TN_SIR_2026_Impact_Analysis

January 6, 2026

1 Tamil Nadu SIR 2026 Draft Voter Roll Impact Analysis

1.0.1 Objective

The goal is to evaluate whether **voter roll revisions under Tamil Nadu SIR (2026)** are large enough to **numerically affect election outcomes**, using **victory margin** from the 2021 Assembly Election as the sensitivity threshold.

This is a **data-driven impact analysis**, not an allegation-based study.

1.1 Tamil Nadu Total Voters Count

SIR - Special Intensive Revision SSR - Special Summary Revision

1.1.1 Key Numbers

2021 Assembly Electoral Roll: **6,29,43,512 crore voters** SSR 2025 (as on 06-01-2025, before SIR): **6,36,12,950 crore voters** After SIR Draft Roll: **5,43,76,756 crore voters**

1.1.2 What Changed

After SIR 2026: **97,37,831 lakh** voters deleted

1.1.3 One-Line Takeaway

The **SIR 2026 exercise removed 97,37,831 lakh voters**, bringing Tamil Nadu's electorate down from **6.36 crore to 5.43 crore**, a **net drop of ~0.97 crore**.

Sources

https://en.wikipedia.org/wiki/2021_Tamil_Nadu_Legislative_Assembly_election
https://www.elections.tn.gov.in/ACwise_Gendercount_06012025.aspx
https://www.elections.tn.gov.in/ACwise_Gendercount_19122025.aspx

1.2 Analytical Framework

This analysis is based on three simple ideas:

1. **Victory Margin (2021)**
   - The number of votes that decided the election result.
2. **Not Voted (2021)**
   - Registered voters who did not vote, used as a baseline for comparison.

3. **Removed Voters (2026)**
    - Changes in the voter roll after the SIR revision. ### Core Question > Are voter roll changes large enough to be considered significant when compared to the margin that decided the election?

[1]:
```
import pandas as pd
import sqlite3

conn = sqlite3.connect("../db/tn_election.db")
```

1.3 Load Curated Tables

The following pre-processed tables are loaded from SQLite:

- `election_results_2021` → victory margin & parties
- `voters_2021` → turnout & not voted baseline
- `voters_2026` → revised voter roll (SIR 2026)

[2]:
```
df_results_2021 = pd.read_sql("SELECT * FROM election_results_2021", conn)

df_voters_2021 = pd.read_sql("SELECT * FROM voters_2021", conn)

df_voters_2026 = pd.read_sql("SELECT * FROM voters_2026", conn)
```

1.4 Compare Voter Rolls (2021 vs 2026)

[3]:
```
df_roll_compare = (
    df_voters_2021
    .merge(
        df_voters_2026,
        on=["ac_no", "ac_name"],
        how="inner"
    )
)

df_roll_compare["removed_voters_2026"] = (
    df_roll_compare["total_eligible_voters_2021"]
    - df_roll_compare["total_eligible_voters_2026"]
)

df_roll_compare.head()
```

[3]:
```
   ac_no        ac_name  total_votes_polled_2021  total_eligible_voters_2021  \
0      1  Gummidipoondi                   222069                      281688
1      2        Ponneri                   210354                      267368
2      3      Tiruttani                   232624                      291336
```

```
3      4    Thiruvallur                    214243                       274982
4      5    Poonamallee                    263736                       358218

   not_voted_2021  total_eligible_voters_2026  removed_voters_2026
0           59619                      242558                39130
1           57014                      239141                28227
2           58712                      252154                39182
3           60739                      238818                36164
4           94482                      348034                10184
```

1.5 Attach Election Outcome Data

Victory margin from the 2021 election is now joined with voter roll changes to assess **outcome sensitivity**.

[4]:
```python
df_analysis = (
    df_results_2021
    .merge(
        df_roll_compare,
        on=["ac_no", "ac_name"],
        how="inner"
    )
)

df_analysis.head()
```

[4]:
```
   ac_no        ac_name winner_party  votes_winner runnerup_party  \
0      1  Gummidipoondi          DMK        126452            PMK
1      2        Ponneri          INC         94528           ADMK
2      3      Tiruttani          DMK        120314           ADMK
3      4    Thiruvallur          DMK        107709           ADMK
4      5    Poonamallee          DMK        149578            PMK

   votes_runnerup  victory_margin  total_votes_polled_2021  \
0           75514           50938                   222069
1           84839            9689                   210354
2           91061           29253                   232624
3           85008           22701                   214243
4           55468           94110                   263736

   total_eligible_voters_2021  not_voted_2021  total_eligible_voters_2026  \
0                      281688           59619                      242558
1                      267368           57014                      239141
2                      291336           58712                      252154
3                      274982           60739                      238818
4                      358218           94482                      348034
```

```
   removed_voters_2026
0                39130
1                28227
2                39182
3                36164
4                10184
```

1.6 Core Impact Tests

Two key comparisons are performed:

1. **Removed Voters vs Victory Margin**
   - Determines if roll changes exceed the threshold that decided the election.
2. **Removed Voters vs Not Voted (2021)**
   - Provides historical inactivity context.

[5]:
```
df_analysis["removed_gt_margin"] = (
    df_analysis["removed_voters_2026"] > df_analysis["victory_margin"]
)

df_analysis["removed_gt_not_voted"] = (
    df_analysis["removed_voters_2026"] > df_analysis["not_voted_2021"]
)
```

1.7 Risk Classification

Each constituency is classified based on the magnitude of voter roll change relative to:

- victory margin
- historical non-participation

This creates a **numerical risk categorization**, not a judgement.

1.7.1 Risk Level Explanation

- **Low Impact**
  Removed voters are smaller than the victory margin.
  → Roll changes are unlikely to affect the result.
- **Moderate Impact**
  Removed voters exceed past non-voters but not the victory margin.
  → Changes are noticeable but still below the outcome threshold.
- **High Impact**
  Removed voters are greater than the victory margin.
  → The constituency becomes sensitive, and the result could potentially change.
- **Critical Impact**
  Removed voters exceed both the victory margin and historical non-voters.
  → Highest sensitivity, where roll changes are large enough to matter significantly.

- **No Removal / Increase**
  Voter count increased or remained stable.
  → No risk from voter roll changes.

[6]:
```
def classify_risk(row):
    if row["removed_voters_2026"] <= 0:
        return "No Removal / Increase"
    if row["removed_gt_margin"] and row["removed_gt_not_voted"]:
        return "Critical Impact"
    if row["removed_gt_margin"]:
        return "High Impact"
    if row["removed_gt_not_voted"]:
        return "Moderate Impact"
    return "Low Impact"

df_analysis["risk_level"] = df_analysis.apply(classify_risk, axis=1)
```

1.8 Final Analytical Dataset

This table represents the **final output** of the project and is used for reporting, visualization, and SQL storage.

[7]:
```
df_final_analysis = df_analysis[
    [
        "ac_no",
        "ac_name",
        "winner_party",
        "runnerup_party",
        "victory_margin",
        "total_eligible_voters_2021",
        "total_eligible_voters_2026",
        "removed_voters_2026",
        "not_voted_2021",
        "risk_level"
    ]
].sort_values(
    "removed_voters_2026",
    ascending=False
)

df_final_analysis.head()
```

[7]:
```
    ac_no           ac_name winner_party runnerup_party  victory_margin  \
26     27  Shozhinganallur          DMK           ADMK           35405
29     30       Pallavaram          DMK           ADMK           37783
6       7      Maduravoyal          DMK           ADMK           31721
25     26        Velachery          INC           ADMK            4352
20     21       Anna Nagar          DMK           ADMK           27445
```

```
    total_eligible_voters_2021  total_eligible_voters_2026  \
26                      698893                      484011
29                      436239                      294712
6                       452195                      325407
25                      314578                      189999
20                      286090                      162135

    removed_voters_2026  not_voted_2021       risk_level
26               214882          310538      High Impact
29               141527          170049      High Impact
6                126788          178325      High Impact
25               124579          137864      High Impact
20               123955          120994  Critical Impact
```

1.9 Key Insights

- Constituencies where **removed voters exceed victory margin** are numerically sensitive to roll revisions.
- Comparing removals with **not voted (2021)** prevents misleading conclusions.
- The analysis highlights **outcome sensitivity**, not intent.

[8]:
```python
df_final_analysis.to_sql(
    "voter_roll_impact_analysis",
    conn,
    if_exists="replace",
    index=False
)
```

[8]: 232

1.10 Visual Impact Analysis

[9]:
```python
import matplotlib.pyplot as plt
import numpy as np
```

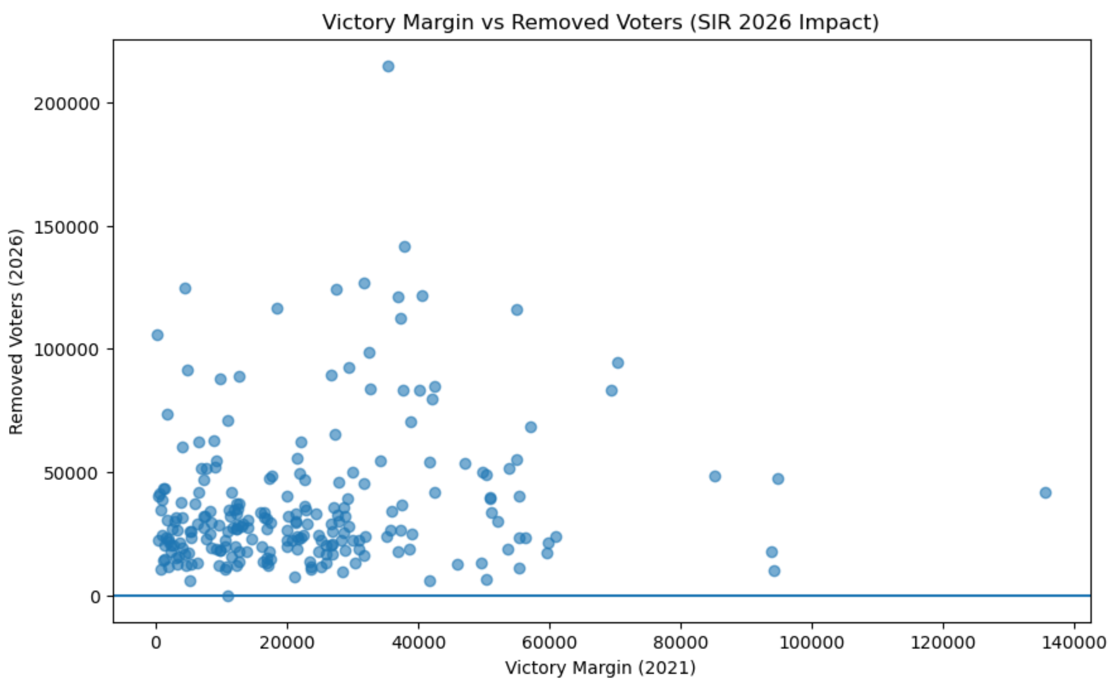
1.10.1 Victory Margin vs Removed Voters (Core Sensitivity Plot)

This scatter plot visualizes whether voter roll changes exceed the margin that decided the 2021 election.

- X-axis: Victory Margin (2021)
- Y-axis: Removed Voters (2026)
- Points above the horizontal axis indicate voter removal

Constituencies where removed voters exceed the victory margin are numerically sensitive to roll revisions.

[10]:
```python
plt.figure(figsize=(10, 6))

plt.scatter(
    df_final_analysis["victory_margin"],
    df_final_analysis["removed_voters_2026"],
    alpha=0.6
)

plt.axhline(y=0)
plt.xlabel("Victory Margin (2021)")
plt.ylabel("Removed Voters (2026)")
plt.title("Victory Margin vs Removed Voters (SIR 2026 Impact)")

plt.show()
```



1.10.2 How to read the chart

- Points above the zero line indicate constituencies where voters were removed
- Further right → constituency was won by a larger margin in 2021
- Higher up → larger voter roll reduction in 2026

1.10.3 Direct Takeaway from the above chart

Key observation:
> In many constituencies, the number of voters removed is higher than the margin that decided

the election.

What this means:
> When more voters are removed than the victory margin, the result could potentially change, especially since the total number of voters did not increase significantly from 2021 to 2025 to balance these removals.

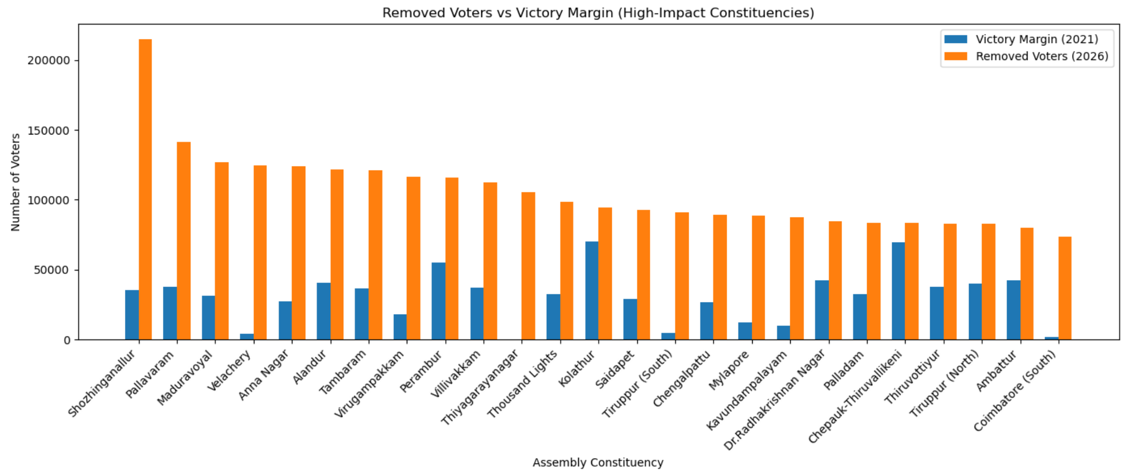
1.10.4 High-Impact Constituencies: Removed Voters vs Victory Margin

This clustered bar chart compares removed voters and victory margins **side-by-side** for constituencies classified as high or critical impact.

This visualization clearly shows where voter roll changes exceed the threshold that decided the election.

[11]:
```python
high_impact = df_final_analysis[
    df_final_analysis["risk_level"].isin(
        ["High Impact", "Critical Impact"]
    )
].sort_values(
    "removed_voters_2026",
    ascending=False
).head(25)

x = np.arange(len(high_impact))
width = 0.35

plt.figure(figsize=(14, 6))

plt.bar(
    x - width/2,
    high_impact["victory_margin"],
    width,
    label="Victory Margin (2021)"
)

plt.bar(
    x + width/2,
    high_impact["removed_voters_2026"],
    width,
    label="Removed Voters (2026)"
)

plt.xticks(
    x,
    high_impact["ac_name"],
    rotation=45,
    ha="right"
)
```

```
plt.xlabel("Assembly Constituency")
plt.ylabel("Number of Voters")
plt.title("Removed Voters vs Victory Margin (High-Impact Constituencies)")
plt.legend()
plt.tight_layout()
plt.show()
```



1.10.5 Direct take away for the above chart

Key observation:
In these high-impact constituencies, the number of voters removed in 2026 is consistently higher than the victory margin from the 2021 election.

What this means:
> When the bar representing removed voters is taller than the victory margin bar, the number of voters removed is large enough to potentially influence the election result, especially since there was no significant increase in voter count between 2021 and 2025 to offset these removals.

Why this chart matters:
> This side-by-side comparison makes it easy to see which constituencies are more sensitive, as the scale of voter roll changes clearly exceeds the margin that decided the election.

[12]:
```
total_critical_high_impact = high_impact = df_final_analysis[
    df_final_analysis["risk_level"].isin(
        ["High Impact", "Critical Impact"]
    )
]
total_critical_high_impact.head(15)
```

[12]:
```
    ac_no          ac_name winner_party runerup_party  victory_margin  \
26     27   Shozhinganallur          DMK          ADMK           35405
```

29	30	Pallavaram	DMK	ADMK	37783
6	7	Maduravoyal	DMK	ADMK	31721
25	26	Velachery	INC	ADMK	4352
20	21	Anna Nagar	DMK	ADMK	27445
27	28	Alandur	DMK	ADMK	40571
30	31	Tambaram	DMK	ADMK	36824
21	22	Virugampakkam	DMK	ADMK	18367
11	12	Perambur	DMK	ADMK	54976
13	14	Villivakkam	DMK	ADMK	37237
23	24	Thiyagarayanagar	DMK	ADMK	137
19	20	Thousand Lights	DMK	BJP	32462
12	13	Kolathur	DMK	ADMK	70384
22	23	Saidapet	DMK	ADMK	29408
111	114	Tiruppur (South)	DMK	ADMK	4709

	total_eligible_voters_2021	total_eligible_voters_2026	\
26	698893	484011	
29	436239	294712	
6	452195	325407	
25	314578	189999	
20	286090	162135	
27	389118	267271	
30	415663	294355	
21	291691	175123	
11	315942	200181	
13	255328	142706	
23	245040	139498	
19	240096	141393	
12	281197	186841	
22	279033	186489	
111	276483	185282	

	removed_voters_2026	not_voted_2021	risk_level
26	214882	310538	High Impact
29	141527	170049	High Impact
6	126788	178325	High Impact
25	124579	137864	High Impact
20	123955	120994	Critical Impact
27	121847	151367	High Impact
30	121308	166695	High Impact
21	116568	122604	High Impact
11	115761	115555	Critical Impact
13	112622	111236	Critical Impact
23	105542	106926	High Impact
19	98703	104156	High Impact
12	94356	107809	High Impact
22	92544	118706	High Impact

```
111                  91201          102065      High Impact
```

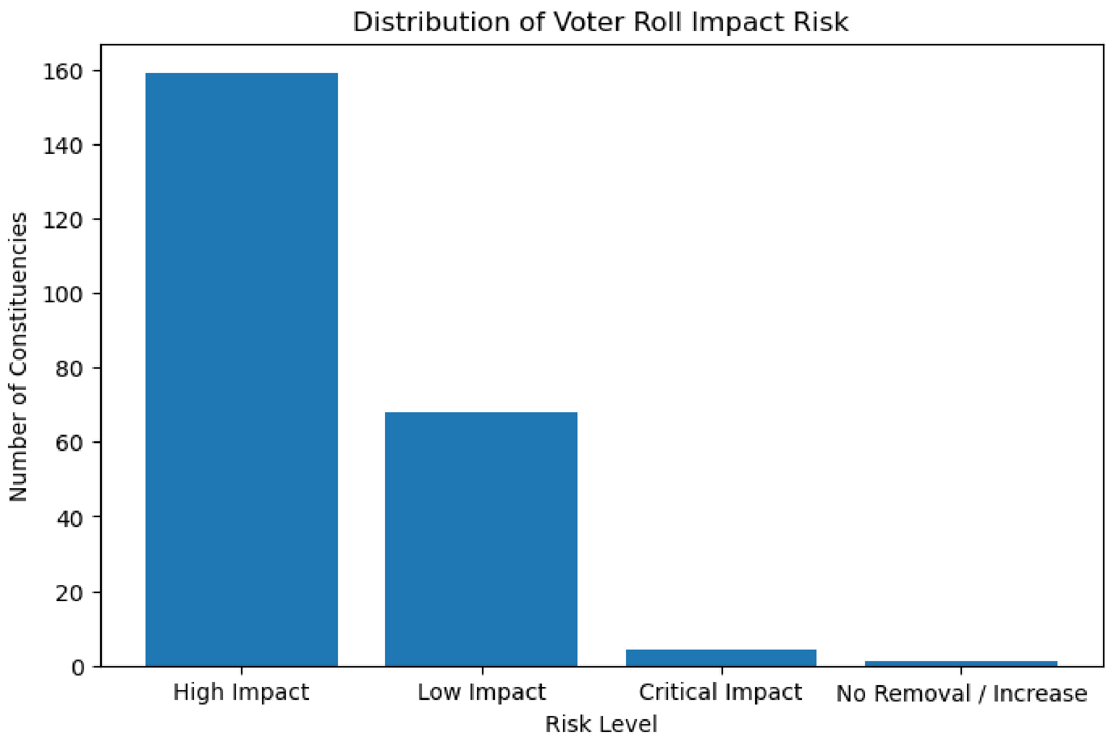
1.10.6 Distribution of Risk Levels

This chart summarizes how many constituencies fall into each risk category, providing a high-level overview of impact spread across the state.

1.10.7 Risk Level Explanation

- **Low Impact**
  Removed voters are smaller than the victory margin.
  → Roll changes are unlikely to affect the result.
- **Moderate Impact**
  Removed voters exceed past non-voters but not the victory margin.
  → Changes are noticeable but still below the outcome threshold.
- **High Impact**
  Removed voters are greater than the victory margin.
  → The constituency becomes sensitive, and the result could potentially change.
- **Critical Impact**
  Removed voters exceed both the victory margin and historical non-voters.
  → Highest sensitivity, where roll changes are large enough to matter significantly.
- **No Removal / Increase**
  Voter count increased or remained stable.
  → No risk from voter roll changes.

[13]:
```
risk_counts = df_final_analysis["risk_level"].value_counts()

plt.figure(figsize=(8, 5))

plt.bar(
    risk_counts.index,
    risk_counts.values
)

plt.xlabel("Risk Level")
plt.ylabel("Number of Constituencies")
plt.title("Distribution of Voter Roll Impact Risk")

plt.show()
```



1.10.8 Winning majority context

- To form a government in Tamil Nadu, **118 out of 234 seats** are required.
- When many seats are high impact, even small changes across multiple constituencies can matter at the overall result level.

1.10.9 Key Findings

- **159 out of 234 constituencies** fall under the **High Impact** category
- The winning side in 2021 secured **133 seats**, with **118 required** to form a government
- A large number of seats are numerically sensitive to voter roll changes

1.11 Logistic Regression: Identifying High-Impact Constituencies

In this section, we use **Logistic Regression** to estimate whether a constituency is **High Impact** based on voter roll changes and election-related factors.

This model does **not predict election results**.
It classifies constituencies based on whether voter roll changes **exceed the historical victory margin**.

[ ]:
```
df_ml = df_final_analysis.copy()

df_ml["high_impact_flag"] = (
```

```
    df_ml["removed_voters_2026"] > df_ml["victory_margin"]
).astype(int)

df_ml[["ac_no", "high_impact_flag"]].head()
```

[ ]:

	ac_no	high_impact_flag
26	27	1
29	30	1
6	7	1
25	26	1
20	21	1

What this does:
- Assigns 1 to constituencies where removed voters exceed the victory margin
- Assigns 0 otherwise

This directly reflects the analytical rule already used in the project.

[ ]:
```
df_ml["turnout_rate_2021"] = (
    df_ml["total_eligible_voters_2021"]
    - df_ml["not_voted_2021"]
) / df_ml["total_eligible_voters_2021"]

df_ml["roll_change_pct"] = (
    df_ml["removed_voters_2026"]
    / df_ml["total_eligible_voters_2021"]
)
```

[ ]:
```
features = [
    "victory_margin",
    "removed_voters_2026",
    "not_voted_2021",
    "turnout_rate_2021",
    "roll_change_pct"
]

X = df_ml[features]
y = df_ml["high_impact_flag"]
```

Features used: - Victory margin (how close the election was) - Removed voters (scale of roll change) - Not voted (historical inactivity) - Turnout rate - Roll change percentage

All features are interpretable and grounded in data.

[17]:
```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.25,
```

```
    random_state=42,
    stratify=y
)
```

We split the data into training and testing sets to evaluate the model fairly.

[18]:
```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
```

[19]:
```
pipeline = Pipeline(steps=[
    ("scaler", StandardScaler()),
    ("log_reg", LogisticRegression(max_iter=1000))
])

pipeline
```

[19]:
```
Pipeline(steps=[('scaler', StandardScaler()),
                ('log_reg', LogisticRegression(max_iter=1000))])
```

[20]:
```
pipeline.fit(X_train, y_train)
```

[20]:
```
Pipeline(steps=[('scaler', StandardScaler()),
                ('log_reg', LogisticRegression(max_iter=1000))])
```

[21]:
```
from sklearn.metrics import accuracy_score, confusion_matrix,␣
 ↪classification_report

y_pred = pipeline.predict(X_test)
```

[22]:
```
accuracy = accuracy_score(y_test, y_pred)
confusion_matrix(y_test, y_pred)
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.94      0.97        17
           1       0.98      1.00      0.99        41

    accuracy                           0.98        58
   macro avg       0.99      0.97      0.98        58
weighted avg       0.98      0.98      0.98        58
```

How to read the results: - Accuracy shows overall classification performance - Confusion matrix shows correct vs incorrect classifications - Precision & recall indicate how well High Impact seats are identified

[ ]:
```
df_ml["high_impact_probability"] = pipeline.predict_proba(X)[:, 1]

df_ml[
    [
        "ac_no",
        "ac_name",
        "victory_margin",
        "removed_voters_2026",
        "high_impact_probability"
    ]
].sort_values(
    "high_impact_probability",
    ascending=False
).head(10)
```

[ ]:
```
     ac_no           ac_name  victory_margin  removed_voters_2026  \
26      27   Shozhinganallur           35405               214882
25      26         Velachery            4352               124579
23      24  Thiyagarayanagar             137               105542
21      22      Virugampakkam           18367               116568
20      21        Anna Nagar           27445               123955
111    114  Tiruppur (South)            4709                91201
29      30        Pallavaram           37783               141527
24      25          Mylapore           12633                88976
6        7        Maduravoyal           31721               126788
13      14        Villivakkam           37237               112622

     high_impact_probability
26                  1.000000
25                  1.000000
23                  1.000000
21                  1.000000
20                  0.999999
111                 0.999999
29                  0.999997
24                  0.999995
6                   0.999995
13                  0.999993
```

Why this matters: Instead of a hard yes/no label, we now have a **probability score** that shows how strongly a constituency belongs to the High Impact category.

[24]:
```
coefficients = pipeline.named_steps["log_reg"].coef_[0]

coef_df = pd.DataFrame({
    "feature": features,
    "coefficient": coefficients
}).sort_values(
```

```
    "coefficient",
    ascending=False
)

coef_df
```

[24]:

	feature	coefficient
1	removed_voters_2026	1.965732
4	roll_change_pct	1.908149
2	not_voted_2021	0.848723
3	turnout_rate_2021	0.284034
0	victory_margin	-3.511599

1.11.1 Interpretation of Logistic Regression Coefficients

The table above shows the coefficients from the logistic regression model, which indicate how each feature influences the likelihood of a constituency being classified as **High Impact**.

- **Positive coefficients** increase the likelihood of a constituency being High Impact.
- **Negative coefficients** decrease the likelihood.
- The **larger the absolute value**, the stronger the influence of that feature.

Key interpretations: - **Removed Voters (2026)** has the strongest positive effect, indicating that higher voter removals significantly increase impact sensitivity. - **Roll Change Percentage** also has a strong positive influence, showing that relative roll reductions matter in addition to absolute numbers. - **Not Voted (2021)** has a moderate positive effect, providing contextual support but not outweighing large roll changes. - **Turnout Rate (2021)** has a smaller influence compared to other factors. - **Victory Margin (2021)** has a strong negative effect, meaning constituencies with larger winning margins are less likely to be High Impact.

Overall takeaway:
Voter roll removals and roll change size are the most influential factors in determining impact sensitivity, while larger victory margins significantly reduce this risk.

[25]:

```
final_sensitive_constituencies = df_ml[
    (df_ml["risk_level"] == "High Impact") &
    (df_ml["high_impact_probability"] >= 0.7)
][
    [
        "ac_no",
        "ac_name",
        "victory_margin",
        "removed_voters_2026",
        "high_impact_probability"
    ]
].sort_values(
    "removed_voters_2026",
    ascending=False
)
```

```
final_sensitive_constituencies.head()
```

[25]:
```
    ac_no           ac_name  victory_margin  removed_voters_2026  \
26     27  Shozhinganallur           35405               214882
29     30        Pallavaram           37783               141527
6       7       Maduravoyal           31721               126788
25     26         Velachery            4352               124579
27     28           Alandur           40571               121847

    high_impact_probability
26                 1.000000
29                 0.999997
6                  0.999995
25                 1.000000
27                 0.999968
```

[26]:
```
final_sensitive_constituencies.to_csv(
    "../result data/high_impact_constituencies.csv",
    index=False
)
```

1.12 Conclusion: Visual Analysis and Logistic Regression

The visual analysis shows that in many constituencies, the number of voters removed in 2026 is higher than the victory margin from the 2021 election. This indicates that these constituencies are numerically sensitive, as voter roll changes are large enough to potentially matter in close contests.

The logistic regression model supports this observation by quantifying the same pattern. It shows that voter roll removals and roll change percentage are the strongest factors increasing the likelihood of a constituency being classified as High Impact, while larger victory margins significantly reduce this likelihood.

Together, the plots and the model lead to a consistent conclusion:
> When voter roll changes exceed historical victory margins, constituencies become more outcome-sensitive, especially in the absence of significant voter growth.

This analysis does not claim that election results have changed, but it highlights where accurate and careful voter roll revision is especially important in closely contested constituencies.