

**Electrical Engineering Department**

Data Mining – Dr.Amir Mazlaghani

**Report Exp 2**

نام دانشجو: علی بابالو – ۹۸۳۱۳۲۲

ایمیل :‌ alibabaloo@aut.ac.ir

در ابتدا کتابخانه های مورد نیاز و دیتاستمان را ایمپورت می­کنیم. سپس با استفاده از تابع loc یک ستون جدید به نام population level تشکیل می­دهیم که مقادیر آن با توجه مقدار اتریبیوت population تعریف می­شود. یعنی مقدار داخل ستون جدید اضافه شده در یکی از چارک های population قرار بگیرد دارای لیبل های Low, Mid, High, Over می­شود. برای چارک بندی کردن از تابع qcut استافاده می­کنیم که ابتدا لیست دیتا ها (در اینجا ستون population)، تعداد تقسیم بندی ها (در اینجا ‌q = 4)‌و نام لیبل هارا به آن می­دهیم.

برای مرحله preprocess ابتدا با استفاده از تابع dropna، data object هایی که مقدار NaN دارند را دراپ می­کنیم. سپس ستون هایی با داده هایی از نوع categorical را با استفاده از تابع LableEncoder، انکود کرده سپس با استفاده از تابع StandardScaler که هردوی آنها در کتابخانه sklearn هستند، نورمالایز می­کنیم.

سپس برای هر دو قسمت رگرسیون و کلسیفیکیشن داده هارا به نسبت ۸۰ به ۲۰ با استفاده از تابع train\_test\_split از کتاخانه sklearn تقسیم بندی می­کنیم. برای قسمت رگرسیون داده های نهایی از ستون population و قسمت کلاسیفیکیشن داده نهایی از ستون population level هستند.

برای قسمت رگرسیون خطی ابتدا یک مدل با استفاده از تابع linear\_model.LinearRegression() تشکیل می­دهیم. سپس این مدل را با داده آموزش fit می­کنیم و در نهایت آن مدل اموزش داده شده را برای داده تست predict می­کنیم. سپس مقدار MSE را بوسیله تابع mean\_squared\_error از کتابخانه sklearn بدست می­آوریم. برای قسمت غیر خطی هم تمامی مراحل بالا را طی می­کنیم اما مدلمان را بصورت غیرخطی با استفاده از تابع PolynomialFeatures با درجه ۲ تشکیل می دهیم.

برای قسمت کلسیفیکیشن ابتدا مدل درخت تصمیم داریم که با استفاده از تابع DecisionTreeClassifier با معیار Entropy درست می­کنیم و آن را با داده های آموزش فیت کرده و برای داده های تست پردیکت می­کنیم. سپس با استفاده از تابع های Accuracy\_score و Precision\_score مقدار دقت و خطا این مدل را برای داده های تست و ترین محاسبه می­کنیم.

برای قسمت رندوم فارست هم مراحل بالا را تکرار می­کنیم اما در اینجا مدلمان RandomForestClassifier می­باشد.

در قسمت KNN ابتدا مدلمان را با استفاده از تابع KNeighborsClassifier تعریف می­کنیم. در هر مرحله باید مقدار پارامتر n\_neighbors را بترتیب برابر ۲ ، ۳ و ۵ قرار می­دهیم. این پارامتر تعداد همسایه هارا نشان می­دهد. سپس با استفاده از تابع Precision\_score و Accuracy\_score بترتیب خطا و دقت این مدل هارا بدست می­آوریم.

در مرحله آخر نیز مدل SVM را با استفاده از تابع SVC تعریف می­کنیم و آن را برای داده های تست و ترین آموزش می­دهیم سپس خطا و accuracy آن مدل را با تابع Precision\_score و Accuracy\_score بدست می­آوریم.

**پایان**