Project Report

Data-Driven Customer Retention: Churn Modeling using Python, Machine Learning & PostgreSQL with Data Analytics

Done by Devaprakash R

For this project, I built a complete end-to-end **churn prediction system** to identify which customers of a subscription-based service are most at risk of leaving. The workflow began with the design of a normalized **PostgreSQL database schema**—four tables linked by primary and foreign keys—to store customer demographics, account details, service usage, and churn labels for 1,500 simulated subscribers. I then extracted the raw data into pandas for cleaning, normalization, and enrichment, handling messy entries and missing values to ensure high data quality.

Next, I performed an in-depth **Exploratory Data Analysis (EDA) using Matplotlib and Seaborn** to visualize churn distributions, feature correlations, and key patterns across demographics, contract types, payment methods, and service usage.

I engineered features by one-hot encoding categorical variables and standardizing numeric fields (tenure, monthly charges, total charges).

For Modeling, I split the data into train and test sets (80/20) and trained four classifiers—Logistic Regression, Decision Tree, Random Forest, and K-Nearest Neighbours—evaluating them with confusion matrices, precision, recall, F1-score, and overall accuracy.

I then fine-tuned each model via 5-fold GridSearchCV, selecting hyperparameters that maximized cross-validated accuracy.

The tuned Random Forest emerged as the best performer for balancing precision and recall on churners. Finally, I saved the model with joblib and demonstrated how to apply it to new customer data for live churn predictions, exporting results for stakeholder review.

Tools Used

Database & SQL

- PostgreSQL (psql / pgAdmin)
- SQLAlchemy (with psycopg2-binary)

Python Environment

- Python 3.x
- Virtual environment (venv)
- Jupyter Notebook & VS Code

Data Handling & Analysis

- pandas
- NumPy
- scipy

Visualization

- matplotlib
- seaborn

Machine Learning

- scikit-learn
    - Models: Logistic Regression, Decision Tree Classifier, Random Forest Classifier, KNeighbors Classifier
    - Model selection: train_test_split, GridSearchCV, cross_val_score
    - Preprocessing: StandardScaler, get_dummies

Model Persistence & Deployment Prep

- joblib for saving/loading models
- (Optional) Streamlit / Flask for app deployment

Miscellaneous

- urllib.parse.quote_plus (for safely encoding DB passwords)
- Microsoft Excel (for preparing new customer input)

Concepts & Methodologies

- **Relational Data Modeling**: Primary keys, foreign keys, table normalization
- **SQL Data Extraction**: Writing efficient JOIN queries to assemble a unified dataset
- **Data Cleaning & Preprocessing**:
    - Normalizing inconsistent categorical values (yes/no/unknown)
    - Imputing missing numeric fields with medians
    - Parsing and coercing messy text to floats
- **Exploratory Data Analysis (EDA)**:
    - Univariate (countplots, histograms)
    - Bivariate (boxplots, KDE, stripplots)
    - Correlation heatmapping
- **Feature Engineering**:
    - One-hot encoding categorical variables
    - Z-score standardization of numeric features
- **Model Building & Evaluation**:
    - Train/test split
    - Logistic Regression, Decision Tree, Random Forest, KNN
    - Metrics: confusion matrix, precision, recall, F1-score, accuracy
- **Hyperparameter Tuning**: GridSearchCV with 5-fold cross-validation
- **Model Selection**: Balancing recall on churners vs. overall accuracy
- **Model Persistence & Deployment Prep**: Serializing with joblib; preprocessing pipeline for new data

Key Business Questions Answered

- **Which customers are most likely to churn?**
    - Probability scores from the Random Forest model highlight at-risk segments.
- **What features drive churn risk?**
    - Feature importances reveal contract type, monthly charges, tenure, and service usage as top predictors.
- **How effective are different Modeling approaches?**
    - Comparative metrics guide the choice of the best algorithm for deployment.
- **How can new customer data be scored in production?**
    - A reusable preprocessing & prediction pipeline enables live churn scoring and automated export of results.