Dataset for Customer Churn Capstone Ideas is taken from Kaggle via this link:

https://www.kaggle.com/datasets/muhammadshahidazeem/customer-churn-dataset/discussion?sort=hotness

Capstone Idea 1: Churn Prediction Using Machine Learning

Process and Understanding:

Building a classification model in Python to predict whether a customer will churn or not. Using algorithms like Logistic Regression and Random Forest, to evaluate the model with metrics like accuracy, precision, recall, and F1-score. Performing feature engineering to create new variables (i.e average support calls per month) and using techniques like cross validation to ensure good results.

Idea Description:

Data preprocessing (Pandas, NumPy).

Feature engineering (i.e creating new columns like "Support Calls per Tenure").

Machine learning (Scikit-learn).

Model evaluation (classification metrics, confusion matrix).

Visualization (Matplotlib, Seaborn).

Capstone Idea 2: Customer Segmentation Using Clustering

Process and Understanding:

Using unsupervised learning techniques like K-Means to place customers based on their behavior and demographics. We can also analyze the data to identify patterns (i.e high-risk churn segments, high-spending customers). This approach will help businesses adapt and produce targeted marketing strategies and improve customer retention.

Idea Description:

Data preprocessing (scaling, handling categorical variables).

Clustering algorithms (Scikit-learn).

Visualization (Matplotlib, Seaborn).

Capstone Idea 3: Customer Lifetime Value (CLV) Prediction

Process and Understanding:

We can predict the Customer Lifetime Value (CLV) using regression models in Python. Features such as Total Spend, Tenure, and Usage Frequency would be used to estimate how much revenue a customer will generate over time. Combining this with churn risk to identify high value customers at risk of leaving.

Idea Description:

Data preprocessing (handling missing values, encoding categorical variables).

Regression modeling (Linear Regression, Random Forest Regressor)

Visualization (Matplotlib, Seaborn).

Capstone Idea 4: Churn Prediction with Deep Learning

Process and Understanding:

We can build a deep learning model using Keras to predict churn. For this, we can use neural networks to capture complex patterns in the data and experiment with different architectures such as hyperparameter tuning. Using dropout and batch normalization methods to prevent overfitting, we will be able to create an accurate churn prediction model that can be deployed in real world business scenarios.

Idea Description:

Deep learning (Keras).

Hyperparameter tuning (GridSearchCV, RandomizedSearchCV).

Model evaluation (ROC-AUC).

Capstone Idea 5: Subscription Type Optimization

Process and Understanding:

We can better recognize the relationship between Subscription Type (i.e Standard, Basic, Premium) and churn rates. A model can be built to recommend the optimal subscription type for each customer based on their usage patterns, demographics, and churn risk. Using classification techniques, we can group customers and make personalized recommendations.

Idea Description:

Data preprocessing (encoding categorical variables, scaling).

Clustering (K-Means) or classification (Random Forest).

Recommendation systems (Collaborative filtering).

Visualization (Matplotlib, Seaborn).

Capstone Idea 6: Payment Delay Prediction and Risk Analysis

Process and Understanding:

A classification model can be built to predict whether a customer will experience a payment delay. Features such as Tenure, Usage Frequency, Support Calls, and Total Spend can be used to identify patterns that lead to payment delays. From there we can analyze the risk factors and initiate strategies to reduce late payments.

Idea Description:

Data preprocessing (handling missing values, encoding categorical variables).

Feature engineering (i.e creating "Average Payment Delay per Tenure").

Classification algorithms (Logistic Regression, Random Forest, Gradient Boosting).

Model evaluation (ROC-AUC).

Visualization (Matplotlib, Seaborn).