Plagiarism Scan Report

12%
Plagiarism

0%
Exact Match

12%
Partial Match

88%
Unique

Words	442
Characters	2879
Sentences	54
Paragraphs	52
Read Time	3 minute(s)
Speak Time	3 minute(s)

Content Checked For Plagiarism

For each model (e.g., Logistic Regression, Random Forest, etc.), it shows:
- Accuracy: Overall model accuracy.
- Precision/Recall/F1-score for:
  - Class 0: Non-churners
  - Class 1: Churners
- Support: Number of true instances for each class.

Model Accuracy F1-score (Churners - Class 1) Comments
Random Forest 0.9720 0.91 Very high accuracy and recall
XGBoost 0.9669 0.90 Almost as good as RF
Decision Tree 0.9251 0.78 Good performance
Gradient Boost 0.9192 0.74 Decent but weaker on churners
Logistic Reg. 0.7648 0.53 Moderate overall
KNN 0.6348 0.52 Poor recall on churners
SVM 0.5214 0.25 Very poor for churn detection

- Best Models:
  - Random Forest: Best recall + precision for churners.
  - XGBoost: Close second.
- Underperformers:
  - SVM: Very low recall and f1-score for churners.
  - KNN and Logistic Regression: Also poor, especially in identifying churners (Class 1).
- Comparing multiple classifiers.
- Evaluating them using both overall accuracy and class-wise performance (which is crucial for imbalanced datasets like churn).
- Concluding based on how well they identify churners, which is more important than just high accuracy (as Class 0 usually dominates).

8. Building simple neural Network

A binary classification problem (likely predicting customer churn) using TensorFlow/Keras.
- Sequential model: Stacks layers linearly.
- Layer 1: 64 neurons, ReLU activation (hidden layer).
- Layer 2: 32 neurons, ReLU activation (another hidden layer).
- Output layer: 1 neuron, Sigmoid activation (since it's binary classification).
- Optimizer: 'adam' — a common optimizer for neural nets.
- Loss function: 'binary_crossentropy' — suitable for binary classification.
- Metric: Accuracy.
- Training data: Uses X_train_resampled (probably SMOTE or other class balancing).

- Validation data: Uses original X_test, y_test.
- Trains for 10 epochs, batch size of 32.

Each epoch shows:

- Training accuracy & loss
- Validation accuracy & loss

Interpretation:

- Training accuracy improves from ~50% to ~68% over 10 epochs.
- Validation accuracy fluctuates, peaking at ~83%, and ending at ~79%.
- Validation loss also varies, showing some instability — possible overfitting or noisy validation data.
- Final accuracy on test set: 0.7910 (or ~79.10%)
- Final loss: ~1.2309

This confirms the model generalizes moderately well on unseen data.

Plot Interpretation

1. Service Score vs. Churn

- Median and distribution are very similar across both churned (1) and non-churned (0) customers.
- Suggests little to no difference in churn behaviour based on Service_Score.

2. Customer Care Agent Score vs. Churn

- Surprisingly, churned customers (1) seem to have slightly higher customer care scores.
- This could indicate that poor customer service isn't the main driver of churn — or that customers rate service high before leaving due to other reasons (like price, value, or product issues).

Matched Source

Similarity 6%

Title:Overfitting and Underfitting With Machine Learning Algorithms

Aug 12, 2019 • The goal of a good machine learning model is to generalize well from the training data to any data from the problem domain. This allows us to•...Missing: moderately | Show results with:

https://www.machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms