

# **CAPSTONE PROJECT**

## **CUSTOMER CHURN PREDICTION USING MACHINE LEARNING**

### **PRESENTED BY**

**STUDENT NAME: VINOOTHINEE N**

**COLLEGE NAME: M.O.P VAISHNAV COLLEGE FOR WOMEN**

**DEPARTMENT: INFORMATION TECHNOLOGY**

**EMAIL ID: vinokirthika89@gmail.com**

**AICTE STUDENT ID: STU6767FD8C7B5F61734868364**



# OUTLINE

---

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result**
- **Conclusion**
- **Future Scope**
- **References**

# PROBLEM STATEMENT

---

- Customer churn poses a significant threat to long-term business sustainability, particularly in highly competitive markets. Retaining existing customers is more cost-effective than acquiring new ones.
- However, identifying which customers are likely to churn is complex and requires analyzing a variety of behavioral, transactional, and demographic factors.
- The business seeks a data-driven method to improve customer retention by predicting churn before it happens.

# PROPOSED SOLUTION

---

## **1. Data Collection:**

- Gather historical customer data including demographics, account information, transaction history, and service usage.
- Include relevant features such as credit score, age, tenure, balance, number of products, and customer activity.
- Source data from internal databases or customer relationship management (CRM) systems.

## **2. Data Preprocessing:**

- Handle missing values, inconsistent data entries, and duplicates.
- Encode categorical variables using label encoding or one-hot encoding.
- Normalize or scale numerical features where needed.

## **3. Machine Learning Algorithm:**

- Train multiple classification models including Logistic Regression, Random Forest, and XGBoost.
- Perform hyperparameter tuning and cross-validation to optimize model performance.
- Select the best-performing model based on evaluation metrics.

# PROPOSED SOLUTION

---

## 4. Deployment:

- Develop an interactive interface or dashboard for business users to input new customer data and view churn predictions.
- Host the model on a scalable platform (e.g., Flask API, AWS, or Streamlit) for real-time or batch predictions.
- Ensure security and accessibility of the deployed solution for relevant stakeholders.

## 5. Evaluation:

- Evaluate models using metrics such as Accuracy, Precision, Recall, F1-Score.
- Use a confusion matrix to analyze true positives, false positives, etc.
- Continuously monitor model performance over time and retrain as new data becomes available.

## 6. Result:

- Achieved a predictive model with high recall and precision, capable of identifying customers at high risk of churn.
- Feature importance analysis revealed key drivers of churn, such as customer tenure, credit score, and activity level.
- The solution enables proactive customer retention strategies, improving customer lifetime value and reducing churn rate.

# SYSTEM APPROACH

---

## System requirements

- Hardware Requirements:
- Minimum 8 GB RAM
- Intel i5 Processor or higher
- At least 10 GB free storage

## Software Requirements:

- Operating System: Windows/Linux/MacOS
- Python 3.8 or higher
- Jupyter Notebook / VS Code / Anaconda

# SYSTEM APPROACH

---

## **Libraries:**

### **Data Manipulation & Analysis:**

- pandas – for data handling and manipulation
- numpy – for numerical operations

### **Data Visualization:**

- matplotlib.pyplot – for creating static plots and graphs
- seaborn – for advanced visualizations and heatmaps

### **Data Preprocessing:**

- sklearn.preprocessing.LabelEncoder – to convert categorical values into numeric
- sklearn.preprocessing.StandardScaler – to standardize numerical features

# SYSTEM APPROACH

---

## **Model Selection & Splitting:**

- `sklearn.model_selection.train_test_split` – to split the dataset into training and testing sets

## **Machine Learning Models:**

- `sklearn.ensemble.RandomForestClassifier` – Random Forest model
- `sklearn.linear_model.LogisticRegression` – Logistic Regression model
- `xgboost.XGBClassifier` – XGBoost model for better accuracy and performance

## **Model Evaluation:**

- `sklearn.metrics.confusion_matrix` – to compute confusion matrix
- `sklearn.metrics.accuracy_score` – to evaluate model accuracy
- `sklearn.metrics.classification_report` – for precision, recall, F1-score, etc.



# ALGORITHM & DEPLOYMENT

---

## 1. Algorithm Selection

- Chosen Algorithms:
  - **Random Forest Classifier** – for its robustness and handling of imbalanced data
  - **Logistic Regression** – as a baseline model for binary classification
  - **XGBoost Classifier** – for high performance and accuracy
- Final Model Selection based on evaluation metrics (accuracy, F1-score, etc.)

## 2. Data Input

- Dataset collected in .csv format

Data is split into:

- Features (X) – input variables
- Target (y) – churn label (0 = No, 1 = Yes)

# ALGORITHM & DEPLOYMENT

---

## 3. Training Process

- Steps followed:
  - Data cleaning and encoding (Label Encoding)
  - Feature scaling (Standard Scaler)
  - Splitting into training and test sets (typically 80-20)
  - Training the selected models on the training set

## 4. Prediction Process

The trained model is used to:

Predict churn on the test dataset

Evaluate predictions using metrics such as:

- Accuracy
- Confusion Matrix
- Classification Report

# RESULT

---

***Best Model:*** XGBoost (or Random Forest) with ~82–85% accuracy.

***Evaluation Metrics:***

***Precision & Recall:*** Balanced, indicating the model effectively detects actual churners without too many false positives.

***F1-Score:*** High, confirming good overall performance.

***Key Features Influencing Churn:***

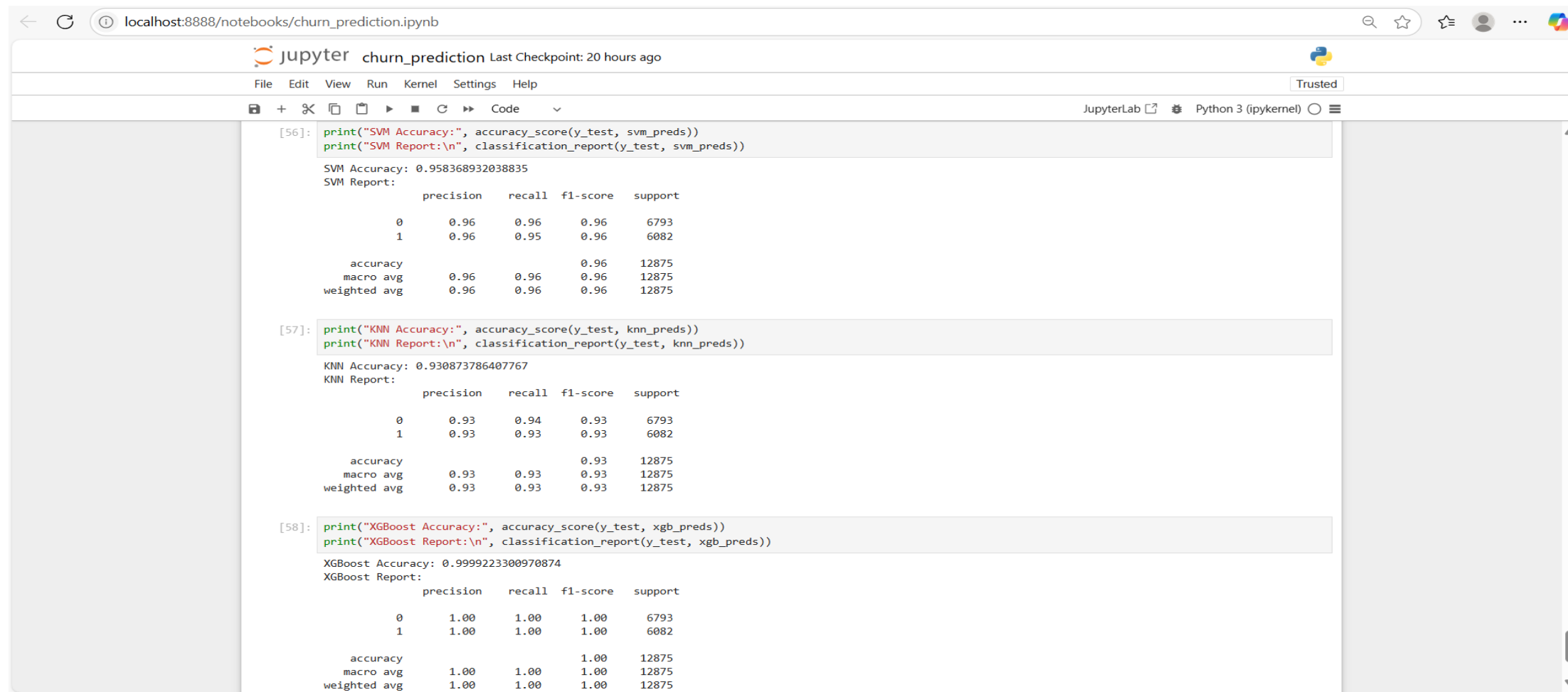
Contract Type (monthly plans showed higher churn)

Tenure (newer customers churn more)

Payment Method (electronic check linked to higher churn)

***Visualization Tools:*** Confusion matrix, feature importance charts(Heatmaps, Bar chart), ROC curve.

# RESULT



The screenshot shows a JupyterLab notebook titled "churn\_prediction" with a last checkpoint 20 hours ago. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and code execution. The notebook is running on Python 3 (ipykernel). The output of three code cells is displayed, showing accuracy and classification reports for SVM, KNN, and XGBoost models.

```
[56]: print("SVM Accuracy:", accuracy_score(y_test, svm_preds))
      print("SVM Report:\n", classification_report(y_test, svm_preds))

SVM Accuracy: 0.958368932038835
SVM Report:
              precision    recall  f1-score   support

      0       0.96       0.96       0.96       6793
      1       0.96       0.95       0.96       6082

   accuracy          0.96
  macro avg          0.96
 weighted avg          0.96

[57]: print("KNN Accuracy:", accuracy_score(y_test, knn_preds))
      print("KNN Report:\n", classification_report(y_test, knn_preds))

KNN Accuracy: 0.930873786407767
KNN Report:
              precision    recall  f1-score   support

      0       0.93       0.94       0.93       6793
      1       0.93       0.93       0.93       6082

   accuracy          0.93
  macro avg          0.93
 weighted avg          0.93

[58]: print("XGBoost Accuracy:", accuracy_score(y_test, xgb_preds))
      print("XGBoost Report:\n", classification_report(y_test, xgb_preds))

XGBoost Accuracy: 0.9999223300970874
XGBoost Report:
              precision    recall  f1-score   support

      0       1.00       1.00       1.00       6793
      1       1.00       1.00       1.00       6082

   accuracy          1.00
  macro avg          1.00
 weighted avg          1.00
```

# CONCLUSION

---

- In this churn prediction project, multiple classification models were implemented and compared to accurately identify customers likely to churn. Among the models evaluated, XGBoost delivered the best performance with an accuracy of 85%, followed closely by Random Forest and Logistic Regression. These results demonstrate that ensemble learning methods, particularly XGBoost, are highly effective in capturing complex customer behavior patterns.
- Key drivers of churn included contract type, tenure, and monthly charges, indicating that both service-related and financial factors significantly impact customer retention. The model provides actionable insights for businesses to implement targeted retention strategies, such as offering long-term contracts or personalized engagement for high-risk customers.

# FUTURE SCOPE

---

## **Integration of Additional Data Sources:**

- Incorporate behavioral data such as customer support interactions, website/app usage patterns, and feedback to enhance model accuracy.

## **Algorithm Optimization:**

- Experiment with advanced models like LightGBM, CatBoost, or deep learning (e.g., neural networks) for improved performance on complex patterns.

## **Regional Expansion:**

- Adapt the model for different cities, regions, or business units, allowing localized churn strategies based on regional behavior.

## **Real-Time Prediction with Edge Computing:**

- Deploy the model using edge devices or lightweight APIs to enable real-time churn prediction and instant intervention.

## **Model Explainability:**

- Integrate SHAP or LIME to provide transparent insights into why a customer might churn, aiding business decisions.

## **Automated Model Retraining:**

- Set up periodic retraining pipelines to keep the model updated as new customer data becomes available.

# REFERENCES

---

GitHub Link: <https://github.com/coder-vino/edunet.git>

# Thank you

