**Name:** R Abhishek

**Reg No:**1832001

**Department:** MSc Data Science

Problem Statement:

Social network ad dataset is provided. The dataset contains 4 input variables and 1 target variable.

Input Variables: User ID, Gender, Age, Estimated Salary

Output Variable: Purchased.

It is a classification problem. Using the 4 input variables we have to build a ML model and classify whether it is purchased (1) or not purchased (0).

Installing Necessary Updates:

To solve the above problem statement, Python language is opted and Colab is the tool to be used. Colab usually maintain a default version of libraries. To use higher version or updated version, we have to install the necessary updates on Colab by specifying the updated version. Pandas-Profiling is an opensource Python module with which we can quickly do an exploratory data analysis with just a few lines of code. For using Pandas Profiling, we have to get it installed in Colab using pip install.

Importing Necessary Libraries:

We have to import pandas for storing data in the form of dataframe. Numpy is imported for working with numpy arrays. Pandas-profiling is imported for performing EDA process in few lines of code. For Scaling data, Standard Scalar is imported. For fitting numerous ML models, sklearn library is imported. For showing any extra graphs or charts, pyplot and seaboarn is imported.

Exploratory Data Analysis:

Data is read and stored as pandas dataframe. By using info() function we can see the information of the data. The data has 400 entries. There is no missing value in the data. The memory usage of the data is 15kb. Pandas Profiling is used to perform EDA for the dataset. Necessary library is imported and report is generated and stored as an HTML file.

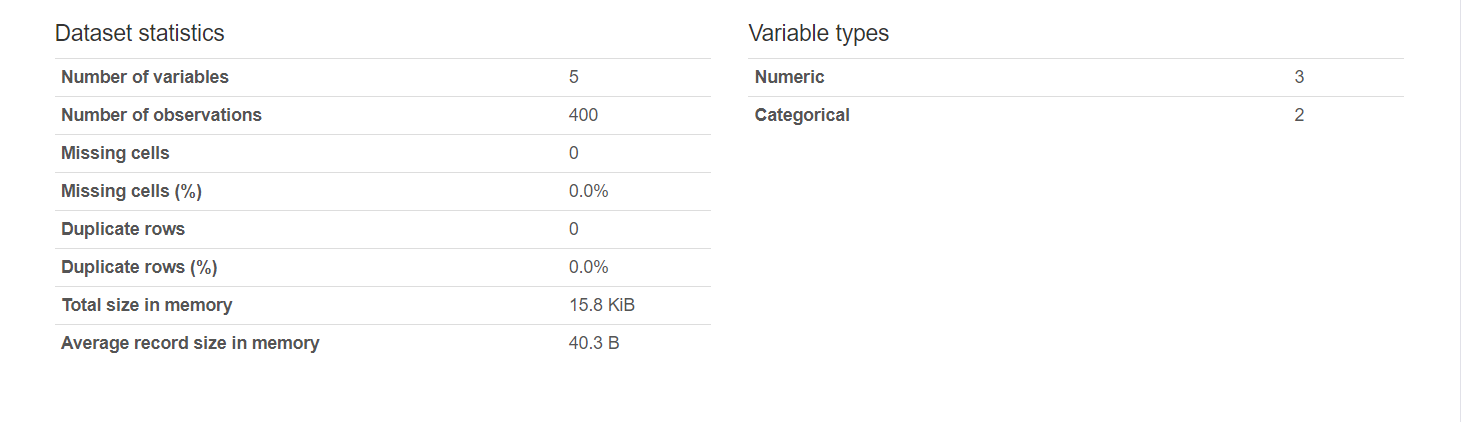


Figure 1. Data Statistics

Figure 1 shows the statistics of the dataset. There is no missing rows and no duplicate rows. There are 3 numeric variables and 2 categorical variables.

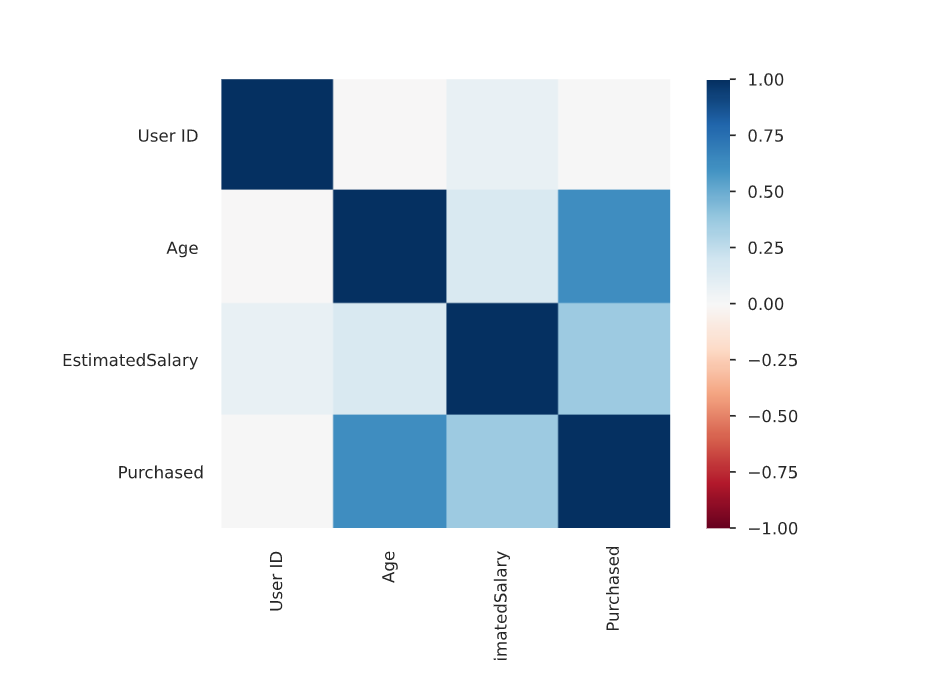


Figure 2 Correlation Heatmap

Among all the independent variables, Age has high correlation with target variable. User ID has very less correlation with the y variable. It does not contribute to the variable, that is it has no effect on the target variable. User ID can be omitted for calculations.

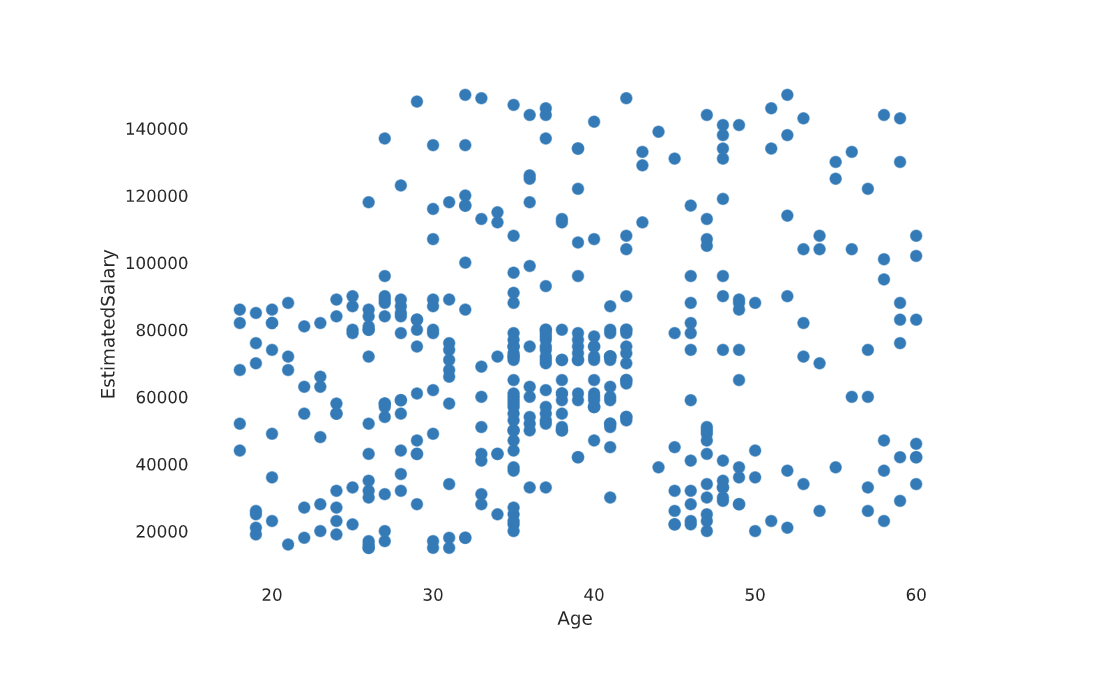


Figure 3 Age Vs Estimated Salary

In figure 3, we can see a scatter plot worked out for age and estimated salary. We can’t see any pattern followed. There is minimal relation between these two variables.

Data Pre-processing:

All independent variables are stored in X and target variable in y using slicing operator and locating through integer index of the variables. Data is Scaled using StandardScaler(). StandardScaler is a scaling technique where the values are centred around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. We scale the data because the data may contain variables which may vary in terms of unit. This will lead to data discrepancy. To split the data into train data and test data we need train\_test\_split library to be imported from sklearn.model\_selection.

Gender is a categorical variable, we can’t input model with categorical variable since ML model accepts numeric values only. Hence we use Label Encoder to Gender variable. By using the label encoder the value of female is given 0 and male is given 1. Now these binary value can be passed to the ML models and the ML models will get trained to treat 0 as female and 1 as male.

Splitting data into train data and test data:

Stratified ShuffleSplit cross-validator. Provides train/test indices to split data in train/test sets. This cross-validation object is a merge of StratifiedKFold and ShuffleSplit, which returns stratified randomized folds. The folds are made by preserving the percentage of samples for each class. Test size is assumed to be 0.2. that is, 20% of the data is used for testing the model and 80% to train the model. Random State is set to a constant so that each and every time we split the data, the data is split into same pieces, so that the model wont fluctuate.

Model Fitting:

SGD Classifier:

We use SGD classifier to classify the data. Stochastic Gradient Descent (SGD) is a simple yet very efficient approach to fitting linear classifiers and regressors under convex loss functions such as (linear) Support Vector Machines and Logistic Regression. We pass through many parameters to the SGD model such as hinge, log, modified huber, squared hinge, perceptron. We used Grid Search CV. It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters.

We get the best SGD model with hinge loss and elasticnet penalty. The best score is 0.907.

Kernel Approximation:

This module contains It helps to loop through predefined hyperparameters and fit your estimator (model) on your training set. So, in the end, you can select the best parameters from the listed hyperparameters. The following feature functions perform non-linear transformations of the input, which can serve as a basis for linear classification or other algorithms.

The advantage of using approximate explicit feature maps compared to the kernel trick, which makes use of feature maps implicitly, is that explicit mappings can be better suited for online learning and can significantly reduce the cost of learning with very large datasets.

Radial Basis Function Kernel:

The RBFSampler constructs an approximate mapping for the radial basis function kernel, also known as Random Kitchen Sinks. This transformation can be used to explicitly model a kernel map, prior to applying a linear algorithm, for example a linear SVM. The model is fit and the best score is 0.951.

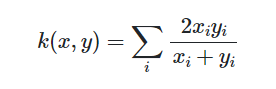
Nystroem:

The Nystroem method, as implemented in Nystroem is a general method for low-rank approximations of kernels. It achieves this by essentially subsampling the data on which the kernel is evaluated. By default, Nystroem uses the rbf kernel, but it can use any kernel function or a precomputed kernel matrix. The number of samples used - which is also the dimensionality of the features computed - is given by the parameter n\_components. The best score is 0.899.

AdditiveChi2Sampler:

The additive chi squared kernel is a kernel on histograms, often used in computer vision.

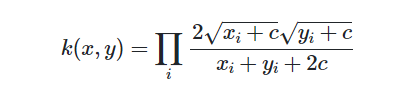
The additive chi squared kernel as used here is given by,



The approximate feature map provided by AdditiveChi2Sampler can be combined with the approximate feature map provided by RBFSampler to yield an approximate feature map for the exponentiated chi squared kernel. The best score is 0.945.

Skewed Chi Squared Kernel:

The skewed chi squared kernel is given by:



It has properties that are similar to the exponentiated chi squared kernel often used in computer vision, but allows for a simple Monte Carlo approximation of the feature map.

The usage of the SkewedChi2Sampler is the same as the usage described above for the RBFSampler. The only difference is in the free parameter, that is called c. The best score is 0.827.

Conclusion:

SGD classifier can be used to classify the given dataset with best score of 0.92.