گزارش پروژه تشخیص تراکنش‌های تقلبی با استفاده از SVM

مقدمه

هدف این پروژه، شناسایی تراکنش‌های تقلبی (Fraud Detection) بر اساس ویژگی‌های موجود در دیتاست است.

دیتاست شامل دو کلاس می‌باشد:

- کلاس ۰: تراکنش عادی
- کلاس ۱: تراکنش تقلبی

به دلیل اینکه تعداد تراکنش‌های تقلبی بسیار کمتر از تراکنش‌های عادی است، با یک دیتاست نامتوازن (Imbalanced) مواجه هستیم. برای همین باید در انتخاب روش پردازش داده‌ها و مدل‌سازی دقت کنیم.

(Data Cleaning) پاک‌سازی داده‌ها

در ابتدا داده‌های پرت (Outlier) مربوط به تراکنش‌های عادی حذف شدند.

برای این کار:

- ابتدا داده‌های کلاس ۰ (تراکنش عادی) و کلاس ۱ (تقلبی) جدا شدند.
- سپس برای کلاس ۰، مقادیر چارک اول (Q1) و چارک سوم (Q3) محاسبه شد.
- فاصله بین چارکی (IQR) بدست آمد.
- داده‌هایی که خارج از محدوده [Q1 - 1.5*IQR , Q3 + 1.5*IQR] بودند حذف شدند.

با این روش، داده‌های پرت فقط از کلاس ۰ حذف شدند و داده‌های تقلبی همانطور باقی ماندند تا اطلاعات مهم آن‌ها از بین نرود.

📌 نتیجه: دیتاست تمیزتر شد و اندازه دیتاست کاهش یافت.

جداسازی ویژگی‌ها و برچسب‌ها

پس از پاک‌سازی داده، متغیر ویژگی‌ها (X) و برچسب (y) از هم جدا شدند:

- X: شامل تمام ستون‌ها به جز ستون Class
- y: ستون Class به عنوان برچسب هدف

استانداردسازی داده‌ها (Standardization)

از آنجایی که مقیاس ویژگی‌ها با هم متفاوت است، داده‌ها با استفاده از StandardScaler استاندارد شدند.

این کار باعث می‌شود که الگوریتم‌های حساس به مقیاس (مانند SVM) عملکرد بهتری داشته باشند.

تقسیم داده‌ها به آموزش و تست

برای آموزش مدل، داده‌ها به دو بخش تقسیم شدند:

- X_train , y_train: ۸۰٪ داده‌های آموزشی)
- X_test , y_test: ۲۰٪ داده‌های تست)

در این مرحله از Stratified Sampling استفاده شد تا نسبت بین کلاس‌ها در داده‌های آموزش و تست حفظ شود.

مدل‌سازی SVM

مدل انتخاب‌شده برای این پروژه SVC از کتابخانه scikit-learn است.

پارامترهای انتخاب‌شده:

- kernel = ‘rbf’ → کرنل شعاعی برای جداسازی داده‌ها
- class_weight = ‘balanced’ → چون داده نامتوازن بود، وزن کلاس‌ها به صورت خودکار متعادل شد

📌 مدل روی داده‌های آموزشی (X_train , y_train) آموزش داده شد.

ارزیابی مدل

پس از آموزش مدل، پیش‌بینی روی داده‌های تست انجام شد.

برای ارزیابی، از معیار F1-Score (weighted) استفاده شد.

- مقدار weighted باعث می‌شود اثر عدم توازن کلاس‌ها کمتر شود.

همچنین گزارش کاملی با classification_report تولید شد که شامل:

- Precision
- Recall
- F1-Score

برای هر کلاس بود.

ساخت فایل Submission

در نهایت، داده‌های تست جداگانه (که بدون برچسب بودند) توسط مدل پیش‌بینی شدند.

خروجی در یک دیتافریم به اسم submission ذخیره شد که شامل ستون زیر بود:

- Class: مقادیر پیش‌بینی شده توسط مدل

تعداد سطرهای دیتافریم ۲۶۱۴ بود

در پایان، فایل submission.csv ذخیره شد تا برای ارسال یا استفاده بعدی آماده باشد.