এই ট্রান্সক্রিপ্টটিতে ডেটা সায়েন্স ফিল্ডের একটি অত্যন্ত গুরুত্বপূর্ণ বিষয়, অর্থাৎ এন্ড-টু-এন্ড মেশিন লার্নিং **(ML)** প্রজেক্ট ফ্লো কীভাবে কাজ করে, তার একটি ধাপে ধাপে বিস্তারিত উদাহরণ দেওয়া হয়েছে।

মূল লক্ষ্য হলো একটি ছোট "টয় ডেটা সেট" ব্যবহার করে দেখানো যে, ডেটা সংগ্রহ থেকে শুরু করে একটি মডেল তৈরি করে তাকে ওয়েবসাইটে ডিপ্লয় করার পুরো প্রক্রিয়াটি কেমন হয়।

## ১. প্রজেক্টের লক্ষ্য এবং ডেটা সেট

- লক্ষ্য: এই প্রজেক্টের প্রধান লক্ষ্য হলো একটি প্রেডিক্টিভ মডেল তৈরি করা। এই মডেলটি কোনো ছাত্রের **CGPA** (Cumulative Grade Point Average) এবং **IQ** (Intelligence Quotient) ইনপুট হিসেবে পেলে বলে দিতে পারবে যে তার প্লেসমেন্ট হবে কি না।
- ডেটা সেট: ব্যবহৃত ডেটা সেটটির নাম "placement.csv"। এতে ১০০ জন ছাত্রের ডেটা রয়েছে, যেখানে তিনটি প্রধান কলাম আছে: IQ, CGPA এবং প্লেসমেন্ট স্ট্যাটাস (হয়েছে/হয়নি)।
- প্রয়োজনীয় লাইব্রেরি: কাজ শুরু করার জন্য Pandas এবং NumPy লাইব্রেরিগুলি আমদানি করা হয়।

## ২. মেশিন লার্নিং ফ্লো-এর ধাপগুলি

সম্পূর্ণ ML ফ্লো-কে কয়েকটি প্রধান ধাপে ভাগ করে কাজ করা হয়েছে:

ধাপ ক: ডেটা লোডিং এবং প্রি-প্রসেসিং **(Pre-processing)**

১. ডেটা লোড: প্রথমে CSV ফাইলটি Pandas ব্যবহার করে ডেটাফ্রেমে লোড করা হয়। ২. প্রাথমিক পর্যবেক্ষণ: ডেটাফ্রেমে দেখা যায় যে ১০০টি রো এবং ৪টি কলাম রয়েছে। একটি কলাম অপ্রয়োজনীয় ছিল। ৩. পরিষ্কার করা: প্রি-প্রসেসিং-এর অর্থ হলো ডেটাকে অ্যালগরিদমের জন্য প্রস্তুত করা। এর মধ্যে মিসিং ভ্যালু, আউটলায়ার বা অপ্রয়োজনীয় কলাম অপসারণ করা অন্তর্ভুক্ত। এই ডেটা সেটে কোনো মিসিং ভ্যালু ছিল না (প্রত্যেক কলামে ১০০টি নন-নাল ভ্যালু ছিল)। তাই প্রি-প্রসেসিং-এর একমাত্র কাজ ছিল অপ্রয়োজনীয় কলামটি বাদ দেওয়া।

ধাপ খ: এক্সপ্লোরেটরি ডেটা অ্যানালিসিস **(EDA)**

- উদ্দেশ্য: EDA-এর মাধ্যমে ডেটার ভেতরের লুকানো প্যাটার্ন বা সম্পর্কগুলি গ্রাফ ব্যবহার করে বোঝা যায়, যাতে মডেলিং-এর জন্য সঠিক প্যাটার্ন নির্ধারণ করা যায়।
- ভিজ্যুয়ালাইজেশন: Matplotlib লাইব্রেরি ব্যবহার করে CGPA এবং IQ-এর মধ্যে একটি স্ক্যাটার প্লট তৈরি করা হয়।
- অন্তর্দৃষ্টি: 'প্লেসমেন্ট' কলাম অনুযায়ী ডেটা পয়েন্টগুলিকে রঙ করা হলে দেখা যায় যে প্লেসমেন্ট হওয়া এবং প্লেসমেন্ট না-হওয়া ডেটাগুলি একটি সরলরেখা দ্বারা পৃথক করা সম্ভব। এই পর্যবেক্ষণ থেকে অনুমান করা হয় যে লজিস্টিক রিগ্রেশন ব্যবহার করা যেতে পারে, কারণ এটি লিনিয়ার ক্লাসিফিকেশনের জন্য একটি ভালো অ্যালগরিদম।

ধাপ গ: ইনপুট/আউটপুট সেপারেশন এবং ট্রেন-টেস্ট স্প্লিট

১. ইনপুট **(X)** এবং আউটপুট **(Y)** আলাদা করা: * **X (**ইন্ডিপেন্ডেন্ট ভেরিয়েবল**):** CGPA এবং IQ কলাম দুটিকে ইনপুট হিসেবে নেওয়া হয়, কারণ এদের মধ্যে কোনো পারস্পরিক সম্পর্ক নেই ধরে নেওয়া হয়। * **Y (**ডিপেন্ডেন্ট ভেরিয়েবল**):** প্লেসমেন্ট স্ট্যাটাস কলামটিকে আউটপুট হিসেবে নেওয়া হয়, কারণ এটি X-এর ওপর নির্ভর করে। ২. ট্রেন-টেস্ট স্প্লিট: মডেলকে সঠিকভাবে মূল্যায়ন করার জন্য ডেটা সেটকে দুই ভাগে ভাগ করা হয়। * গুরুত্ব: সরাসরি ওয়েবসাইটে ডিপ্লয় করার আগে মডেলের কর্মক্ষমতা পরীক্ষা করা জরুরি। গ্রাহকদের দিয়ে পরীক্ষা করানো ভুল পদ্ধতি। * পদ্ধতি:

Scikit-learn (Sklearn) থেকে `train_test_split` ফাংশন ব্যবহার করা হয়। * বিভাজন: ১০০টি রো-এর মধ্যে ১০% (১০টি রো) টেস্ট সেট (`X_test`, `Y_test`) হিসেবে গোপন রাখা হয়, যা মডেল প্রশিক্ষণের সময় দেওয়া হয় না। বাকি ৯০% (৯০টি রো) ট্রেনিং সেট (`X_train`, `Y_train`) হিসেবে ব্যবহার করা হয় মডেলকে শেখানোর জন্য।

ধাপ ঘ: ভ্যালু স্কেলিং **(Scaling)**

- প্রয়োজনীয়তা: ইনপুট কলামগুলির রেঞ্জ বা মান যদি খুব ভিন্ন হয় (যেমন, CGPA ০-১০ এর মধ্যে এবং IQ ৫০-১৫০ এর মধ্যে), তবে কিছু ML অ্যালগরিদম (যা দূরত্বের ওপর ভিত্তি করে কাজ করে) ভুল ফলাফল দিতে পারে।
- পদ্ধতি: Sklearn-এর **StandardScaler** ব্যবহার করে সমস্ত ইনপুট ভ্যালুগুলিকে একটি নির্দিষ্ট রেঞ্জে (সাধারণত -১ থেকে +১ এর মধ্যে) নিয়ে আসা হয়।
- গুরুত্বপূর্ণ: Scaler শুধুমাত্র ট্রেনিং ডেটা (`X_train`)-এর ওপর ফিট **(Fit)** করে প্যাটার্ন বোঝে এবং তারপর সেই প্যাটার্ন ব্যবহার করে ট্রেনিং ডেটা ও টেস্ট ডেটা উভয়কেই ট্রান্সফর্ম **(Transform)** করে।

ধাপ ঙ: মডেল প্রশিক্ষণ এবং মূল্যায়ন

১. মডেল প্রশিক্ষণ: ক্লাসিফিকেশনের জন্য **Logistic Regression** ক্লাসিফায়ারটি নির্বাচন করা হয়। `fit` ফাংশন ব্যবহার করে স্কেল করা ট্রেনিং ডেটা (`X_train`, `Y_train`) দিয়ে মডেলকে প্রশিক্ষণ দেওয়া হয়। ২. মডেল মূল্যায়ন: প্রশিক্ষণ সম্পূর্ণ হওয়ার পর, মডেলের কর্মক্ষমতা পরীক্ষা করা হয়। * মডেলটি গোপন রাখা টেস্ট ডেটা (`X_test`) ব্যবহার করে প্রেডিকশন (`Y_pred`) করে। * এই প্রেডিকশনগুলিকে আসল ফলাফল (`Y_test`)-এর সাথে তুলনা করা হয়। * অ্যাকুরেসি স্কোর **(Accuracy Score)** নামক ম্যাট্রিক্স ব্যবহার করে ফলাফল গণনা করা হয়। * এই উদাহরণে মডেলটি **90%** অ্যাকুরেসি অর্জন করে, অর্থাৎ ১০টি টেস্ট ইনপুটের মধ্যে ৯টি সঠিক পূর্বাভাস দেয়।

ধাপ চ: ডিসিশন বাউন্ডারি ভিজ্যুয়ালাইজেশন

- মডেলটি ডেটাতে আসলে কী প্যাটার্ন খুঁজেছে, তা বোঝানোর জন্য ডিসিশন বাউন্ডারি প্লট করা হয়।
- লজিস্টিক রিগ্রেশন ডেটা ক্লাসগুলিকে বিভক্ত করার জন্য যে সরলরেখাটি ব্যবহার করেছে, সেটি এই ভিজ্যুয়ালাইজেশনে দেখানো হয়। এটি আরও স্পষ্ট করে যে মডেলটি কোথায় সামান্য ভুল করেছে, যা ৯০% অ্যাকুরেসিকে সমর্থন করে।

ধাপ ছ: ডিপ্লয়মেন্টের প্রস্তুতি **(Serialization)**

১. মডেল সেভ করা: মডেলের কর্মক্ষমতা সন্তোষজনক হলে, এটিকে ওয়েবসাইটে ব্যবহারের জন্য সেভ করতে হয়। ২. পিকল **(Pickle)** ব্যবহার: পিকল লাইব্রেরি ব্যবহার করে মেশিন লার্নিং মডেল অবজেক্টটিকে (`clf`) একটি বাইনারি ফাইলে (`model.pkl`) রূপান্তর করা হয়। এই ফাইলটি অন্য কোনো এনভায়রনমেন্টে (যেমন ওয়েবসাইট সার্ভারে) মডেলটিকে লোড করে ব্যবহার করার সুযোগ দেয়।

## ৩. ডিপ্লয়মেন্টের সংক্ষিপ্ত বিবরণ

- একটি ডেমো ওয়েবসাইট দেখানো হয়, যেখানে CGPA এবং IQ ইনপুট করলে মডেলটি প্লেসমেন্টের পূর্বাভাস দেয়।

- চূড়ান্ত ধাপে, এই ওয়েবসাইটটিকে একটি সার্ভারে ডিপ্লয় করতে হয়। বক্তা উল্লেখ করেছেন যে এই কাজের জন্য Heroku, Amazon Web Services (AWS) Elastic Beanstalk, বা Google Cloud Platform (GCP)-এর মতো প্ল্যাটফর্ম ব্যবহার করা যেতে পারে।
- এই সম্পূর্ণ প্রক্রিয়াজুড়ে (ডেটা থেকে ওয়েবসাইট পর্যন্ত) কাজ করাই হলো এন্ড-টু-এন্ড মেশিন লার্নিং ফ্লো।

Notebook name : end to end ml model . dataset : placement csv,modified placement csv