TELECOM CUSTOMER CHURN ANALYSIS - MODEL AND DEPLOYMENT

SIYA PANGAM - SAP_ID: 77721767513

CONTENT

01 PROBLEM STATEMENT

02 DATA SET & INSIGHTS

03 VISUALIZATION

04 MODEL EVALUATION & SELECTION

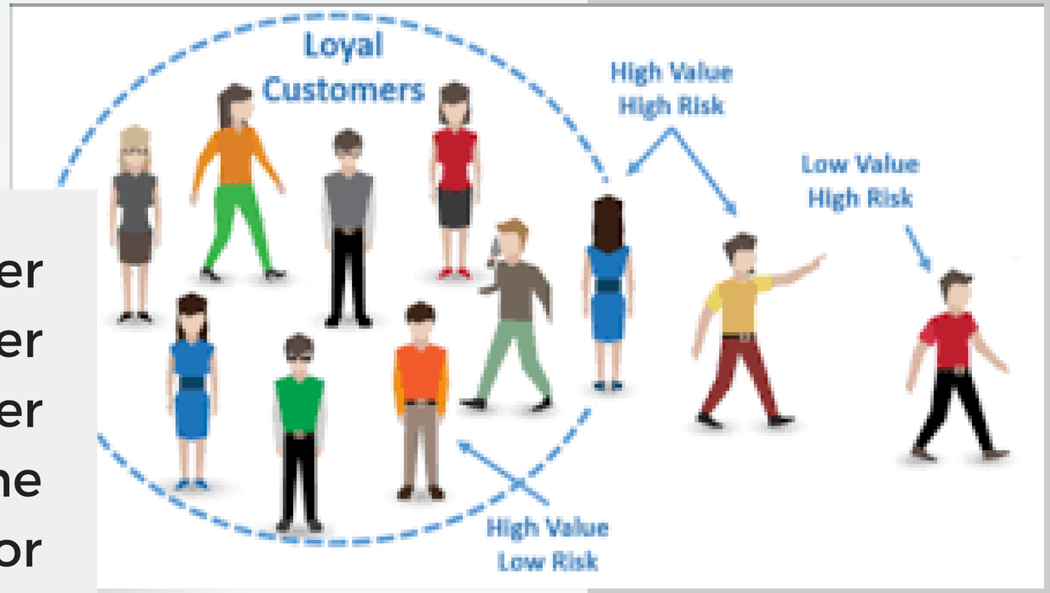
05 DEPLOYMENT

06 SOLUTIONS

PROBLEM STATEMENT

PREDICTING CUSTOMER CHURN IN A TELECOM COMPANY

A telecom company is facing a high customer churn rate and wants to reduce it. Customer churn refers to the process where a customer stops doing business with a company. In the telecom industry, customer churn is a major problem as acquiring new customers is more expensive than retaining existing customers. The company wants to use machine learning to predict which customers are likely to churn so that they can take proactive measures to retain them.



DATA SET & INSIGHTS

ABOUT DATA SET

The dataset contains information on customer demographic, account information, services used, and whether or not they have churned. The target variable is a binary column indicating whether the customer has churned (1) or not (0).

DATA SET SNIPPET

	customerID	gender	SeniorCitizen	Partner	Dependents
0	7590-VHVEG	Female	0	Yes	No
1	5575-GNVDE	Male	0	No	No
2	3668-QPYBK	Male	0	No	No
3	7795-CFOCW	Male	0	No	No

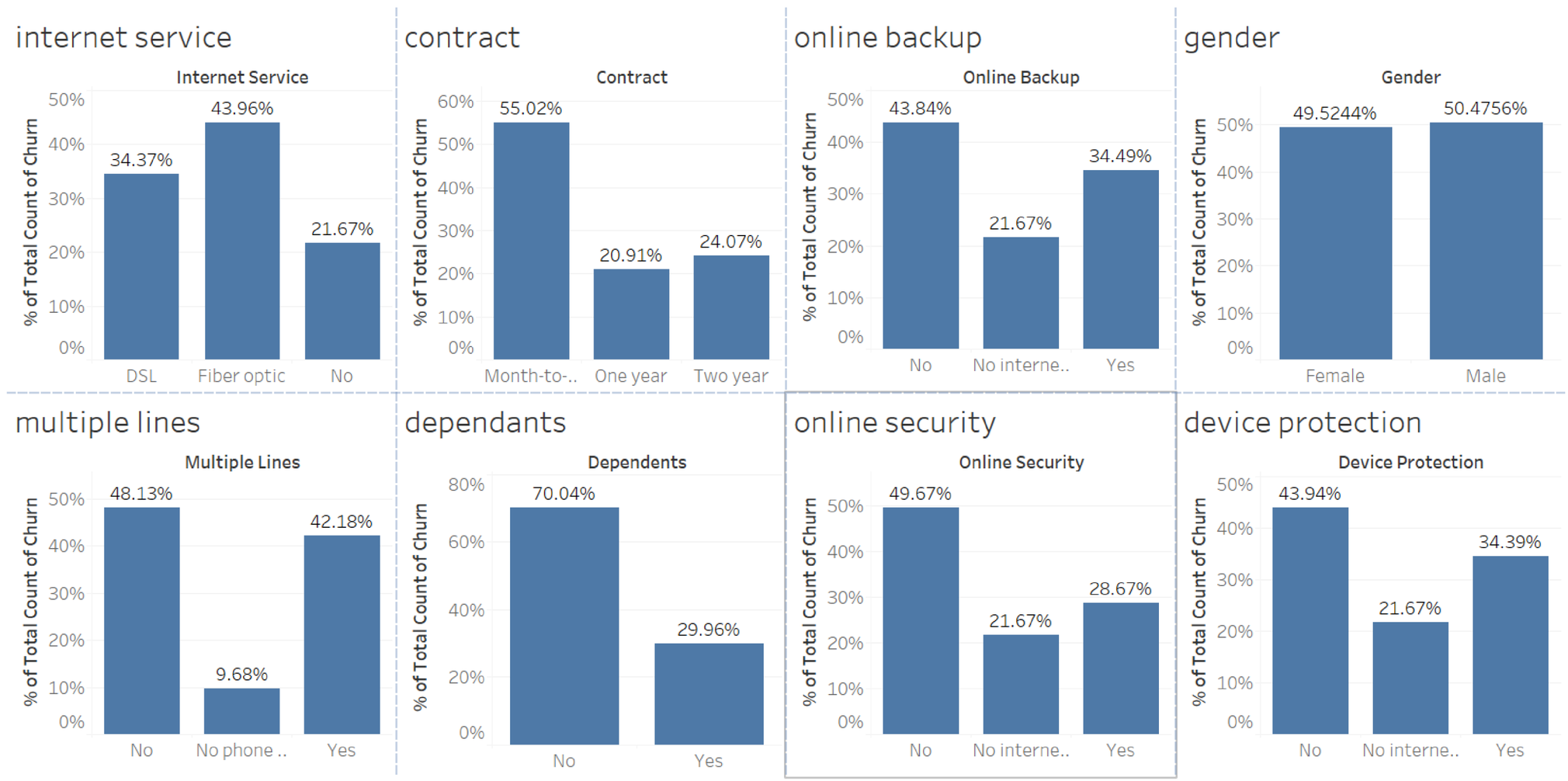
DATA SET & INSIGHTS

INSIGHTS:

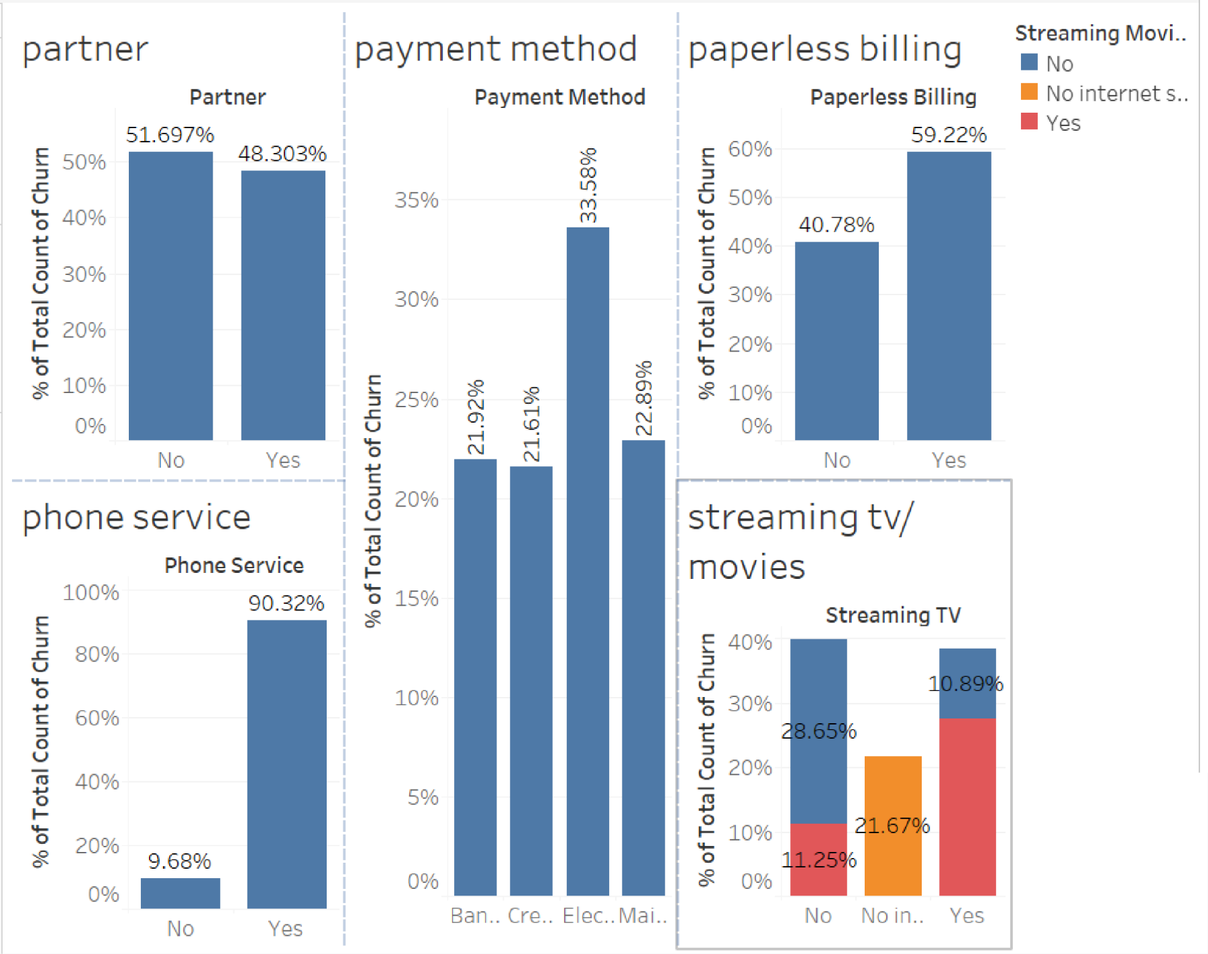
Performing exploratory data analysis and data visualizations, we can gather that :-

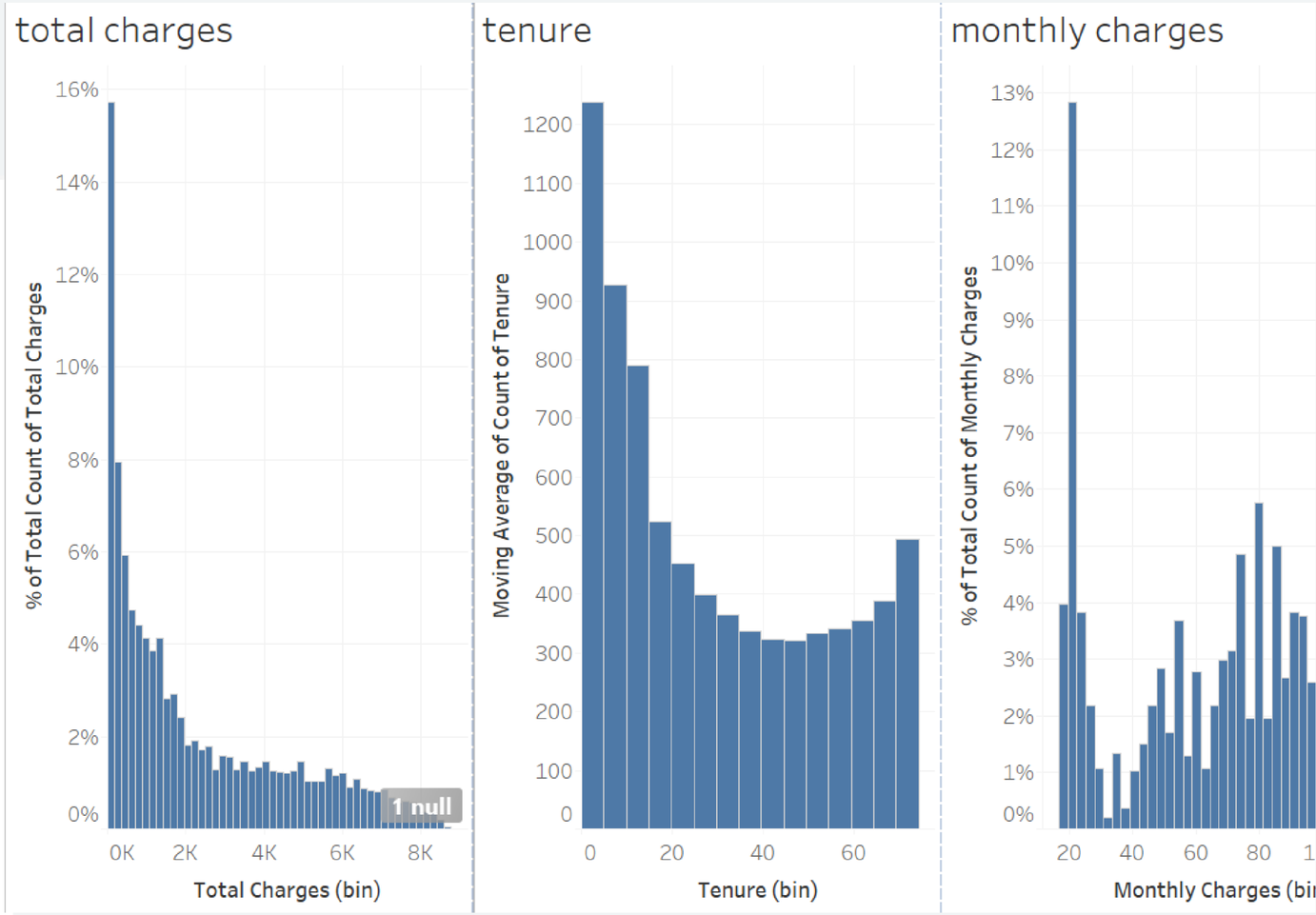
1. customers with month to month contracts show a high churn compared to those with long-term contracts
2. Those with no online security will churn faster
3. Customers without internet subscriptions faster
4. Those who do not get tech support churn faster
5. Customers are more likely to churn in their first year of subscription compared to those engaged for 5+ years
6. Those who have fibre optics internet connection show higher churn
7. Higher churn seen in senior citizens
8. The client base is largely young , unmarried and with no dependants

VISUALIZATION



VISUALIZATION





MODEL EVALUATION & SELECTION

Notes:

- The problem in consideration is a classification problem. i.e. the prediction we require is whether the customer will churn or not.
- The data set is imbalanced, i.e. none of the customers who churn are more than the customers who do not in the dataset. Hence, we proceed by balancing the dataset and following data preprocessing and feature selection.
- Hence, we use the following 4 supervised learning algorithms : 1. Gradient Boost Classifier , 2. Logistic Regression 3. Random Forest Classifier 4. Decision tree classifier
- Mentioned below are the evaluation metrics ->

index	Model	Accuracy	Precision	Recall	F1 Score	f1 score
3	GradientBoost	0.7439659252	0.5088547816	0.7751798561	NaN	0.6143977192
0	RandomForest	0.7150970185	0.4764826176	0.8381294964	NaN	0.6075619296
1	LogisticRegression	0.6994794132	0.4611602753	0.8435251799	NaN	0.5963127781
2	DecisionTree	0.695693327	0.4558375635	0.8075539568	NaN	0.5827384815

- The Gradient Boosting Classifier stands out as the most suitable model for this case due to its performance, as evidenced by a high F1 score. The F1 score, a balanced measure of precision and recall, reflects the classifier's ability to accurately identify positive instances while minimizing false positives and false negatives. The F1 score achieved by the Gradient Boosting Classifier underscores its effectiveness in striking a harmonious balance between precision and recall, making it a robust choice for tackling this specific problem and delivering reliable and accurate predictions.

DEPLOYMENT

In our machine learning project deployment, we have harnessed the power of GitHub, Flask API, Docker, and Jenkins to orchestrate a robust and streamlined deployment process. GitHub serves as our collaborative code repository, allowing multiple contributors to work harmoniously on the project, manage version control, and facilitate efficient code sharing. The Flask API provides a user-friendly interface for interacting with our machine learning model, enabling seamless communication between users and the underlying predictive capabilities. Docker plays a pivotal role by encapsulating the entire application, along with its dependencies, into a portable container, ensuring consistent behavior across different environments and simplifying deployment across various platforms. Jenkins, our automation powerhouse, automates critical tasks like integration, testing, and deployment, guaranteeing a reliable and reproducible deployment pipeline. This comprehensive deployment strategy enhances collaboration, accelerates the deployment process, and ensures scalability, maintainability, and version control. By leveraging these technologies, we are not only optimizing efficiency and reducing errors but also establishing a robust foundation for deploying our machine learning project with precision and confidence in real-world scenarios.

GitHUb Repository

STEP 1

Docker image & containerizatio n

STEP 2

Jenkins integration

STEP 3

SOLUTIONS

- Providing discounted packages on long term contracts so that customers prefer it over monthly contracts
- Marketing strategies to engage young customers
- Marketing strategies to engage unmarries customers who live independently
- Providing and improving tech support services
- Collaboration packages with OTT companies to engage customers who stream movies and TV online
- Improving online payment facilities