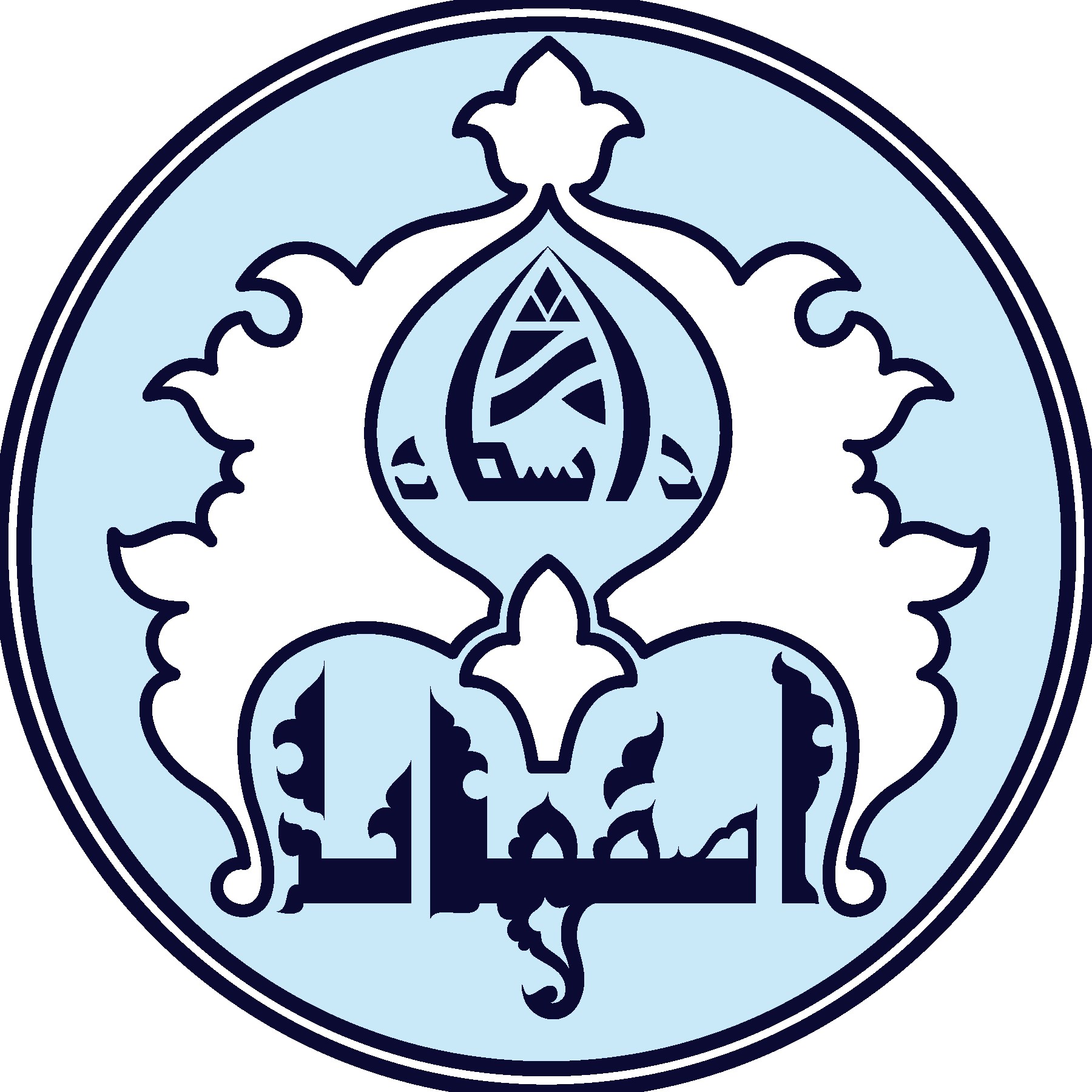
**به نام خداوند بخشنده مهربان**

****

**عنوان پروژه**پیش‌بینی قیمت منزل با استفاده از رگرسیون خطی

**عنوان درس**

یادگیری ماشین

**استاد**

دکتر الهام قصر‌الدشتی

**دستیاران آموزشی**

مهرداد قصابی

مریم صفوی

**گردآورنده**

سید حسین حسینی

**بهار ۱۴۰۴**

**دانشکده مهندسی کامپیوتر**

**دانشگاه اصفهان**

این پروژه با هدف ساخت یک مدل رگرسیون خطی برای پیش‌بینی یک تابع چندمتغیره‌ی غیرخطی طراحی شده است. ورودی مدل سه متغیر x,y,z بوده و خروجی، مقدار تقریبی تابع چندجمله‌ای F(x,y,z) می‌باشد**.**

* **روش تحلیلی**
* **کتابخانه‌ها**
* Numpy : برای کار با آرایه‌ها و داده‌ها و انجام عملیات جبری مناسب
* Pandas : برای تحلیل داده‌ها و همچنین انجام عملیات مناسب بر روی دیتافریم‌ها
* Sklearn : یکی از کتابخانه‌های یادگیری ماشین برای انجام عملیات‌های مرتبط با آن
* Gdown : برای دانلود دیتاست از لینک دریافتی
* **خواندن داده فایل از اکسل**

کد ابتدا بررسی می‌کند که آیا فایل وزن‌های آموزش‌یافته وجود دارد یا نه (regression\_weights.npy). در صورت نبود آن، از فایل اکسل (Polynomial\_Functions.xlsx) داده‌های آموزشی را می‌خواند.

* **تولید بردار ویژگی ها**

توابعی برای استخراج ویژگی‌ها از داده‌های خام پیاده‌سازی شده است، در راستای اینکه معادله مورد نظر به ما داده شده و ما هم با استفاده از مهندسی ویژگی، ویژگی های مورد نیاز را بدست می آوریم تا بتوانیم ماتریس ویژگی ها را ساخته و مقدار وزن ها را مشخص کنیم.

این تابع ۲۰ ویژگی برای ترکیب‌های چندجمله‌ای مختلف از سه متغیر تولید می‌کند. این ویژگی‌ها شامل توانی از متغیرها و ترکیب‌های آن‌ها مثل x2y, xyz هستند.

* **آموزش مدل با معادله نرمال**

با استفاده از معادله نرمال زیر وزن‌ها آموزش داده می‌شوند:

β = (XTX)−1XTY

 در صورتی که ماتریس منفرد باشد، از شبه‌وارون (Pseudo-Inverse) استفاده می‌شود.

* **ذخیره سازی یا بارگذاری وزن ها**

در صورت موفقیت‌آمیز بودن آموزش، وزن‌ها در فایل .npy ذخیره می‌شوند و در اجراهای بعدی مستقیماً بارگذاری خواهند شد و با توجه به همین مسئله وزن های بدست آمده را میتوانیم در مدل استفاده کنیم.

* **روش گرادیان کاهشی**

معادله هدف مدل:

F(x,y,z) = b + w₁x + w₂x² + w₃x³ + w₄y¹ + w₅y² + w₆y³ + w₇z + w₈z² + w₉z³ + w₁₀xy + w₁₁x²y + w₁₂xy² + w₁₃xz + w₁₄x²z + w₁₅xz² + w₁₆yz + w₁₇y²z + w₁₈yz² + w₁₉xyz

که در آن b بایاس (bias) و wᵢ وزن‌های (weights) مربوط به هر ویژگی هستند.

* **کتابخانه‌ها**
  + pandas برای کار با داده‌ها خواندن فایل اکسل، ایجاد.
  + numpy برای محاسبات عددی (کار با آرایه‌ها، عملیات ریاضی).
  + scikit-learn برای تقسیم داده (train\_test\_split). توجه: مدل اصلی با گرادیان کاهشی دستی پیاده‌سازی شده، نه با LinearRegression  اسکیت‌لرن برای آموزش نهایی.
  + Matplotlib برای رسم نمودارها (نمودار یادگیری).
  + seaborn برای بهبود ظاهر نمودارها (اختیاری).
  + gdown برای دانلود فایل از گوگل درایو در این کد استفاده نشده، داده از فایل محلی خوانده می‌شود.
  + warnings برای مدیریت هشدارهای پایتون.

فایل داده: فایل اکسل با نام Polynomial\_Functions.xlsx باید در مسیر اجرای نوت‌بوک وجود داشته باشد یا مسیر صحیح آن در کد مشخص شود.

* **شرح دیتاست (Dataset Description)**
* ساختار: دیتاست شامل ۱۰۰۰۰ نمونه (ردیف) و ۴ ستون است:
  + x: ویژگی ورودی اول (float64).
  + y: ویژگی ورودی دوم (float64).
  + z: ویژگی ورودی سوم (float64).
  + F(x, y, z): متغیر هدف یا خروجی (float64).
* کیفیت داده: داده‌ها فاقد مقادیر گمشده (Missing Values) هستند و همگی از نوع عددی می‌باشند. آمار توصیفی و تعداد مقادیر یکتا در کد بررسی شده است.
* **مراحل پیاده‌سازی و متدولوژی**

پروژه مراحل زیر را دنبال می‌کند:

1. بارگذاری و بررسی داده:
   * فایل اکسل با استفاده از pandas.read\_excel خوانده می‌شود.
   * اطلاعات کلی دیتاست (df.info())، مقادیر گمشده (df.isnull().sum())، آمار توصیفی (df.describe()) و تعداد مقادیر یکتا (df.nunique()) نمایش داده می‌شود.
2. تقسیم داده (Train/Test Split):
   * داده‌ها به دو بخش آموزش (Train) و آزمون (Test) تقسیم می‌شوند.
   * ویژگی‌ها (x, y, z) در متغیر X و متغیر هدف (F(x, y, z)) در متغیر y قرار می‌گیرند.
   * از train\_test\_split با نسبت 80% برای آموزش و 20% برای آزمون (test\_size=0.2) و random\_state=42 برای تکرارپذیری نتایج استفاده می‌شود.
3. پیش‌پردازش (Preprocessing):
   * حذف داده‌های پرت (Remove Outliers):
     + تابعی به نام remove\_outliers\_auto\_xy\_with\_output تعریف شده که از روش دامنه بین چارکی (IQR - Interquartile Range) برای شناسایی و حذف داده‌های پرت استفاده می‌کند.
     + این تابع *فقط روی داده‌های آموزش* (X\_train, y\_train) اعمال می‌شود.
     + نکته: بر اساس خروجی کد، در این اجرا فقط داده‌های پرت مربوط به متغیر هدف (y) حذف شده‌اند.
   * نرمال‌سازی داده (Normalize Data):
     + تابعی به نام normalize\_data تعریف شده که نرمال‌سازی Z-score (میانگین صفر و انحراف معیار یک) را انجام می‌دهد.
     + میانگین و انحراف معیار *فقط از داده‌های آموزش* محاسبه می‌شود.
     + این مقیاس‌بندی روی هر دو مجموعه داده آموزش و آزمون (هم ویژگی‌ها X و هم هدف y) اعمال می‌شود.
     + پارامترهای نرمال‌سازی (میانگین و انحراف معیار) برای ویژگی‌ها (feature\_scaler) و هدف (output\_scaler) ذخیره می‌شوند تا بعداً برای پیش‌بینی ورودی جدید و دنرمال‌سازی خروجی استفاده شوند.
     + تابعی برای دنرمال‌سازی (denormalize) نیز تعریف شده است.
4. مهندسی ویژگی (Feature Engineering):
   * تابعی به نام feature\_engineering تعریف شده است.
   * این تابع ورودی‌های نرمال‌شده x, y, z را گرفته و ۱۹ ویژگی جدید مطابق با معادله چندجمله‌ای مورد نظر ایجاد می‌کند:  
     x, x², x³, y, y², y³, z, z², z³, xy, x²y, xy², xz, x²z, xz², yz, y²z, yz², xyz
   * این تبدیل روی داده‌های نرمال‌شده آموزش (X\_train\_norm) و آزمون (X\_test\_norm) اعمال می‌شود و بردارهای ویژگی گسترش‌یافته (X\_train\_vector, X\_test\_vector) را تولید می‌کند. این کار باعث می‌شود مدل رگرسیون خطی بتواند روابط غیرخطی را مدل کند.
5. پیاده‌سازی و آموزش مدل (Model Implementation):
   * تابع هزینه، خطای میانگین مربعات (mse\_loss) به صورت دستی تعریف می‌شود.
   * تابع linear\_regression برای آموزش مدل با استفاده از گرادیان کاهشی پیاده‌سازی شده است.
   * این تابع وزن‌ها (weights) و بایاس (bias) را به صورت تکراری در طول epochs با استفاده از learning\_rate مشخص شده، به‌روزرسانی می‌کند.
   * ورودی‌های این تابع، بردارهای ویژگی مهندسی‌شده (X\_train\_vector, X\_test\_vector) و مقادیر هدف نرمال‌شده (y\_train\_norm, y\_test\_norm) هستند.
   * در هر epoch، خطای آموزش و آزمون محاسبه و چاپ می‌شود.
   * در نهایت، وزن‌ها و بایاس نهایی مدل و همچنین مقادیر خطای نهایی آموزش و آزمون برگردانده می‌شوند.
   * نمودار یادگیری (Learning Curve) که خطای آموزش و آزمون را در طول epochها نشان می‌دهد، با استفاده از matplotlib رسم می‌شود تا روند همگرایی مدل بررسی شود.
6. پیش‌بینی با تابع چندجمله‌ای (Polynomial Function Prediction):
   * آخرین بخش کد به کاربر اجازه می‌دهد مقادیر جدیدی برای x, y, و z وارد کند.
   * مراحل زیر برای هر ورودی جدید انجام می‌شود:
     + ورودی کاربر (x, y, z) دریافت می‌شود.
     + ورودی‌ها با استفاده از feature\_scaler (مقیاس‌کننده ویژگی‌ها که در مرحله پیش‌پردازش ذخیره شد) نرمال‌سازی می‌شوند.
     + ویژگی‌های چندجمله‌ای و تعاملی با استفاده از تابع feature\_engineering روی ورودی نرمال‌شده ساخته می‌شوند.
     + پیش‌بینی نرمال‌شده با استفاده از وزن‌ها و بایاس یادگرفته شده محاسبه می‌شود (np.dot(input\_norm\_vector, weights) + bias).
     + نتیجه پیش‌بینی نرمال‌شده با استفاده از output\_scaler (مقیاس‌کننده هدف) دنرمال‌سازی می‌شود تا به مقیاس اصلی تابع F برگردد.
     + مقدار پیش‌بینی نهایی (دنرمال‌شده) چاپ می‌شود.

۵. توضیح توابع کلیدی

* remove\_outliers\_auto\_xy\_with\_output(x, y): داده‌های پرت را با روش IQR از x و y (داده آموزش) حذف می‌کند.
* normalize\_data(X\_train, X\_test, y\_train, y\_test): نرمال‌سازی Z-score را روی داده‌های آموزش و آزمون اعمال کرده و مقیاس‌کننده‌ها را برمی‌گرداند.
* denormalize(data, scaler): داده نرمال‌شده را با استفاده از مقیاس‌کننده به مقیاس اصلی برمی‌گرداند.
* feature\_engineering(df): ویژگی‌های چندجمله‌ای و تعاملی درجه ۳ را از ستون‌های 'x', 'y', 'z' یک DataFrame می‌سازد.
* mse\_loss(y\_true, y\_pred): خطای میانگین مربعات را محاسبه می‌کند.
* linear\_regression(X\_train, y\_train, X\_test, y\_test, learning\_rate, epochs): مدل رگرسیون خطی را با گرادیان کاهشی دستی روی داده‌های ورودی آموزش می‌دهد و وزن‌ها، بایاس و تاریخچه خطا را برمی‌گرداند.

۶. نحوه استفاده

1. نصب پیش‌نیازها: اطمینان حاصل کنید که تمام کتابخانه‌های لیست شده در بخش ۲ نصب هستند (pip install pandas numpy scikit-learn matplotlib seaborn gdown openpyxl).
2. آماده‌سازی داده: فایل Polynomial\_Functions.xlsx را در کنار فایل نوت‌بوک قرار دهید یا مسیر آن را در کد (سلول مربوط به pd.read\_excel) به‌روز کنید.
3. اجرای نوت‌بوک: سلول‌های نوت‌بوک را به ترتیب از بالا به پایین اجرا کنید.
4. مشاهده نتایج: خروجی هر سلول را بررسی کنید، از جمله اطلاعات دیتاست، اندازه‌های مجموعه آموزش/آزمون، روند حذف داده‌های پرت، داده‌های نرمال‌شده، بردارهای ویژگی مهندسی‌شده، روند کاهش خطا در طول آموزش (خروجی متنی و نمودار یادگیری) و وزن‌ها و بایاس نهایی مدل.
5. انجام پیش‌بینی: در آخرین سلول، وقتی برنامه از شما مقادیر x, y, و z را درخواست کرد، اعداد مورد نظر خود را وارد کنید تا پیش‌بینی مدل برای آن ورودی نمایش داده شود.

۷. نتایج

* مدل با موفقیت بر روی داده‌های مهندسی‌شده آموزش داده شده است.
* نمودار یادگیری نشان می‌دهد که خطای آموزش و آزمون در طول زمان کاهش یافته و مدل به همگرایی رسیده است.
* مقادیر نهایی خطای میانگین مربعات (MSE) برای داده‌های نرمال‌شده آموزش و آزمون بسیار پایین و نزدیک به هم هستند (حدود 0.005)، که نشان‌دهنده برازش (fit) خوب مدل بدون بیش‌برازش (overfitting) قابل توجه است.
* وزن‌ها و بایاس نهایی مدل که ضرایب معادله چندجمله‌ای را نشان می‌دهند، محاسبه و چاپ شده‌اند.
* بخش نهایی کد امکان پیش‌بینی مقادیر جدید F را برای ورودی‌های دلخواه x, y, z فراهم می‌کند.

۸. نکات مهم

* حذف داده‌های پرت و محاسبه پارامترهای نرمال‌سازی (میانگین و انحراف معیار) *فقط* بر اساس داده‌های آموزش انجام می‌شود تا از نشت اطلاعات از مجموعه آزمون به آموزش (Data Leakage) جلوگیری شود.
* مدل رگرسیون با استفاده از گرادیان کاهشی دستی پیاده‌سازی شده است که درک عمیق‌تری از نحوه کارکرد الگوریتم ارائه می‌دهد.
* **منابع**
* Ng, Andrew. Machine Learning (Coursera)
* شریفی زارچی، علی. دوره یادگیری ماشین، دانشگاه صنعتی شریف
* Géron, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019
* OpenAI. ChatGPT

**سید حسین حسینی دولت آبادی**

**😊 موفق باشید 😉**