**Author:**

* **Yan Frenklakh 211340823**

**Introduction and description of the dataset:**

**The Data Science Job Salaries Dataset contains information related to job salaries in the field of data science.**

**Description of the 11 columns present in the dataset:**

1. work\_year: This column represents the year in which the salary was paid. It provides a temporal aspect to the dataset.
2. experience\_level: This column indicates the experience level of the job during the specified year. It may include categories such as entry-level, intermediate, or senior.
3. employment\_type: This column describes the type of employment for the role, which could include full-time, part-time, contract, or freelance.
4. job\_title: The job\_title column contains the specific role or position that was worked in during the given year.
5. salary: This column represents the total gross salary amount paid to the employee for the specified year.
6. salary\_currency: The salary\_currency column indicates the currency in which the salary was paid, following the ISO 4217 currency code standard.
7. salaryinusd: This column represents the salary amount converted to USD (United States Dollars). It provides a standardized currency for comparison.
8. employee\_residence: This column specifies the primary country of residence for the employee during the work year, following the ISO 3166 country code standard.
9. remote\_ratio: The remote\_ratio column provides information about the overall amount of work done remotely. It may indicate the percentage or ratio of remote work performed.
10. company\_location: This column denotes the country of the employer's main office or contracting branch.
11. company\_size: The company\_size column represents the median number of people who worked for the company during the specified year. It provides an indication of the company's scale or workforce size.

These columns provide important attributes and context related to data science job salaries, including temporal information, job characteristics, salary details, currency conversions, location factors, and company attributes.

**Comparison of the different results abut “Data Science Job Title”:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classifier** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **GaussianNB** | **0.58** | **0.79** | **0.61** | **0.69** |
| **DecisionTreeClassifier** | **0.74** | **0.81** | **0.85** | **0.83** |
| **KNeighborsClassifier** | **0.76** | **0.78** | **0.94** | **0.85** |
| **KMeans** | 0.52 | **0.73** | **0.57** | **0.64** |

**Conclusions about the comparison table:**

* The accuracy metric indicates the overall performance of the classifier in correctly predicting instances from the "Data Science" class.

The KNeighborsClassifier achieves the highest accuracy (0.76), followed by the DecisionTreeClassifier (0.74), GaussianNB (0.58), and KMeans (0.52).

The KNeighborsClassifier has the highest overall accuracy in predicting instances of the "Data Science" class among the listed classifiers.

* Precision measures the accuracy of positive predictions made by the classifier for the "Data Science" class.

The DecisionTreeClassifier has the highest precision (0.81) among the classifiers, indicating that it has a better ability to make accurate positive predictions for the "Data Science" class.

* Recall measures the ability of the classifier to correctly identify positive instances of the "Data Science" class.

The KNeighborsClassifier achieves the highest recall (0.94), indicating that it has a better ability to capture most of the true positive instances for the "Data Science" class.

* The F1-score is the harmonic mean of precision and recall and provides a balanced measure of a classifier's performance for the "Data Science" class.

The KNeighborsClassifier has the highest F1-score (0.85), suggesting a good balance between precision and recall for that class.

**Overall Performance:**

The KNeighborsClassifier performs relatively well in terms of accuracy, precision, recall, and F1-score for the "Data Science" class. The DecisionTreeClassifier also demonstrates good precision, recall, and F1-score. GaussianNB and KMeans show relatively lower performance in terms of accuracy, precision, recall, and F1-score for the "Data Science" class.

**Data Processing and Feature Engineering summary:**

* Convert country codes to country names.

The “company\_location” and “employee\_residence” columns are transformed by applying the “country\_code\_to\_name” function, which converts country codes to their respective country names.

* Adjust salaries for inflation.

Inflation rates for the United States (us\_inflation\_rates) and global rates (global\_inflation\_rates) are defined.

The “adjust\_salary” function adjusts the “salary\_in\_usd” column based on the inflation rates for each year, considering the currency.

The adjusted salaries are stored in the “adjusted\_salary” column.

* Convert Country Names to Continents.

The “country\_to\_continent” function is defined to map country names to continent names using the “pycountry\_convert” library.

The “company\_location” and “employee\_residence” columns are transformed to represent continents.

* Categorize Salaries into Bins.

The salaries are divided into four bins based on quantiles.

The “salary\_range” column is created to indicate the salary range for each entry.

* Dropping Unnecessary Columns.

Columns that are no longer needed (“salary”, “salary\_currency”, “job\_title”, “work\_year”, “salary\_in\_usd”) are dropped.

**Preprocessing steps conclusions:**

* The preprocessing steps involve converting country codes to country names (“company\_location” and “employee\_residence”).

This conversion enhances the interpretability of the data, as country names are more easily understandable than codes.

* The conversion of country names to continents (“company\_location” and “employee\_residence”) enables the analysis to focus on broader geographical regions rather than specific countries.

This information can be useful for examining salary trends or conducting comparative analysis across continents.

* The adjustment of salaries based on inflation rates provides a more accurate representation of salary values in different years.

y considering inflation, the analysis accounts for the changing value of money over time and ensures fair comparisons across different years.

* The categorization of salaries into bins based on quantiles (low, mid, high, Top) allows for a simplified representation of salary ranges.

This categorization helps in analysing salary distributions and identifying different salary tiers.

* The removal of unnecessary columns (“salary”, “salary\_currency”, “job\_title”, “work\_year”, ”salary\_in\_usd”) reduces the dimensionality of the data and focuses the analysis on the relevant features.

These preprocessing steps aim to enhance the quality and usability of the data by converting and adjusting variables, facilitating visualization, and simplifying the representation of salary information.

They provide a solid foundation for further analysis and building predictive models.