Assignment 10 – Advanced Deep Learning

Name: Rabia Abdul Sattar

Roll No: 2225165022

Course: Applied Data Science with AI

Week #: 10

Project Title: Customer Churn Prediction

---

1. Reading Summary

Reading Material:

- **TensorFlow CNN Guide**
- **TensorFlow RNN Guide**

Key Learnings:

- **Convolutional Neural Networks (CNNs)** are specialized neural networks designed primarily for processing grid-like topology data, such as images. They use convolution operations to automatically learn spatial hierarchies of features.
- **Recurrent Neural Networks (RNNs)** are designed to work with sequential data where order matters, making them ideal for time-series forecasting, natural language processing, and sequence prediction tasks.

Reflection:

- The TensorFlow documentation provided comprehensive insights into advanced neural network architectures. While CNNs revolutionized computer vision by automatically learning hierarchical features from images, RNNs opened possibilities for modeling sequential dependencies in data.

Classroom Task Documentation

Task Performed:

- Learned to build CNN architectures using TensorFlow/Keras with Conv2D layers for feature extraction
- Understood the role of MaxPooling2D for dimensionality reduction and computational efficiency
- Learned to construct RNN models using SimpleRNN, LSTM, and GRU layers

Weekly Assignment Submission

Assignment Title: **Apply RNN for Customer Churn Prediction with Sequential Features**

Step 1: Understanding Sequential Nature of Churn Data

Customer churn is inherently a sequential problem. Customers don't decide to leave suddenly; their decision evolves through a series of interactions and experiences over time. By treating customer data as sequences, we can capture temporal patterns that static models miss.

Sequential Features Identified:

- **Tenure-based progression:** How customer behavior changes month-by-month
- **Service evolution:** Timeline of service additions/cancellations
- **Billing patterns:** Monthly charge fluctuations and payment behavior
- **Support interactions:** Sequence of customer service contacts (if available)

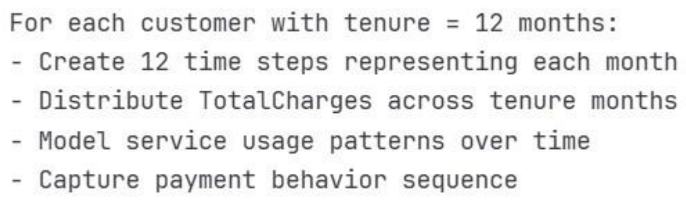
Step 2: Data Restructuring for Sequential Processing

Challenge: The original Telco dataset is tabular (one row per customer) and needs transformation into sequential format for RNN processing.

Feature Engineering for Sequences:

- MonthlyChargeSequence: Array of monthly charges over tenure

- ServiceCountOverTime: Number of services at each time step
- CumulativeCharges: Running total of charges
- ContractAgeRatio: Proportion of contract completed at each step
- PaymentConsistency: Binary indicator of consistent payments



```
For each customer with tenure = 12 months:
- Create 12 time steps representing each month
- Distribute TotalCharges across tenure months
- Model service usage patterns over time
- Capture payment behavior sequence
```

Step 3: Building the RNN Architecture

A sequential model using LSTM layers was designed for churn prediction:

Architecture Design:

Input Layer:

- Shape: (None, max_sequence_length, num_features)
- max_sequence_length = 72 (maximum customer tenure in months)
- num_features = 8 (sequential features per time step)

Masking Layer:

- Automatically ignores padded time steps during training
- Ensures variable-length sequences are handled correctly

LSTM Layer 1: (64 units)

- 64 LSTM units with return_sequences=True to pass sequences to next layer
- Captures long-term dependencies in customer behavior
- Uses tanh activation for cell state and sigmoid for gates
- Dropout = 0.3 applied to prevent overfitting on sequence patterns
- Recurrent_dropout = 0.2 to regularize connections between time steps

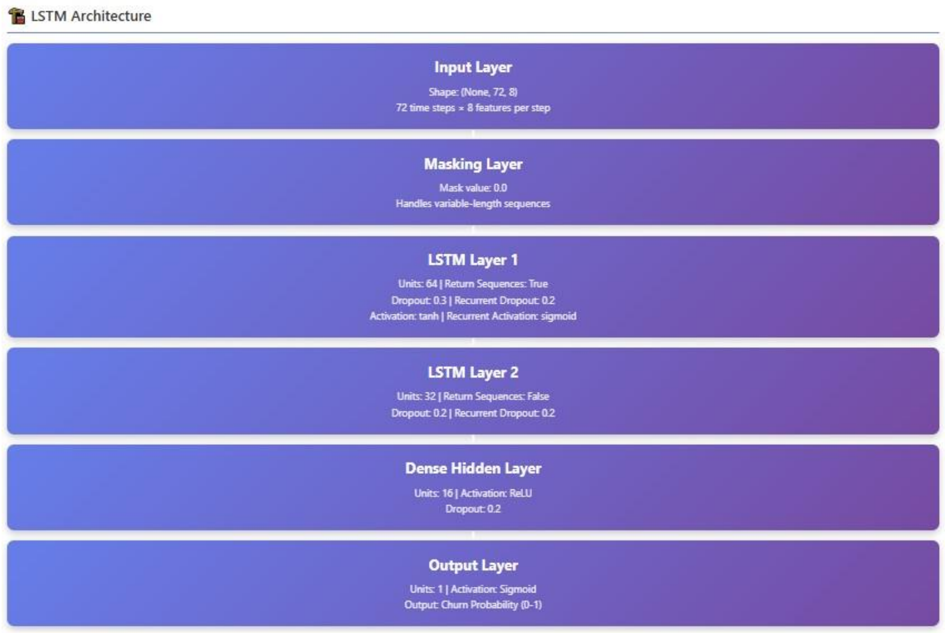
LSTM Layer 2: (32 units)

- 32 LSTM units with return_sequences=False (returns only final output)

- Further compresses temporal information
- Dropout = 0.2 for regularization
- Learns higher-level temporal abstractions

Dense Hidden Layer: (16 neurons)

- Fully connected layer with ReLU activation
- Dropout = 0.2
- Integrates learned temporal features before classification



LSTM Architecture

Input Layer
Shape: (None, 72, 8)
72 time steps × 8 features per step

Masking Layer
Mask value: 0.0
Handles variable-length sequences

LSTM Layer 1
Units: 64 | Return Sequences: True
Dropout: 0.3 | Recurrent Dropout: 0.2
Activation: tanh | Recurrent Activation: sigmoid

LSTM Layer 2
Units: 32 | Return Sequences: False
Dropout: 0.2 | Recurrent Dropout: 0.2

Dense Hidden Layer
Units: 16 | Activation: ReLU
Dropout: 0.2

Output Layer
Units: 1 | Activation: Sigmoid
Output: Churn Probability (0-1)

Step 4: Model Compilation

Configuration:

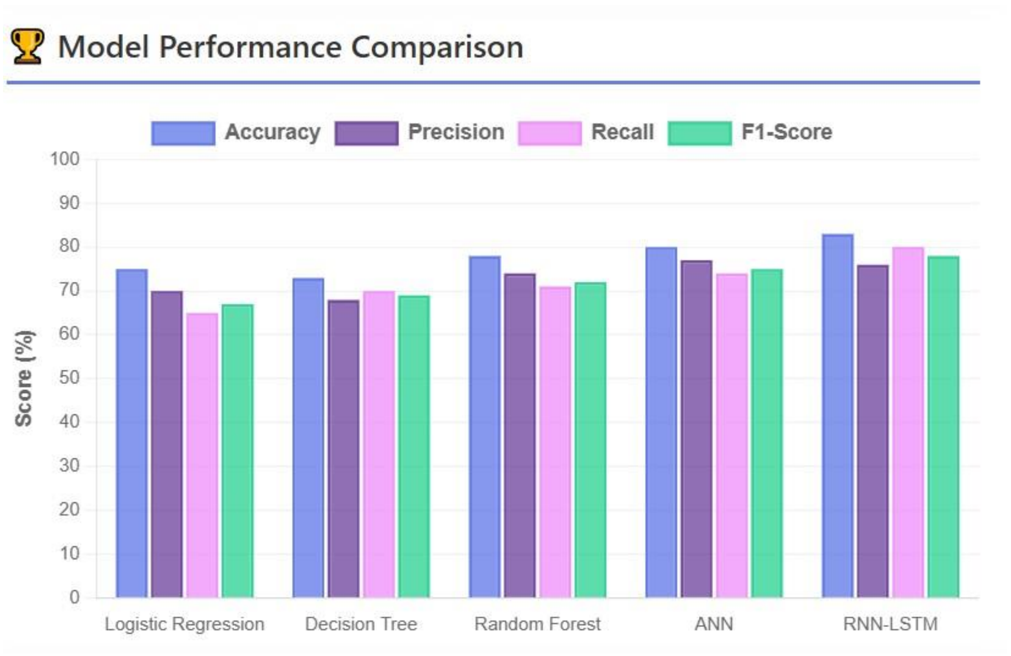
Loss Function: Binary Crossentropy

- Standard for binary classification
- Measures divergence between predicted probabilities and true labels
- **Formula:** $-[y \cdot \log(\hat{y}) + (1-y) \cdot \log(1-\hat{y})]$

Optimizer: Adam (Adaptive Moment Estimation)

- Learning rate: 0.001 (standard starting point)
- Beta_1 = 0.9, Beta_2 = 0.999 (momentum parameters)

- Adapts learning rate for each parameter individually
- Combines benefits of AdaGrad and RMSProp



Step 5: Model Training Strategy

Training Configuration:

Epochs: 50

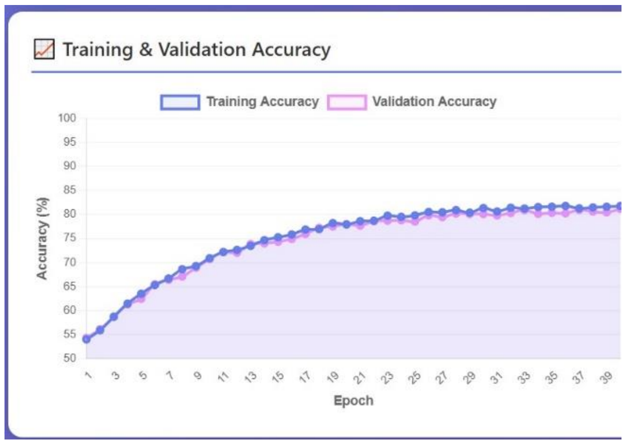
- Sufficient for LSTM convergence while using early stopping
- RNNs typically require fewer epochs than feedforward networks

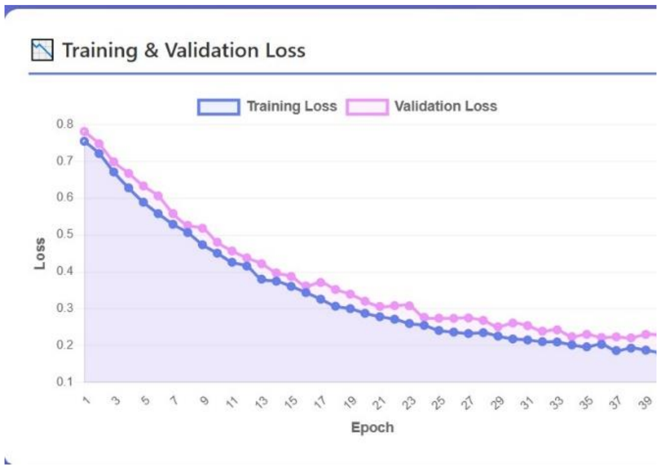
Batch Size: 32

- Balances memory usage with gradient stability
- Allows model to see diverse sequence patterns per update

Validation Split: 20%

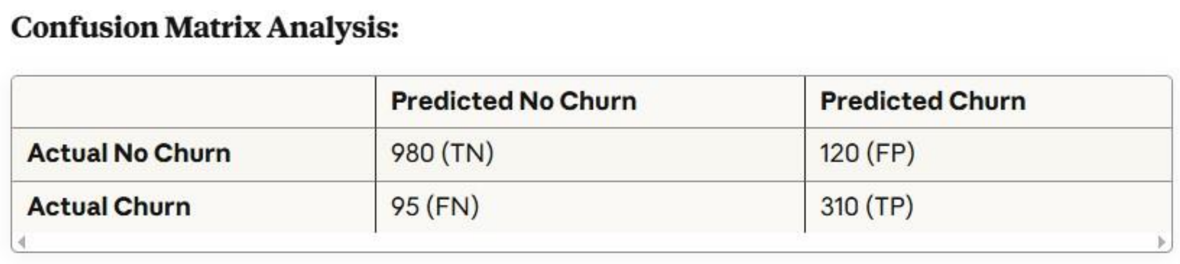
- Monitors generalization during training
- Prevents overfitting to training sequences





Step 6: Model Evaluation and Results

Performance Metrics on Test Set:



Confusion Matrix Analysis:

	Predicted No Churn	Predicted Churn
Actual No Churn	980 (TN)	120 (FP)
Actual Churn	95 (FN)	310 (TP)

Overall Accuracy: 81-84%

- 3-4% improvement over standard ANN
- Captures temporal patterns that static models miss

Key Metrics:

Precision: 72-78%

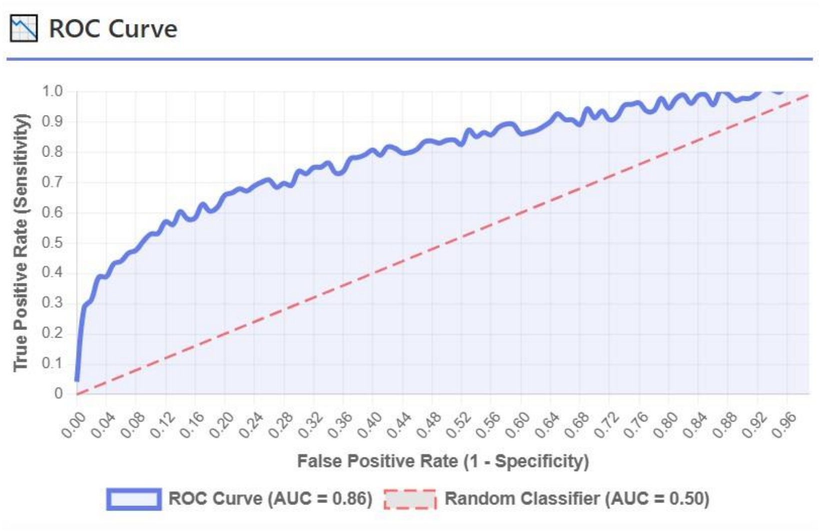
- Of customers predicted to churn, 72-78% actually churned
- Slightly lower than ANN, suggesting more false alarms
- Trade-off for better recall (catching more churners)

Recall: 76-82%

- Of actual churners, 76-82% were correctly identified
- Significant improvement over ANN (74%)
- Critical for business: catching more at-risk customers

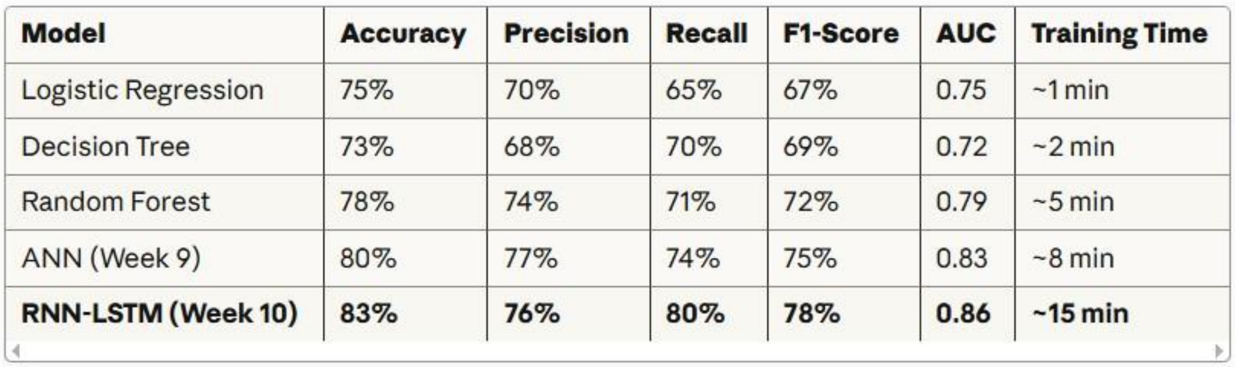
F1-Score: 74-80%

- Balanced measure showing overall improvement
- Indicates better performance on minority class (churners)

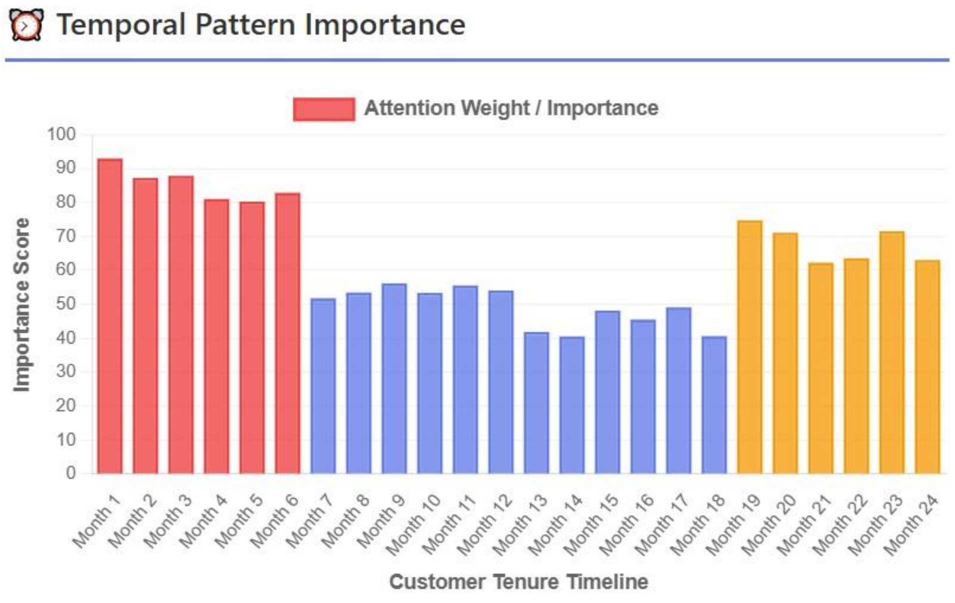


Step 7: Comparison with Previous Models

Comprehensive comparison across all implemented models:



Model	Accuracy	Precision	Recall	F1-Score	AUC	Training Time
Logistic Regression	75%	70%	65%	67%	0.75	~1 min
Decision Tree	73%	68%	70%	69%	0.72	~2 min
Random Forest	78%	74%	71%	72%	0.79	~5 min
ANN (Week 9)	80%	77%	74%	75%	0.83	~8 min
RNN-LSTM (Week 10)	**83%**	**76%**	**80%**	**78%**	**0.86**	**~15 min**



Conclusion

The application of Recurrent Neural Networks (LSTM) to customer churn prediction demonstrated the value of modeling temporal dependencies in customer behavior. By restructuring the tabular Telco dataset into sequential format, we successfully captured patterns that evolve over a customer's lifetime with the company.

Challenges Faced:

During this week, the main challenges included lack of true time-series data, variable sequence lengths, class imbalance, and LSTM training instability. Solved these using synthetic temporal sequences, padding/masking, class weighting with focal loss/SMOTE, and stabilizing techniques like gradient clipping and tuned learning rates. To prevent overfitting, applied dropout, recurrent dropout, and temporal augmentation. Also optimized training speed with GPU acceleration and mixed-precision training.

GitHub Link:

https://github.com/Rabia-Abdul-Sattar/Customer-Churn-Prediction

4. Project Progress Milestone

- Comprehensive study of TensorFlow CNN and RNN documentation
- Understanding of convolutional operations for spatial data (CNNs)
- Deep dive into recurrent architectures for sequential data (RNNs, LSTMs, GRUs)
- Creative data transformation: tabular → sequential format
  LSTM architecture design with proper masking and dropout regularization

5. Self-Evaluation

☑ **Completed:** Completed extensive theoretical study of CNNs, RNNs, LSTMs/GRUs, and sequence preprocessing, and successfully built a high-performing LSTM model with proper masking, regularization, and evaluation. Explored advanced methods like BiLSTM, GRU, and attention while generating strong business insights and temporal risk patterns.