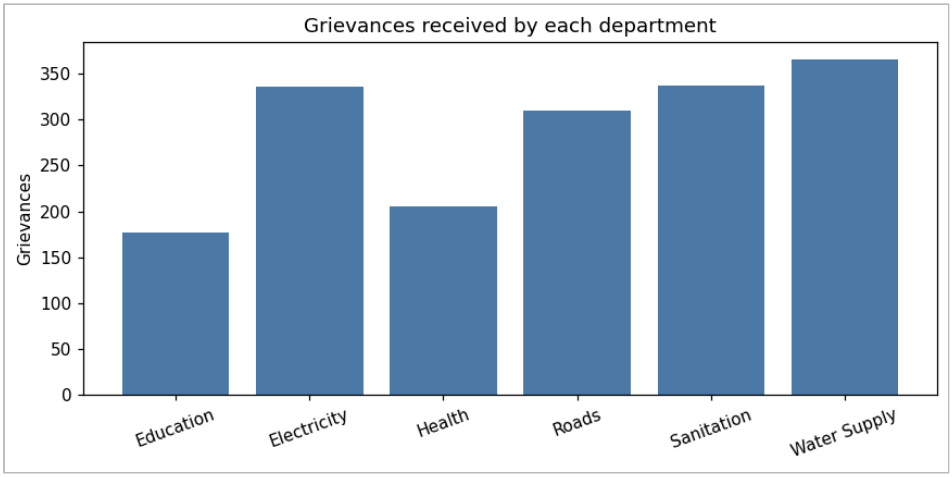


Figure 1 — Grievances received by each department, a bar chart drawn by the program (chart1_departments.png).

How to read it. The number of grievances for each department, after cleaning.

Water Supply, Sanitation and Electricity are the busiest; Education and Health the quietest. The classes are uneven, so macro F1 is used to judge the models fairly.

Figure 2: How long the grievances are

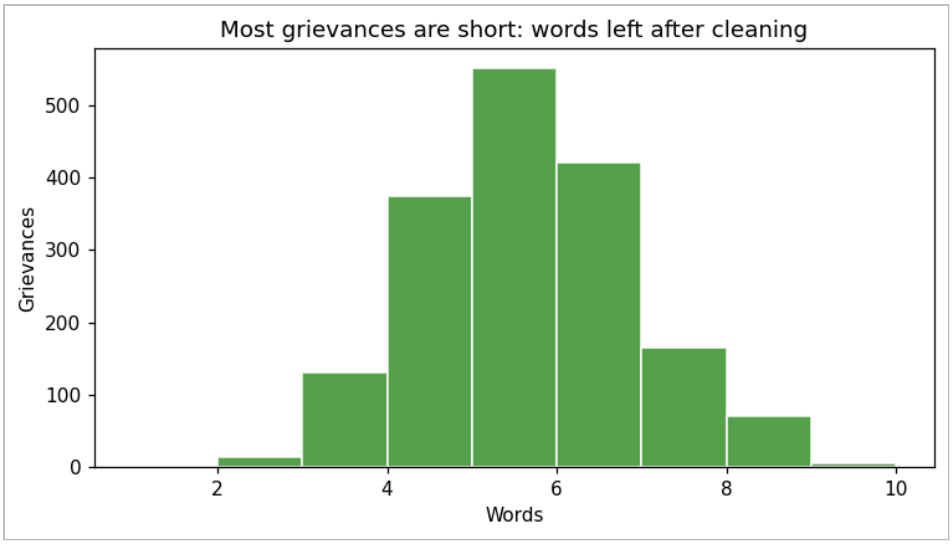


Figure 2 — How long the grievances are, a histogram drawn by the program (chart2_lengths.png).

How to read it. How many words each grievance has after cleaning.

Most grievances have four to seven words. The model must decide from very little text, so every word counts.

DATA PREPARATION

What additional work is required to manipulate and work with the data?

PROBLEM	HOW MUCH	WHAT WAS DONE, AND WHY
Grievances submitted twice	`12`	Removed by grievance ID, so each complaint counts once. Two different citizens writing the same words are kept — they are different grievances.
Blank grievances	`9`	Removed: there are no words to learn from.
Department names in different cases	`6 departments after fixing`	Stripped and put in title case, so 'ROADS' and 'roads' become 'Roads'.
Informal text	`Capitals, punctuation such as '!!', filler words`	Text normalisation: lowercase, punctuation removed, stop words such as 'the', 'please', 'kindly' and numbers removed.

Feature engineering

TF-IDF turns each cleaned grievance into a row of numbers, one column per word in the vocabulary — 404 words, learned from the training grievances only. A word's value is high when it appears in this grievance (term frequency) and is rare across all grievances (inverse document frequency).

The program prints one grievance's TF-IDF values: in 'low water pressure old city', 'pressure' and 'low' score highest, while 'water' scores lower because it appears in so many grievances — the IDF part at work.

Stop-word removal has a cost worth noticing: removing 'not' turns 'did not come' into 'did come'. For routing it does not matter — the department is the same — but for judging what a sentence means, it would. This is a known limit of bag-of-words.

Figure 3: The commonest words in each department

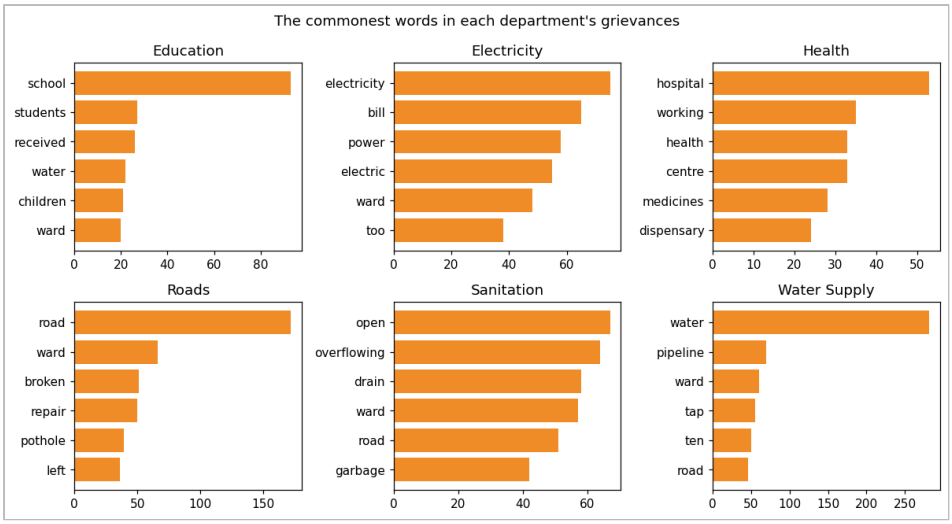


Figure 3 — The commonest words in each department, a bar chart drawn by the program (chart3_top_words.png).

How to read it. The six most frequent words in each department's cleaned grievances.

Each department has clear signal words, which is why a text model can work. Words like 'ward' and 'road' appear in several departments — the first carries no information, the second causes confusion between Roads and Sanitation.

MODELLING

In what way can the data be visualised to get to the answer that is required?

AI model ▶ Learning based ▶ Supervised learning ▶ Classification

Each past grievance has the department that handled it, so this is supervised classification into six classes.

The data is split 80 : 20 with stratify, so each department has the same share in training and testing: 1384 grievances to train on and 347 to test.

How the models work

Each model is a pipeline of two steps: TfidfVectorizer turns text into numbers, then the classifier learns. Keeping both in one pipeline means the vocabulary is learned from the training grievances only, so the test grievances stay truly unseen — and inside cross validation, from each fold's training part only.

Naive Bayes learns how often each word appears in each department. Logistic Regression learns a weight for every word for every department; for a new grievance, the department with the largest total gets the highest probability.

EVALUATION

Does the model used really answer the initial question, or does it need to be adjusted?

On the test set, Naive Bayes routed 0.9164 of grievances correctly and Logistic Regression 0.9222; their macro F1 scores were 0.9152 and 0.9216.

Because the classes are uneven, stratified 5-fold cross validation was used to compare them fairly: mean macro F1 0.9106 for Naive Bayes and 0.9126 for Logistic Regression. The difference is small; Logistic Regression is chosen, also because its probabilities are needed for the threshold.

27 test grievances were routed wrongly. Reading them is more useful than any score: every one of them is a grievance that touches two departments — 'hospital has no electricity', 'dengue mosquitoes from blocked drain' — or a vague one like 'officials not responding about the problem', which no model could route from the words alone. These are the grievances clerks disagree about too.

The model's confidence helps. Its average highest probability was 0.799 when it was right but 0.447 when wrong. With a threshold of 0.5, 17 of 27 wrong routings would go to an officer instead, and the automatic routings would be right 0.9682 of the time.

Figure 4: Confusion matrix, test set

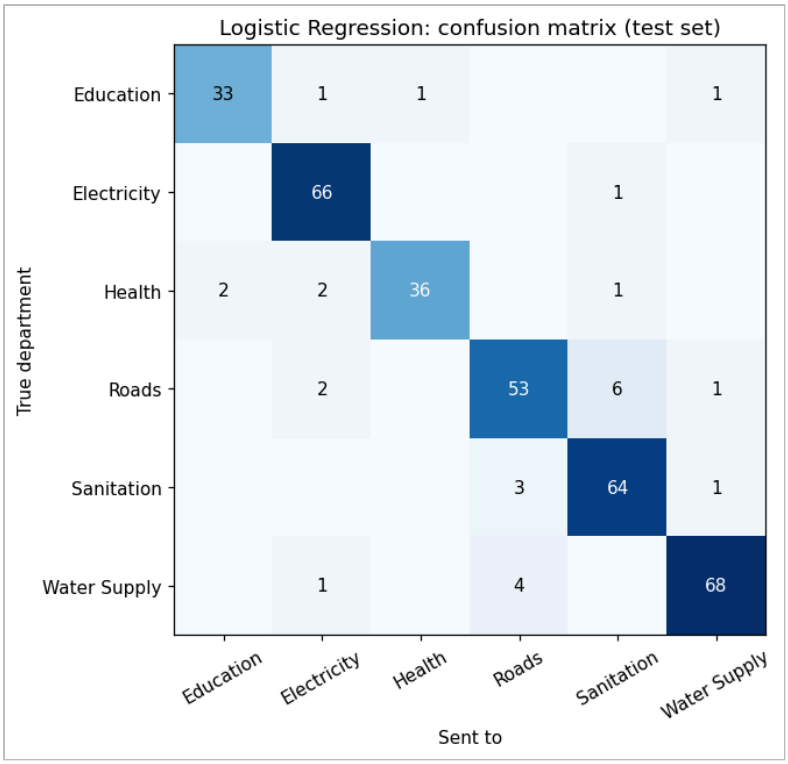


Figure 4 — Confusion matrix, test set, a confusion matrix drawn by the program (chart4_confusion.png).

How to read it. Rows are the true departments, columns where the model sent each test grievance. The diagonal is correct routings.

Most grievances sit on the diagonal. The mistakes cluster between departments that share words — Roads and Sanitation above all, then Roads and Water Supply, and Health with Electricity or Education — exactly the pairs the clerk said were judgement calls.

Figure 5: Precision and recall for each department

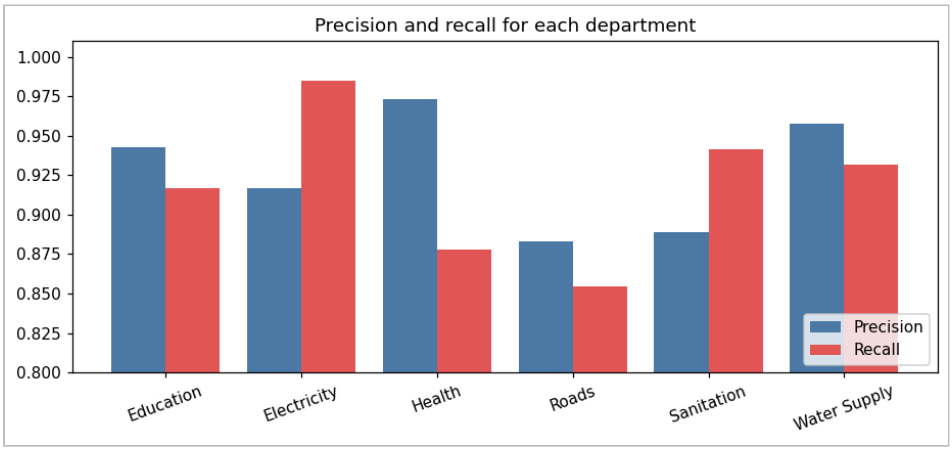


Figure 5 — Precision and recall for each department, a bar chart drawn by the program (chart5_per_department.png).

How to read it. For each department, precision (of grievances sent there, how many belonged) and recall (of its grievances, how many arrived).

Roads has the lowest scores, because it shares words with Sanitation and Electricity. Health has high precision but lower recall: some health grievances mention a school, electricity or a drain and are sent there instead.

Figure 6: Stratified 5-fold cross validation

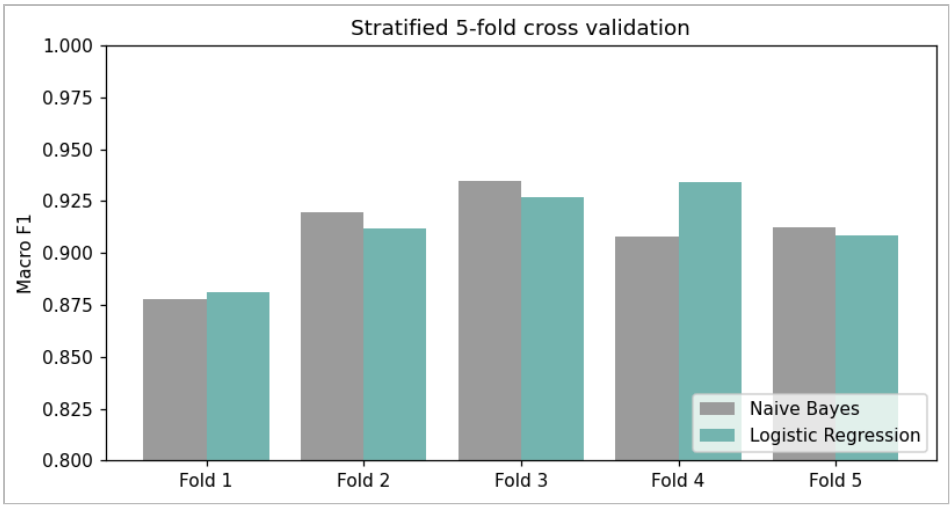


Figure 6 — Stratified 5-fold cross validation, a bar chart drawn by the program (chart6_kfold.png).

How to read it. Macro F1 for both models in each of five folds, with every department in the same proportion in each fold.

The two models are very close and each wins some folds. The averages decide, narrowly, for Logistic Regression — which also gives the probabilities the threshold needs.

Figure 7: How sure the model was

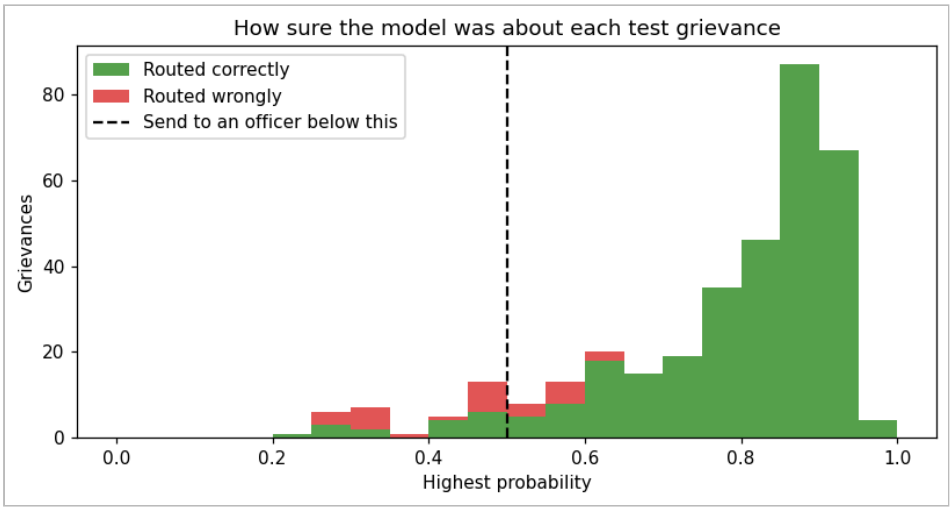


Figure 7 — How sure the model was, a histogram drawn by the program (chart7_confidence.png).

How to read it. The model's highest probability for each test grievance; green were routed correctly, red wrongly. The dashed line is the 0.5 threshold.

Correct routings are mostly confident; wrong ones mostly are not. Sending everything below 0.5 to an officer removes most of the mistakes while leaving most grievances automatic.

DEPLOYMENT

Can you put the model into practice?

The chosen pipeline is saved to grievance_model.pkl. As each grievance arrives, it is cleaned exactly as in training and given to the model. If the highest probability is 0.5 or more, the grievance goes straight to that department; otherwise it goes to a human officer in the grievance cell.

The program shows three: a street light (Electricity), garbage and a blocked drain (Sanitation), and a missing pension — which belongs to none of the six departments — sent to an officer because the model is unsure. A system that knows when it does not know is safer than one that always answers.

Figure 8: Feedback: the new month

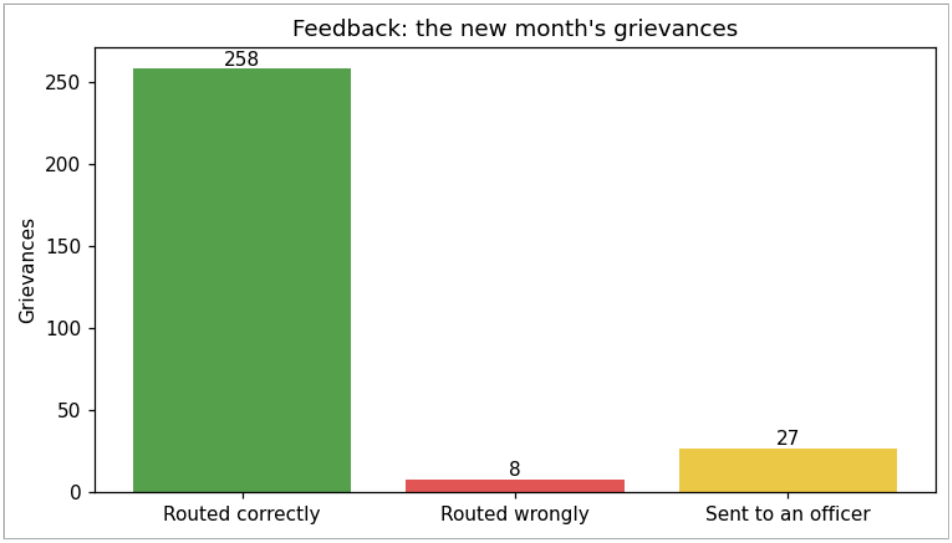


Figure 8 — Feedback: the new month, a bar chart drawn by the program (chart8_new_month.png).

How to read it. What happened to each of the new month's grievances: routed correctly, routed wrongly, or sent to an officer.

Most were routed correctly and automatically; only a few were routed wrongly. Every online-fraud grievance, a kind never seen before, went to an officer instead of being forced into the wrong department.

FEEDBACK

Can you get constructive feedback into answering the question?

The new month brought 293 grievances. 266 were routed automatically and 27 sent to an officer; of the automatic routings, 258 of 266 were correct.

The month included grievances about online fraud — a department the model has never seen. All of them went to an officer (6 of 6). That is the threshold doing its job, but it is also a signal: when officers keep receiving the same new kind of grievance, a new department (here, the cyber cell) should be added, and the model retrained with those examples.

The feedback loop is the officers themselves: every grievance an officer routes by hand, and every wrong routing a department sends back, becomes a new labelled example for the next retraining.

ETHICAL CONSIDERATIONS

Personal information

Grievances can contain names, phone numbers and addresses. These must be removed before the text is used for training, and the model kept within the government's systems.

Every voice is heard

Citizens who write in another language, or with many spelling mistakes, must not be routed worse. The officer threshold catches unclear grievances, and mistakes should be checked by language and region.

Accountability

Routing speeds up a grievance; it does not decide it. Departments remain responsible for resolving complaints, and a citizen can always ask for a review.

Transparency

Citizens should be told that routing is automatic and that they can ask for a human review if it seems wrong.

THE PROJECT IN ORANGE

Orange's Text Mining add-on has the same steps as widgets: preprocessing, bag of words with TF-IDF, and classifiers. The tool is **Orange Data Mining (Text add-on)**.

WIDGET	WHAT IT DOES
Corpus	Load grievances.csv; use 'grievance' as the text and 'department' as the target.
Preprocess Text	Lowercase, remove punctuation, and filter stop words.
Bag of Words	Choose TF-IDF as the term frequency weighting.
Naive Bayes and Logistic Regression	Connect both to Test and Score.
Test and Score	Choose stratified 5-fold cross validation; compare F1.
Confusion Matrix	See which departments are confused, and click a cell to read the grievances.

TECHNICAL DETAILS

Programming language	Python 3
Libraries	pandas, NumPy, Matplotlib, scikit-learn, joblib
Where to run it	Jupyter Notebook or Google Colab (grievance_desk.ipynb), or any Python IDE (grievance_desk.py)
No-code alternative	Orange Data Mining (Text add-on)
Model type	Learning based → Supervised learning → Classification
Charts produced	8
Test cases run	10

Setting it up

1. Install Python 3 (Anaconda includes Jupyter and most libraries), or open Google Colab in a browser.
2. Install the libraries: **pip install pandas numpy matplotlib scikit-learn**
3. Keep the program and its data files together in one folder.
4. Run **python grievance_desk.py**, or open **grievance_desk.ipynb** and run each cell in order.

The library calls it uses

CALL	STEP	WHAT IT DOES
`pd.read_csv(keep_default_na=False)`	Step 4	Reads blank grievances as empty text, not NaN
`drop_duplicates(subset="id")`	Step 6	Removes grievances submitted twice
`str.strip().str.title()`	Step 6	Fixes the case of department names
`clean(): lower, punctuation, stop words`	Step 6	Text normalisation
`TfidfVectorizer()`	Steps 6–7	Turns text into TF-IDF numbers
`make_pipeline(TfidfVectorizer(), MultinomialNB())`	Step 7	The Naive Bayes text classifier
`make_pipeline(TfidfVectorizer(), LogisticRegression())`	Step 7	The Logistic Regression text classifier
`StratifiedKFold + cross_val_score(scoring="f1_macro")`	Step 8	Fair comparison across uneven departments
`predict_proba().max()`	Steps 8–9	The model's confidence, for the officer threshold
`joblib.dump() / load()`	Step 9	Saves and reloads the pipeline

SOURCE CODE

The complete program, laid out as the ten steps of the methodology. Every figure and number in this documentation came from running it.

`grievance_desk.py`

```
# ----------------------------------------------------------------------------
# grievance_desk.py
#
# Capstone project: a grievance desk that reads each citizen's complaint and
# sends it to the right department, so it is not left waiting in a pile.
#
# CBSE Class XII, Artificial Intelligence (843), Part C - Capstone Project.
# SDG 16 (Peace, Justice and Strong Institutions).
#
# Data Science Methodology, 10 steps. NATURAL LANGUAGE PROCESSING: the text
# is cleaned, turned into numbers with TF-IDF, and classified into six
# departments. Naive Bayes is compared with Logistic Regression, with
# stratified 5-fold cross validation. Grievances the model is unsure about
# go to a human officer.
#
# Libraries: pandas, numpy, matplotlib, scikit-learn, joblib.
# ----------------------------------------------------------------------------

# %% Step 1 - Business Understanding
print("=" * 64)
print("THE GRIEVANCE DESK")
print("=" * 64)
print("Problem : grievances wait for days before someone reads them and")
print("          forwards them to the right department")
print("Goal    : read each grievance and route it at once; send the")
print("          unclear ones to a human officer")
print()

# %% Step 2 - Analytic Approach
print("Analytic approach : predictive - text classification into 6 departments")
print()

# %% Step 3 - Data Requirements
import string
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.model_selection import train_test_split, StratifiedKFold, cross_val_score
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score,
f1_score
import joblib

# %% Step 4 - Data Collection
data = pd.read_csv("grievances.csv", keep_default_na=False)
print("Grievances collected :", len(data))
print()
```

```
# %% Step 5 - Data Understanding
print("--- Step 5: Data Understanding ---")
print("Department names as typed:")
print(data["department"].value_counts().to_string())
print()

# %% Step 6 - Data Preparation
print("--- Step 6: Data Preparation ---")
before = len(data)
data = data.drop_duplicates(subset="id")
print("Grievances submitted twice, removed :", before - len(data))
blank = data["grievance"].str.strip() == ""
print("Blank grievances removed            :", int(blank.sum()))
data = data[~blank].copy()
data["department"] = data["department"].str.strip().str.title()
print("Department names after fixing case  :", data["department"].nunique())

STOP = {"a", "an", "the", "in", "on", "at", "of", "to", "is", "are", "and", "for", "from", "near",
        "my", "our", "this", "it", "by", "has", "was", "very", "please", "kindly", "since",
        "so", "no", "not", "we", "nobody", "again", "every", "last", "take", "action", "help",
        "soon", "solve", "urgent", "request", "quick", "listening", "be", "days", "week", "month"}


def clean(text):
    text = text.lower()
    for mark in string.punctuation:
        text = text.replace(mark, " ")
    words = [w for w in text.split() if w not in STOP and not w.isdigit()]
    return " ".join(words)


data["clean"] = data["grievance"].apply(clean)
print("Example before cleaning :", data["grievance"].iloc[2])
print("Example after cleaning  :", data["clean"].iloc[2])
data["words"] = data["clean"].str.split().str.len()
print("Average words per grievance after cleaning :", round(data["words"].mean(), 1))
print()

depts = sorted(data["department"].unique())
counts = data["department"].value_counts().reindex(depts)
print("Grievances per department:")
print(counts.to_string())
print()

# Chart 1: grievances per department.
plt.figure(figsize=(8, 4))
plt.bar(depts, counts.values, color="#4e79a7")
plt.title("Grievances received by each department")
plt.ylabel("Grievances")
plt.xticks(rotation=20)
plt.tight_layout()
plt.savefig("chart1_departments.png")
plt.close()

# Chart 2: how long the grievances are.
plt.figure(figsize=(7, 4))
plt.hist(data["words"], bins=range(1, data["words"].max() + 2), color="#59a14f", edgecolor="white")
plt.title("Most grievances are short: words left after cleaning")
plt.xlabel("Words")
plt.ylabel("Grievances")
plt.tight_layout()
```

```
plt.savefig("chart2_lengths.png")
plt.close()

# Chart 3: the commonest words in each department.
plt.figure(figsize=(11, 6))
for k, d in enumerate(depts):
    words = " ".join(data[data["department"] == d]["clean"]).split()
    top = pd.Series(words).value_counts().head(6)
    plt.subplot(2, 3, k + 1)
    plt.barh(top.index[::-1], top.values[::-1], color="#f28e2b")
    plt.title(d)
plt.suptitle("The commonest words in each department's grievances")
plt.tight_layout()
plt.savefig("chart3_top_words.png")
plt.close()

# %% Step 7 - Modelling
print("--- Step 7: Modelling ---")
X_train, X_test, y_train, y_test = train_test_split(
    data["clean"], data["department"], test_size=0.2, stratify=data["department"], random_state=4)
print("Training grievances :", len(X_train))
print("Testing grievances  :", len(X_test))
bayes = make_pipeline(TfidfVectorizer(), MultinomialNB())
logistic = make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=2000))
bayes.fit(X_train, y_train)
logistic.fit(X_train, y_train)
vocab = bayes.named_steps["tfidfvectorizer"].get_feature_names_out()
print("Vocabulary learned (distinct words) :", len(vocab))
example = bayes.named_steps["tfidfvectorizer"].transform([X_train.iloc[0]])
print("TF-IDF example :", X_train.iloc[0])
for j in example.nonzero()[1]:
    print("   ", vocab[j], ":", round(example[0, j], 3))
print()

# %% Step 8 - Evaluation
print("--- Step 8: Evaluation ---")
pred_b = bayes.predict(X_test)
pred_l = logistic.predict(X_test)
print("Accuracy, Naive Bayes         :", round(accuracy_score(y_test, pred_b), 4))
print("Accuracy, Logistic Regression :", round(accuracy_score(y_test, pred_l), 4))
print("Macro F1, Naive Bayes         :", round(f1_score(y_test, pred_b, average="macro"), 4))
print("Macro F1, Logistic Regression :", round(f1_score(y_test, pred_l, average="macro"), 4))
folds = StratifiedKFold(n_splits=5, shuffle=True, random_state=1)
cv_b = cross_val_score(make_pipeline(TfidfVectorizer(), MultinomialNB()), data["clean"],
data["department"],
                       cv=folds, scoring="f1_macro")
cv_l = cross_val_score(make_pipeline(TfidfVectorizer(), LogisticRegression(max_iter=2000)),
data["clean"],
                       data["department"], cv=folds, scoring="f1_macro")
print("Macro F1 in each fold, Naive Bayes         :", np.round(cv_b, 3))
print("Macro F1 in each fold, Logistic Regression :", np.round(cv_l, 3))
print("Mean CV macro F1, Naive Bayes         :", round(cv_b.mean(), 4))
print("Mean CV macro F1, Logistic Regression :", round(cv_l.mean(), 4))
best_name = "Logistic Regression" if cv_l.mean() >= cv_b.mean() else "Naive Bayes"
best = logistic if best_name == "Logistic Regression" else bayes
pred = pred_l if best_name == "Logistic Regression" else pred_b
print("Chosen model :", best_name)
print()

prec = precision_score(y_test, pred, labels=depts, average=None)
rec = recall_score(y_test, pred, labels=depts, average=None)
```

```
print("Department      precision  recall")
for d, p, r in zip(depts, prec, rec):
    print(f"{d:<15} {p:9.3f} {r:7.3f}")
print()
wrong = [i for i in range(len(y_test)) if pred[i] != y_test.iloc[i]]
print("Grievances sent to the wrong department (test set) :", len(wrong))
print("The first eight:")
for i in wrong[:8]:
    print("  ", repr(data.loc[y_test.index[i], "grievance"]), "| true:", y_test.iloc[i], "| sent
to:", pred[i])
print()

# Chart 4: the confusion matrix.
cm = confusion_matrix(y_test, pred, labels=depts)
plt.figure(figsize=(7, 6))
plt.imshow(cm, cmap="Blues")
for a in range(len(depts)):
    for b in range(len(depts)):
        if cm[a, b] > 0:
            plt.text(b, a, cm[a, b], ha="center", va="center",
                     color="white" if cm[a, b] > cm.max() / 2 else "black")
plt.xticks(range(len(depts)), depts, rotation=30)
plt.yticks(range(len(depts)), depts)
plt.xlabel("Sent to")
plt.ylabel("True department")
plt.title(best_name + ": confusion matrix (test set)")
plt.tight_layout()
plt.savefig("chart4_confusion.png")
plt.close()

# Chart 5: precision and recall for each department.
x = np.arange(len(depts))
plt.figure(figsize=(8.5, 4))
plt.bar(x - 0.2, prec, width=0.4, label="Precision", color="#4e79a7")
plt.bar(x + 0.2, rec, width=0.4, label="Recall", color="#e15759")
plt.xticks(x, depts, rotation=20)
plt.ylim(0.8, 1.01)
plt.title("Precision and recall for each department")
plt.legend(loc="lower right")
plt.tight_layout()
plt.savefig("chart5_per_department.png")
plt.close()

# Chart 6: macro F1 in each fold.
x = np.arange(5)
plt.figure(figsize=(7.5, 4))
plt.bar(x - 0.2, cv_b, width=0.4, label="Naive Bayes", color="#9c9c9c")
plt.bar(x + 0.2, cv_l, width=0.4, label="Logistic Regression", color="#76b7b2")
plt.xticks(x, ["Fold 1", "Fold 2", "Fold 3", "Fold 4", "Fold 5"])
plt.ylim(0.8, 1.0)
plt.ylabel("Macro F1")
plt.title("Stratified 5-fold cross validation")
plt.legend(loc="lower right")
plt.tight_layout()
plt.savefig("chart6_kfold.png")
plt.close()

# How sure is the model? Its highest probability for each test grievance.
sure = best.predict_proba(X_test).max(axis=1)
right = pred == y_test.values
print("Average confidence when right :", round(sure[right].mean(), 3))
```

```
if (~right).sum() > 0:
    print("Average confidence when wrong :", round(sure[~right].mean(), 3))
else:
    print("Average confidence when wrong : none were wrong")
THRESHOLD = 0.5
print("Test grievances below", THRESHOLD, "confidence :", int((sure < THRESHOLD).sum()))
print("Wrong routings caught by the threshold :", int((~right & (sure < THRESHOLD)).sum()), "of",
int((~right).sum()))
auto_test = sure >= THRESHOLD
print("Accuracy of the automatic routings :", round(right[auto_test].mean(), 4))
print()

# Chart 7: confidence of right and wrong routings.
plt.figure(figsize=(7.5, 4))
plt.hist([sure[right], sure[~right]], bins=np.linspace(0, 1, 21), stacked=True,
         color=["#59a14f", "#e15759"], label=["Routed correctly", "Routed wrongly"])
plt.axvline(THRESHOLD, color="black", linestyle="--", label="Send to an officer below this")
plt.title("How sure the model was about each test grievance")
plt.xlabel("Highest probability")
plt.ylabel("Grievances")
plt.legend(loc="upper left")
plt.tight_layout()
plt.savefig("chart7_confidence.png")
plt.close()

# %% Step 9 - Deployment
print("--- Step 9: Deployment ---")
joblib.dump(best, "grievance_model.pkl")
model = joblib.load("grievance_model.pkl")


def route(text):
    cleaned = clean(text)
    probs = model.predict_proba([cleaned])[0]
    top = probs.argmax()
    if probs[top] < THRESHOLD:
        return "Human officer", probs[top]
    return model.classes_[top], probs[top]


for text in ["The street light outside my house is not working for a week",
             "Garbage is lying on the road and the drain is blocked",
             "My pension has not come for two months"]:
    dept, p = route(text)
    print(repr(text), "->", dept, "(confidence", str(round(p, 2)) + ")")
print()

# %% Step 10 - Feedback
print("--- Step 10: Feedback ---")
new = pd.read_csv("new_month.csv", keep_default_na=False)
print("Grievances in the new month :", len(new))
sent = [route(t)[0] for t in new["grievance"]]
new["sent_to"] = sent
known = new["department"].isin(model.classes_)
auto = new["sent_to"] != "Human officer"
print("Routed automatically        :", int(auto.sum()))
print("Sent to a human officer     :", int((~auto).sum()))
print("Automatic routings correct  :", int((new["sent_to"] == new["department"])[auto].sum()),
"of", int(auto.sum()))
print("Departments never seen in training :", sorted(new.loc[~known, "department"].unique()))
print("Of these grievances, sent to an officer :", int((~auto & ~known).sum()), "of",
```

```
int((~known).sum()))
for i in new.index[~known]:
    print("  ", repr(new.loc[i, "grievance"]), "->", new.loc[i, "sent_to"])

# Chart 8: what happened to the new month's grievances.
outcome = pd.Series(np.where(~auto, "Sent to an officer",
                    np.where(new["sent_to"] == new["department"], "Routed correctly", "Routed
wrongly")))
outcome = outcome.value_counts().reindex(["Routed correctly", "Routed wrongly", "Sent to an
officer"], fill_value=0)
plt.figure(figsize=(7, 4))
plt.bar(outcome.index, outcome.values, color=["#59a14f", "#e15759", "#edc948"])
for k, v in enumerate(outcome.values):
    plt.text(k, v + 2, str(v), ha="center")
plt.title("Feedback: the new month's grievances")
plt.ylabel("Grievances")
plt.tight_layout()
plt.savefig("chart8_new_month.png")
plt.close()
```

OUTPUT OF A REAL RUN

Output — python grievance_desk.py

```
================================================================
THE GRIEVANCE DESK
================================================================
Problem : grievances wait for days before someone reads them and
          forwards them to the right department
Goal    : read each grievance and route it at once; send the
          unclear ones to a human officer

Analytic approach : predictive - text classification into 6 departments

Grievances collected : 1752

--- Step 5: Data Understanding ---
Department names as typed:
department
Water Supply    364
Sanitation      341
Electricity     339
Roads           311
Health          206
Education       177
roads             4
ELECTRICITY       3
SANITATION        2
WATER SUPPLY      2
education         1
water supply      1
health            1

--- Step 6: Data Preparation ---
Grievances submitted twice, removed : 12
Blank grievances removed            : 9
Department names after fixing case  : 6
Example before cleaning : Water tanker did not come to nehru park area. please help
Example after cleaning  : water tanker did come nehru park area
Average words per grievance after cleaning : 5.2

Grievances per department:
department
```

```
Education       177
Electricity     336
Health          205
Roads           310
Sanitation      337
Water Supply    366

--- Step 7: Modelling ---
Training grievances : 1384
Testing grievances  : 347
Vocabulary learned (distinct words) : 404
TF-IDF example : low water pressure old city
    city : 0.416
    low : 0.529
    old : 0.416
    pressure : 0.536
    water : 0.295

--- Step 8: Evaluation ---
Accuracy, Naive Bayes         : 0.9164
Accuracy, Logistic Regression : 0.9222
Macro F1, Naive Bayes         : 0.9152
Macro F1, Logistic Regression : 0.9216
Macro F1 in each fold, Naive Bayes         : [0.878 0.92  0.935 0.908 0.912]
Macro F1 in each fold, Logistic Regression : [0.881 0.912 0.927 0.934 0.909]
Mean CV macro F1, Naive Bayes         : 0.9106
Mean CV macro F1, Logistic Regression : 0.9126
Chosen model : Logistic Regression

Department      precision  recall
Education           0.943   0.917
Electricity         0.917   0.985
Health              0.973   0.878
Roads               0.883   0.855
Sanitation          0.889   0.941
Water Supply        0.958   0.932

Grievances sent to the wrong department (test set) : 27
The first eight:
   'street light not working and road is dark and broken in block c. kindly take action' | true: Roads |
sent to: Electricity
   'Pothole filled with sewer paani in sector 4. we request quick action!!' | true: Sanitation | sent
to: Roads
   'hospital has no electricity for three days. nobody is listening' | true: Health | sent to:
Electricity
   'Naali watr flooding the sadak in our colony. nobody is listening' | true: Roads | sent to:
Sanitation
   'road dug for paani pipeline and left open in ward 12. nobody is listening' | true: Water Supply |
sent to: Roads
   'Officials not responding about the problem in our colony. kindly take action' | true: Water Supply |
sent to: Electricity
   'dengue mosquitoes from blocked drain in block c. kindly take action' | true: Health | sent to:
Sanitation
   'dirty water in the school tap in shanti vihar. nobody is listening' | true: Education | sent to:
Water Supply

Average confidence when right : 0.799
Average confidence when wrong : 0.447
Test grievances below 0.5 confidence : 33
Wrong routings caught by the threshold : 17 of 27
Accuracy of the automatic routings : 0.9682

--- Step 9: Deployment ---
'The street light outside my house is not working for a week' -> Electricity (confidence 0.73)
'Garbage is lying on the road and the drain is blocked' -> Sanitation (confidence 0.83)
'My pension has not come for two months' -> Human officer (confidence 0.42)

--- Step 10: Feedback ---
Grievances in the new month : 293
```

```
Routed automatically        : 266
Sent to a human officer     : 27
Automatic routings correct  : 258 of 266
Departments never seen in training : ['Cyber Crime']
Of these grievances, sent to an officer : 6 of 6
   'money taken from my bank account by an online fraud call' -> Human officer
   'fake website took my payment' -> Human officer
   'online shopping fraud, money not returned' -> Human officer
   'someone asked my otp on phone and took money' -> Human officer
   'my upi was used by a stranger' -> Human officer
   'fake job offer message took registration money' -> Human officer
```

TESTING

The program was tested by running it. Every case gives the whole program a small input of its own and records what actually came back.

TEST CASE	EXPECTED	ACTUAL RESULT	RESULT
Every grievance read	`Grievances collected : 1752`	`Grievances collected : 1752`	Pass
Department names typed in other cases are merged	`Department names after fixing case : 6`	`Department names after fixing case : 6`	Pass
Cross validation picks the better text model	`Chosen model : Logistic Regression`	`Chosen model : Logistic Regression`	Pass
A grievance about no known department goes to a person	`'My pension has not come for two months' -> Human officer (confidence 0.42)`	`'My pension has not come for two months' -> Human officer (confidence 0.42)`	Pass
Every grievance of a never-seen kind goes to an officer	`Of these grievances, sent to an officer : 6 of 6`	`Of these grievances, sent to an officer : 6 of 6`	Pass
Clear, distinct words: every test grievance routed right	`Accuracy, Logistic Regression : 1.0`	`Accuracy, Logistic Regression : 1.0`	Pass
Four blank grievances are removed	`Blank grievances removed : 4`	`Blank grievances removed : 4`	Pass
Three grievances submitted twice are removed	`Grievances submitted twice, removed : 3`	`Grievances submitted twice, removed : 3`	Pass
Upper-case department names are fixed	`Department names after fixing case : 6`	`Department names after fixing case : 6`	Pass
The data file is missing	`FileNotFoundError: [Errno 2] No such file or directory: 'grievances.csv'`	`FileNotFoundError: [Errno 2] No such file or directory: 'grievances.csv'`	Pass

TEAM ROLES AND TIMELINE

Division of work

ROLE	RESPONSIBLE FOR
Project lead	Business understanding; interviews with the clerk, officers and citizens
Data lead	Exporting grievances; removing personal details
Text analyst	Text normalisation, stop words and TF-IDF
Modeller	The two classifiers, cross validation and the threshold
Documentation and video	The report, the feedback month, and the 3-minute video

Timeline

WHEN	METHODOLOGY STEPS	DELIVERABLE
July	1 Business Understanding, 2 Analytic Approach	Problem statement; interviews at the grievance cell
August	3 Data Requirements, 4 Data Collection	Grievances exported and anonymised
September	5 Data Understanding, 6 Data Preparation	Text cleaned; TF-IDF built
October	7 Modelling, 8 Evaluation	Models compared; mistakes read one by one
November	9 Deployment, 10 Feedback	Threshold set; a new month tested
December – January	Documentation and video	Report and 3-minute video

RESULTS AND FUTURE SCOPE

What the project found

1. Text can be routed by its words: both models route about nine grievances in ten correctly.
2. Cross validation found the two models close — mean macro F1 0.9106 against 0.9126.
3. The mistakes are grievances about two departments, or vague ones — the same ones clerks disagree on.
4. A 0.5 confidence threshold catches 17 of 27 mistakes and leaves the automatic routings 0.9682 accurate.
5. A new kind of complaint (online fraud) went to officers every time, instead of the wrong department.

Limitations

- Sample grievances written from templates, for six departments only
- Bag-of-words ignores word order and negation
- Grievances in other Indian languages are not handled
- Some labels are judgement calls, which limits any model

Future scope

- Grievances in Hindi and other Indian languages
- Add new departments, such as the cyber cell, as feedback shows the need
- Also predict urgency, so a sparking transformer is handled before a bill query

- Let a grievance go to two departments when it is about both

What the group had to learn

- Text normalisation: tokens, stop words, punctuation
- Bag of words and TF-IDF, in scikit-learn
- Naive Bayes for text
- Macro F1 for uneven classes
- Confidence thresholds and human-in-the-loop systems

CONCLUSION

The capstone built a grievance desk that reads complaints in citizens' own words and routes about nine in ten correctly — and, with a confidence threshold, routes the clear ones almost without error while sending the unclear ones to a person.

Reading the mistakes taught more than the scores. They were grievances that belong to two departments, or say too little — the cases where people disagree too. A model cannot be more certain than the labels it learns from.

The most useful design choice was letting the model say 'I am not sure'. It kept a new kind of grievance out of the wrong department, and showed the city that a new department was needed.

VIDEO SCRIPT

Part C requires a video of exactly three minutes with six parts: the problem statement, the SDG the project is aligned to, the AI concepts and algorithms used, how the project works, the conclusion, and an acknowledgement to the teacher. The script below is timed to the second.

TIME	PART	WHAT TO SAY AND SHOW
0:00 – 0:25	a. Problem statement	Show a pile of complaints. "Every grievance waits for someone to read it and forward it. Some wait days."
0:25 – 0:40	b. SDG alignment	SDG 16 icon. "Responsive, accountable institutions — starting with reading every complaint quickly."
0:40 – 1:15	c. AI concepts and algorithms	"Natural language processing: cleaning text, TF-IDF, and two classifiers — Naive Bayes and Logistic Regression — compared with stratified 5-fold cross validation."
1:15 – 2:15	d. How it works	Type a grievance, show it cleaned, then routed. Then chart 7: confident routings go straight through, unsure ones to an officer. Then the fraud grievances going to an officer.
2:15 – 2:45	e. Conclusion	"Nine in ten routed correctly — and the model knows when to ask a person."
2:45 – 3:00	f. Acknowledgement	"We thank our AI teacher, [name], and the grievance cell staff who helped us."

VIVA PREPARATION

The viva voce is based on the Capstone Project. These are the questions the group prepared for, with the answers each member should be able to give in their own words.

Q1. Why must text be cleaned before modelling?
So that 'Water', 'water' and 'water!!' count as the same word, and filler words like 'please' do not take up the model's attention.

Q2. What is TF-IDF?
A value for each word in a document: high when the word is frequent in that document but rare across all documents.

Q3. Why did 'water' get a lower TF-IDF value than 'pressure'?
Because 'water' appears in many grievances, so its inverse document frequency is low.

Q4. Why macro F1 and not accuracy?
The departments have uneven numbers of grievances; macro F1 averages each department's F1, so a small department that is badly served still pulls the score down.

Q5. Why put TfidfVectorizer inside the pipeline?
So that the vocabulary is learned only from the training data, in every fold, and the test grievances remain unseen.

Q6. What does the confidence threshold do?
Grievances the model is less than 50% sure about go to a human officer instead of being guessed.

Q7. Why did the model make mistakes?
Most were grievances about two departments, or vague ones — cases where even clerks disagree.

Q8. Which SDG does it align to?
SDG 16, Peace, Justice and Strong Institutions — targets 16.6 and 16.7 on effective and responsive institutions.

GLOSSARY

The terms used in this documentation, as the Class XII handbook defines them.

TERM	MEANING
Data Science Methodology	A prescribed sequence of iterative steps (ten, in the IBM methodology) that data scientists follow to approach a problem and find a solution.
5W1H Problem Canvas	Who, What, Where, When, Why and How — the questions used in Business Understanding to define a problem fully.
Feature	An input column the model learns from.
Target / label	The column the model must predict.
Training data	The part of the data the model learns from.
Testing data	Data held back from training, used to judge the model on rows it has never seen.
Train-test split	Dividing the data once into training and testing sets, commonly 80 : 20.
K-fold cross validation	Splitting the data into k folds and training k times, each time testing on a different fold, so every row is tested once.
Overfitting	When a model learns the training data too closely, including its noise, and does worse on new data.
Data preparation	Cleaning, combining and transforming data into a form a model can use — usually the longest step.
Feature engineering	Selecting, modifying or creating features from raw data to improve a model.
Deployment	Putting the model into practice where its users can apply it.
Feedback	Results and comments from using the deployed model, used to refine it — the step that makes the methodology a cycle.
Classification	Supervised learning where the target is one of a fixed set of classes.
Confusion matrix	A table of actual against predicted classes: true positives, true negatives, false positives and false negatives.
Accuracy	(TP + TN) / (TP + TN + FP + FN): the share of all predictions that are correct.
Precision	TP / (TP + FP): of the cases predicted positive, the share that really are.
Recall	TP / (TP + FN): of the cases that really are positive, the share the model found.
F1 score	2 × precision × recall / (precision + recall): a balance of precision and recall.

REFERENCES

1. CPGRAMS — Centralised Public Grievance Redress and Monitoring System, DARPG — https://pgportal.gov.in/
2. Department of Administrative Reforms and Public Grievances — https://darpg.gov.in/
3. scikit-learn: Working with text data — https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html
4. CBSE Class X AI — Natural Language Processing (bag of words and TF-IDF)
5. CBSE, Artificial Intelligence (Subject Code 843), Class XII curriculum, 2026-27 — Part C.
6. CBSE, Artificial Intelligence Student Handbook, Class XII — Unit 2: Data Science Methodology.
7. Rollins, J. B. (2015). Foundational Methodology for Data Science. IBM Analytics.
8. scikit-learn documentation — https://scikit-learn.org/stable/
9. pandas, NumPy and Matplotlib documentation.
10. LambdaLab — https://www.lambdalab.in

APPENDIX: THE DATA

The first rows of each data file, exactly as the program read them.

grievances.csv — the first 40 rows

ID	GRIEVANCE	DEPARTMENT
G10001	Dustbin overflowing near gandhi nagar. kindly take action!!	Sanitation
G10002	pipeline burst and water is wasted near gandhi nagar	Water Supply
G10003	Water tanker did not come to nehru park area. please help	Water Supply
G10004	dran pani floodig the road in our colony	Roads
G10005	muddy water coming from the tap in ward 12. kindly take action	Water Supply
G10006	Dengue mosquitoes from blocked naali in ward 7. this is very urgent	Sanitation
G10007	work not done properly in gandhi nagar	Education
G10008	footpath broken and people fall in ward 12. please help!!	Roads
G10009	Water tanker did not come to gandhi nagar	Water Supply
G10010	drain water flooding the road in our lane. kindly take action	Sanitation
G10011	new road already damaged in station road. please help	Roads
G10012	Transformer sparking near gandhi nagar	Electricity
G10013	Water comes only for ten minutes in shanti vihar. kindly take action	Water Supply
G10014	speed breaker needed near the vidyalaya in block c!!	Roads
G10015	Drain is blocked and overflowing in shanti vihar. we request quick action	Sanitation
G10016	x ray machine not working at the hospital!!	Health
G10017	dustbin overflowing near the main market	Sanitation
G10018	Dirty water in the school tap in ward 12	Education
G10019	garbage not collected in block c for a month. we request quick action!!	Sanitation
G10020	health centre closed during working hours in our colony. this is very urgent	Health
G10021	tap paani is dirty and smells bad in the old city. nobody is listening	Water Supply
G10022	Dustbin overflowing near the main market. kindly take action!!	Sanitation
G10023	very low water pressure in station road for two weeks	Water Supply
G10024	power cut in block c every evening. we request quick action	Electricity

G10025	pipeline burst and water is wasted near ward 7. please solve this soon	Water Supply
G10026	uniform money not received by students. this is very urgent	Education
G10027	Very low pani pressure in ward 12 since sunday!!	Water Supply
G10028	teacher absent at the government vidyalaya in shanti vihar for ten days. please solve this soon	Education
G10029	Public toilet very dirty in gandhi nagar!!	Sanitation
G10030	Drinking paani not available since last week in nehru park area!!	Water Supply
G10031	hospital staff rude to patients	Health
G10032	Deep pothole caused an accidnt in sector 4!!	Roads
G10033	sadak full of gravel after repair in the old city. nobody is listening	Roads
G10034	pipeline burst and water is wasted near shanti vihar. nobody is listening	Water Supply
G10035	road full of gravel after repair in ward 12. we request quick action	Roads
G10036	Dead animal not removed from sector 4. kindly take action	Sanitation
G10037	garbage not collected in the main market for ten days. please help	Sanitation
G10038	No water supply in sector 4 since last week. kindly take action	Water Supply
G10039	no water supply in ward 7 for a month. nobody is listening	Water Supply
G10040	complaint given earlier but nothing done in ward 12	Water Supply

new_month.csv — the first 40 rows

ID	GRIEVANCE	DEPARTMENT
G20001	meter reading wrong in my electricity bill. this is very urgent	Electricity
G20002	no electricity in gandhi nagar for ten days. please help	Electricity
G20003	Bijli bill is too high this month. kindly take action	Electricity
G20004	Water meter not working at my house in our lane!!	Water Supply
G20005	tap water is dirty and smells bad in nehru park area	Water Supply
G20006	Water pieline leaking near station road since sunday	Water Supply
G20007	Public toilet very dirty in our colony	Sanitation
G20008	Hopital staff rude to patients	Health
G20009	meter reading wrng in my electricity bill. please help	Electricity
G20010	No water supply in sector 4 every evening. we request quick action	Water Supply
G20011	manhole open and smelling in our colony!!	Sanitation
G20012	school building roof leaking in block c. please solve this soon!!	Education
G20013	Very low watr pressure in block c for a month. please help	Water Supply
G20014	Public toilet very dirty in our colony. we request quick action	Sanitation
G20015	Work not done properly in block c. kindly take action	Sanitation
G20016	Water pipeline leaking near shanti vihar for three days	Water Supply
G20017	Water comes only for ten minutes in ward 12. nobody is listening!!	Water Supply
G20018	new road already damaed in gandhi nagar	Roads
G20019	Big potholes on the road in the main market. kindly take action	Roads
G20020	water pipeline leaking near the old city since last week. please help	Water Supply
G20021	No water supply in gandhi nagar for two weeks. we request quick action!!	Water Supply
G20022	Transformer sparkng near our lane!!	Electricity
G20023	Footpath broken and people fall in ward 12!!	Roads
G20024	dirty paani in the vidyalaya tap in our colony. kindly take action!!	Water Supply

G20025	meter reading wrong in my electricity bill!!	Electricity
G20026	fever spreading in the main market for ten days	Health
G20027	Pothole filled with sewer paani in nehru park area. kindly take action	Sanitation
G20028	Power cut in our lane again and again. please solve this soon!!	Electricity
G20029	money taken from my bank account by an online fraud call	Cyber Crime
G20030	work not done prperly in sector 4	Education
G20031	paani mter not working at my house in our colony	Water Supply
G20032	sadak dug for watr pipeline and left open in our lane. kindly take action	Roads
G20033	street light not working in gandhi nagar for three days	Electricity
G20034	Water tanker did not come to ward 7	Water Supply
G20035	sadak repair work left incomplete in our lane. please solve this soon	Roads
G20036	teacher absent at the government school in ward 7 since the rains. we request quick action	Education
G20037	meter reading wrong in my bijli bill. please solve this soon	Electricity
G20038	mosquitoes breeding in open drain in our colony. please solve this soon	Sanitation
G20039	dead animal not removed from sector 4	Sanitation
G20040	fake website took my payment	Cyber Crime