1️⃣ What is the model's accuracy?

From my experiments:

- Accuracy ranges from **71.2% (k=1)** to **~79.0% (k=9–15)**
- **Best accuracy:**
    - **k = 15 → 78.99%**
    - Very close to **k = 9 → 78.92%**

Interpretation:

The KNN model correctly predicts churn for roughly **79% of customers**.

This is a **solid baseline**, but accuracy alone is not enough for churn problems.

2️⃣ Which K is actually "best"?

Let's look at what matters.

Metric comparison (churn = positive class)

k	Accuracy	Precision	Recall	F1-score
1	0.712	0.457	0.455	0.456
3	0.762	0.566	0.449	0.501
5	0.766	0.579	0.430	0.494
7	0.781	0.626	0.439	0.516
9	**0.789**	**0.656**	**0.433**	**0.522**
11	0.787	0.661	0.406	0.503
15	0.790	0.674	0.404	0.505

Key observations

- **Precision increases steadily** with higher k
- **Recall decreases** as k increases
- **F1-score peaks at k = 9**

Final choice

Best K = 9, because it gives the **highest F1-score (0.522)**, balancing precision and recall.

3 What features seem most important?

Based on my exploratory data analysis and how KNN separates customers:

◇ Contract type

- Month-to-month customers churn far more than long-term contracts
- This aligns with KNN relying heavily on categorical proximity

◇ Tenure

- Short-tenure customers cluster strongly with churners
- Longer tenure pushes customers closer to non-churn neighbors

◇ MonthlyCharges

- Higher charges increase churn probability
- Especially influential when combined with month-to-month contracts

◇ TotalCharges

- Low total charges (often low tenure) correlate strongly with churn

◇ Service add-ons

- Lack of OnlineSecurity / TechSupport correlates with churn
- These features influence neighborhood similarity in KNN

Summary:

Contract length, tenure, monthly charges, and service add-ons are the strongest drivers of churn in my model.

4️⃣ What would you recommend to the company?

Based on **my best model (k = 9)**:

🎯 Recommendation 1: Prioritize recall-sensitive segments

- Recall is only **~43%**
- Many churners are still missed
- Focus on:
  - New customers
  - Month-to-month contracts
  - High MonthlyCharges

🎯 Recommendation 2: Incentivize contract upgrades

- Push month-to-month users to 1–2 year contracts
- This is the single strongest churn reducer in my data

🎯 Recommendation 3: Bundle retention services

- Offer OnlineSecurity / TechSupport to at-risk customers
- These services shift customers closer to “non-churn” clusters

🎯 Recommendation 4: Use the model for targeted, not blanket, actions

- Precision at k=9 is **~66%**
- Two out of three flagged customers are true churners
- Use targeted retention campaigns instead of mass discounts

5️⃣ What are the limitations of *your* model?

Now grounded in my results:

⚠️ Limitation 1: Low recall

- Best recall is **~45% (k=1–3)**
- Even the best balanced model (k=9) misses more than half of churners

⚠️ Limitation 2: Accuracy vs recall trade-off

- Higher accuracy comes at the cost of recall
- The model favors "playing safe" and predicting non-churn

⚠️ Limitation 3: KNN scalability

- KNN requires comparing each prediction to all training points
- Not suitable for large-scale production systems

⚠️ Limitation 4: Limited explainability

- No direct feature importance

- Hard to justify individual predictions to business stakeholders

⚠️ Limitation 5: No probability outputs by default

- Hard to tune decision thresholds
- Limits fine-grained risk scoring

Final conclusion (clean, submission-ready)

The KNN model achieves close to **79% accuracy**, with the best overall performance at **k = 9**, where the F1-score is highest.

While the model provides useful segmentation and highlights key churn drivers such as contract type, tenure, pricing, and service bundles, it struggles to identify all churners due to relatively low recall.

As a result, the model is best suited as a **baseline or exploratory tool**. For production use, a more scalable and explainable model with better recall, such as Logistic Regression or Gradient Boosting, would be recommended.