**ML ASSIGNMENT – 5 REPORT**

*Apoorva Shukla (as11445)*

*Akanksha Nilosey (arn337)*

**Problem Definition:**

To predict the yearly income from wages (in dollars) of a person in the United States, based on other features of the person.

**Feature Selection & Handling of ? values:**

We have handled each column in the following way:

1.idnum: We eliminated the idnum column as it does not contribute in any way to the prediction of income.

2.Age: This numerical attribute did not have any ? values so we have replaced all the missing values with the mean of all the ages as most of the workers could belong to the mean of the ages.

3. Workerclass: We have replaced all the ? with a category (0). We have grouped the categorical classes given into the following 5 classes to lessen the number of categories and made a logical grouping of the type of employees:

* Category (1) - Private Employees: Employee of a private for-profit company or business, or of an individual, for wages, salary, or commissions, Employee of a private not-for-profit, tax-exempt, or charitable organization
* Category (2) - Public Employees: Local government employee, (city, county, etc.), State government employee, Federal government employee.
* Category (3) - Self-employed: in own not incorporated business, professional practice, or farm. Self-employed in own incorporated business, professional practice or farm.
* Category (4) - Working without pay in family business or farm.
* Category (5) - Unemployed and last worked 5 years ago or earlier or never worked.

4. Interestincome: Replaced all ? with a category (0).

5. Traveltimetowork : Replaced all ? with a category (0) as this contains the wokers who are unemployed or who work from home so their travel time would be 0.

6. Vechicleoccupancy and Meansoftransport: We have combined these two features. For all the ‘?’ in vechicleoccupancy we replace it with all the categories of Meansoftransport except car truck or van. This is because this worker could have used either Meansoftransport if he was not using car/truck/van.

7. Schoolenrollment: Replaced all the ? with a category (0).

8. Educationalattain: Replaced all the ? with a category (0).

9.Workarrivaltime: We converted this categorical attribute to a numeric attribute and replaced all ? with 0. We have also combined classes as follows:

category 1 - 12 am to 7:30 am

category 2 - 7:30 am to 8 am

category 3 - 8 am to 8:30 am

category 4 - 8:30 am to 9 am

category 5 - 9 am to 9:30 am

category 6 - 9:30 to 10 am

category 7 - 10 am to 12 pm

category 8 - 12 pm to 5 pm

category 9 - 5 pm to 12 am

10. Hoursworkperweek: Replaced ? with a category (0).

11. Ancestryofparent: We dropped the ancestry attribute first as we thought using ancestry would not be ethical. However, for the course of this assignment we will not be considering ethics in ML. So we just Replaced ? with a category (0).

### (this will lead to a lot of categories but as we check later keeping it does not have much effect on our results so we kept the column for now and j)

12. Degreefield: N/A value represented as a category (0).

13. Industryworkedin: We clubbed the categories with similar short form such as AGR, EXT etc together so that we can decrease the number of categories for Industryworkedin. For instance,

0170 AGR-CROP PRODUCTION

0180 AGR-ANIMAL PRODUCTION AND AQUACULTURE

13. Marital: There were no missing or ? values so we left it as is.

So, after data cleaning we get the following:

Categorical Attributes: 'workerclass', 'marital', 'schoolenrollment', 'educationalattain', 'sex', 'workarrivaltime', 'degreefield', 'industryworkedin', 'ancestry'

Numerical Attributes: 'age', 'interestincome', 'traveltimetowork', 'hoursworkperweek'

**Feature Encoding:**

We plan to implement regression algorithms in order to predict the wages of each worker. Regression analysis requires numerical variables. We have many categorical attributes so we decided to perform One Hot Encoding on the categorical attributes in order to perform Regression techniques.

One hot encoding creates new (binary) columns, indicating the presence of each possible value from the original data. It is recommended to process the numerical and categorical features separately, and join them later. we performed one hot encoding on only our categorical attributes ('workerclass', 'marital', 'schoolenrollment', 'educationalattain', 'sex', 'workarrivaltime', 'degreefield', 'industryworkedin','ancestry') and joined it with our numeric dataframe.

After we performed one hot encoding our number of columns increased to 145 (the shape of encoded training dataframe being 1137\*145).

**Cross Validation & Grid Search CV:**  We perform cross validation over 3 folds on our training set and GridSearchCV to obtain the best parameters for the learning method.

GridSearchCV: Hyper-parameters are parameters that are not directly learnt within estimators. In scikit-learn they are passed as arguments to the constructor of the estimator classes.

We used GridSearchCV as it exhaustively considers all parameter combinations and retaining the best combination. We tried Grid Search CV with many different parameters for all the algorithms tried and their values.

**Models Used:**

We decided to try Linear regression first as it is the most basic and simple model used to predict a particular value.

1. Linear Regression:

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Values |
| Fit\_intercept | Whether to calculate the intercept for this mode | True, False |
| normalize | X will be normalized before regression | True, False |

Result:

Best\_score: 0.09752175838605466

Best\_params: {'fit\_intercept': False, 'normalize': False}

Thus, GridSearchCV gave us default parameter values itself. So, we applied Linear Regression method to predict the wages of the workers using default parameters which are the same as the hyperparameters obtained from GridSearchCV.

2. Random Forest

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Values |
| n\_estimators | The number of trees in the forest. | 10,20,30,40,50,60 |
| max\_features | The number of features to consider when looking for the best split | auto, sqrt, log2 |
| max\_depth | The maximum depth of the tree. | 2,4,6,8 |

Result:

Best\_score: 0.24942598656898468

Best\_params: {'max\_depth': 6, 'max\_features': 'sqrt', 'n\_estimators': 30}

3. Decision Trees

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Values |
| min\_samples\_split | The minimum number of samples required to split an internal node. | 2,3,4,5,6,7,8,9 |
| max\_features | The number of features to consider when looking for the best split | auto, sqrt, log2 |
| max\_depth | The maximum depth of the tree. | 2,4,6,8 |

Result:

Best\_score: 0.1840135809812704

Best\_params: max\_depth= 6, max\_features= 'sqrt', min\_samples\_split=4

After finding the results for hyperparameters using GridSearchCV we train our data using Cross Validation with the following models and we get the following RMSE’s.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Default Parameters | Tuned Hyperparameters | RFE | RMSE |
| Linear Regression | ✔ | ✔ |  | 66152.5 |
| Random Forest | ✔ |  |  | 62287.9 |
| Random Forest |  | ✔ |  | 61679.8 |
| Random Forest |  | ✔ | ✔ | 61451.5 |
| Decision Tree | ✔ |  |  | 83178.8 |
| Decision Tree |  | ✔ |  | 77042.8 |
| Decision Tree |  | ✔ | ✔ | 69256.5 |

As we can see that Random Forest gives us the best RMSE after applying GridSearchCV and RFE.

The RMSE for Decision Tree Regressor came out to be high (83178.8) with default parameters, but using the results from GridSearchCV and RFE we were able to bring it down to 69516.0.

**Running the Model on Test Data:**

We finally run the Random Forest model with parameters {'max\_depth': 6, 'max\_features': 'sqrt', 'n\_estimators': 30} and RFE on the test data to get our prediction of income.

Since the number of columns in the train data and the test data which will result in a difference in the total number of features after performing one hot encoding.

So we append columns with value 0 for each row with a one-hot encoded feature that is missing and drop the attribute-value pairs which are in the test data but not in the train data as the model we trained does not handle these attribute-value pairs.

**References:**

Packages Used-

1. Sklearn- <https://scikit-learn.org/stable/index.html>

2. Pandas - <https://pandas.pydata.org/>