geoffreykemboi / phase3project

Type / to search

Code Issues Pull requests Agents Actions Projects Wiki Security Insights Settings

phase3project Public

1 Branch 0 Tags Go to file Go to file Add file Code

geoffreykemboi renaming my notebook from index1 to notebook d5c4807 · 24 minutes ago 13 Commits

data	first commit	2 days ago
notebook	renaming my notebook from index1 to...	24 minutes ago
presentation	renaming my notebook from index1 to...	24 minutes ago
Presentation.pdf	renaming my notebook from index1 to...	24 minutes ago
Presentation.pptx	renaming my notebook from index1 to...	24 minutes ago
README.md	Update ReadMe	3 hours ago
regression_cheat_sheet (1).py	Commit fit model & test accuracy	yesterday

README

PHASE 3 MACHINE LEARNING PROJECT

About

phase3project_regression_models_lasso_ridge

Readme

Activity

0 stars

0 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Churn Prediction Project

Project Overview

This project focuses on predicting customer churn using machine learning models. Churn prediction is a critical business objective because retaining existing customers is significantly more cost-effective than acquiring new ones. By identifying at-risk customers, the company can deploy targeted retention strategies, such as specialized discounts or loyalty programs, to maintain its revenue base and market share. This analysis compares a **Logistic Regression** baseline with a **Decision Tree Classifier** to determine the most effective approach for identifying customers likely to leave.

Business and Data Understanding

Stakeholder Audience

The primary stakeholders are the **Customer Success and Retention Teams**. These teams require a tool that flags customers who are likely to leave before they actually do. A predictive model allows them to transition from reactive support to proactive intervention.

Dataset Choice

The analysis uses the `ChurnInTelecom.csv` dataset, which contains 3,333 records of customer behavior.

- **Features:** Usage patterns (day/evening/night/intl minutes), account details (account length, international plan, voice mail plan), and customer service interactions.
- **Target Variable (`churn`):** * **Class 0:** Not churned (customers who stayed)
- **Class 1:** Churned (customers who left)

● Jupyter Notebook 99.2% ● Python 0.8%

Suggested workflows

Based on your tech stack

Pylint

Configure

Lint a Python application with pylint.

Publish Python Package

Configure

Publish a Python Package to PyPI on release.

SLSA Generic generator

Configure

Generate SLSA3 provenance for your existing release workflows

More workflows

Dismiss suggestions

- **Challenge:** The dataset is imbalanced (approximately churn rate). This makes identifying the minority class (churners) more difficult, requiring a focus on metrics beyond simple accuracy.

Modeling

The project followed an iterative approach to classification:

1. **Logistic Regression:** A linear model used as a baseline. It provides high interpretability regarding how each feature (like service calls) increases or decreases the log-odds of churn.
2. **Decision Tree Classifier:** A non-linear model that splits data into decision rules. This model is capable of capturing complex interactions between features that a linear model might miss.

Both models were trained on the same prepared dataset to ensure a fair comparison of their predictive capabilities.

Evaluation

The performance of both models was evaluated using **Precision**, **Recall**, and the **F1-score**. In this business context, **Recall** is a critical metric because failing to identify a churner (a "False Negative") results in lost revenue that is harder to recover than the cost of a retention offer.

Based on the final testing, both models achieved identical performance metrics:

Performance Metrics (Both Models)

- **Precision (71.88):**
- **Recall (65.90):**
- **F1-Score (68.76):**

Interpretation

- **Precision (71.88):** When the model flags a customer as a churn risk, there is a chance they are actually planning to leave.
- **Recall (65.90):** The model successfully identifies of all customers who actually churn.
- **F1-Score (68.76):** This represents a strong balance between precision and recall, ensuring the model is reliable for business deployment.

Conclusion

Rationale

Both the Logistic Regression and Decision Tree models provided a significant improvement over a majority-class baseline. With a recall of , the business can now proactively reach out to nearly two-thirds of all at-risk customers.

Results

The models demonstrated consistent predictive power. While Logistic Regression offers a clear mathematical view of feature influence, the Decision Tree provides a rule-based logic that is often easier to explain to non-technical stakeholders (e.g., "If calls and usage is high, then Churn").

Limitations

- **Class Imbalance:** While the models perform well, the underlying imbalance in the data means there is still room to improve the detection of the minority class.
- **Thresholding:** These results are based on a standard probability threshold. Adjusting this threshold could potentially increase recall at the expense of precision.

Recommendations

1. **Deploy the Decision Tree Model:** Given its equivalent performance and intuitive rule-based structure, the Decision Tree is recommended for its transparency in explaining "why" a customer is at risk.

2. **Focus on High-Impact Features:** Prioritize retention efforts on customers showing high day-time usage and those who have made multiple customer service calls.
3. **Implement Automated Flagging:** Integrate the model's predictions into the CRM dashboard to alert account managers in real-time.
4. **Future Work:** Explore ensemble methods like **Random Forest** or **XGBoost** to see if the recall can be pushed