# CA3 Report

## Human Centric Deep Learning

*Beatrice Cipriani & Michalis Thomas*

# Introduction (M)

Through this project, we aimed to take a look and analyze how different job titles, experience levels and employment conditions affect the potential salary of professionals within the Data Analysis space. The purpose of this project was to use existing data from the past 4 years to try to develop and compare different models' performances, in predicting the salary range of a given job position, but most importantly compare the different types of models' explainability when predicting salary ranges. The models that were tested were Decision Trees as well as Neural Networks. Three Decision Trees were tested that had no max depth, max depth of 3 and max depth of 2, meanwhile multiple Neural Networks were tested and tuned with only the best performing model being chosen for the comparison.

# Methods

## Data description and pre-processing (M)

Dataset used was over 14000 entries of salaries for Data Analysis related jobs over the past 4 years (2020-2024 so far). In the dataset each entry contained information on the job title, the company and employee location, the ratio of remote work, the original currency and salary, salary in US dollars, employment type and also the experience level.

In order to proceed to start developing the models, we had to do some basic preprocessing of the dataset for it to be usable. The first action was to drop some unnecessary columns/features. These aforementioned columns were the 'employment_type', 'salary' and 'salary_currency'. The 'employment_type' column was removed because it described whether the job was full time or part time, and every entry in the dataset was full-time. 'Salary' and 'salary_currency' were removed because the data of the salary were inconsistent with each other since they were measured in different currencies, so we decided to only use the 'salary_in_usd' column to measure and predict salaries. The next step was to create a category for 'remote_work' that had three categories, 'No', 'Yes' and 'Hybrid'. This column was generated from the 'remote_ratio' column, where values 0-20 were converted to 'No', values 20-80 were converted to

‘Hybrid’ and values 80-100 were converted to ‘Yes’. After that was done the ‘remote_ratio’ column was removed. Following that, a new ‘salary’ column was created where the values of ‘salary_in_usd’ were converted into categories. All the values of ‘salary_in_usd’ that were 0-50000, took the category value ‘Low’, 50001-175000 were categorized as ‘Medium’ and anything over 175001 were categorized as ‘High’.

Moreover, the ‘company_loc’ and ‘employee_loc’ columns needed to be recategorized and  converted into two new columns because most of the entries had the value of ‘US’ and also there were too many distinct values that did not appear consistently throughout the dataset. The values for ‘US’ and ‘GB’ stayed as they were, but everything else was categorized as ‘Others’. Then the ‘company_loc’ and ‘employee_loc’ columns were dropped. Additionally a new column for the categories of experience (‘experience_cat’) was created by converting the values from ‘experience_level’. Firstly, the ‘SE’ and ‘EX’ were categorized both as ‘Senior’, ‘MI’ as ‘Middle’ and everything else was categorized as ‘ Entry’. Then the ‘experience_level’ column was dropped.

After the basic preprocessing was done, we proceeded to identify the most common and highest average salary jobs. Using the two graphs below and some more analyzing of the data it was concluded that the jobs with the highest average salary were jobs that only appeared very few times in the dataset and had extremely high salaries, so it was decided that those values were not representative of the real highest averaging job titles in the dataset, so we proceeded to focus on the most common jobs, which had a high number of entries, thus providing a more accurate and realistic average.

## Graphs (M)

Looking at the graph in *fig.3* we can see how the average salary, and the range of the salary, for the 5 most common job titles over the past 4 years changes. We notice that there is an overall increase in the average salary. From *fig. 4* we can see how having remote work condition in the most common jobs affects the average salary. From the graph we conclude that if there is no remote work the average salary is slightly higher than if there is remote work. In addition hybrid showcases the lowest average, however it should be taken into serious consideration that there are significantly less entries with hybrid remote work values inside the dataset, thus it is not necessarily an accurate representation.

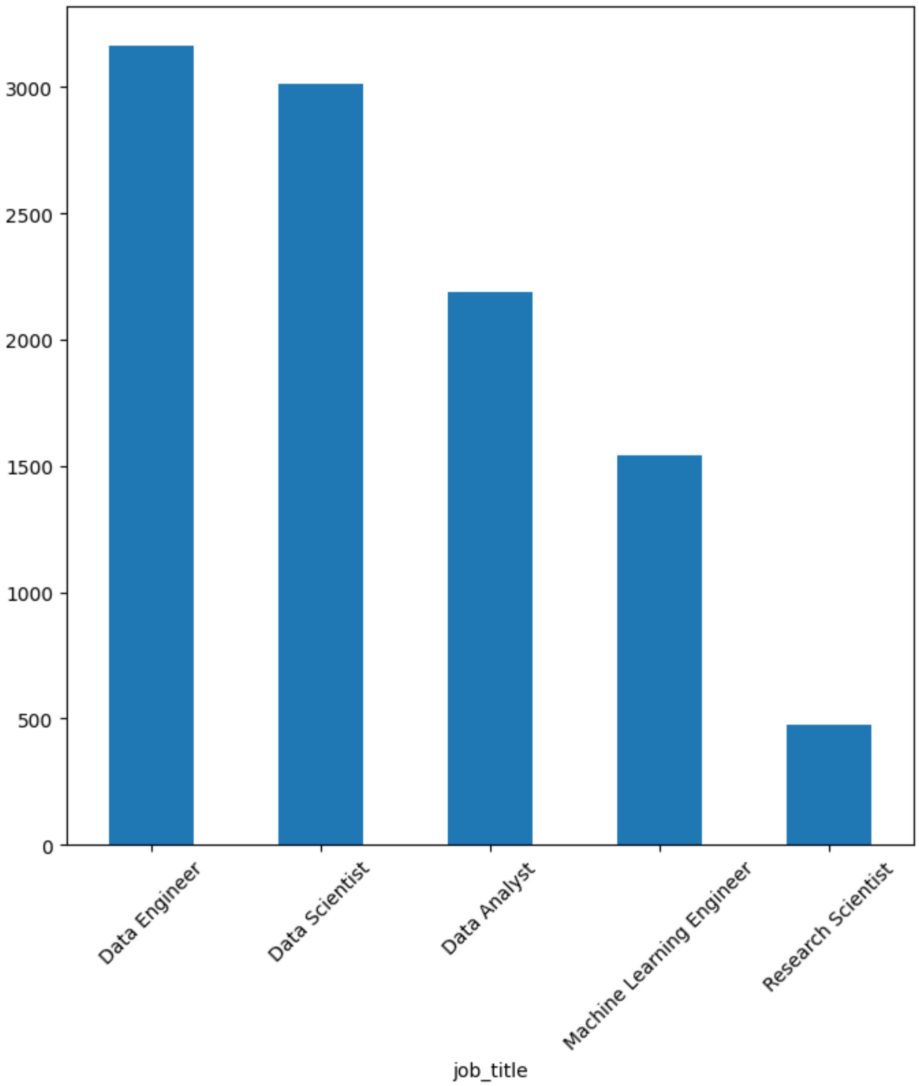
3000
2500
2000
1500
1000
500
0
Data Engineer
Data Scientist
Data Analyst
Machine Learning Engineer
Research Scientist
job_title

**Fig.1**

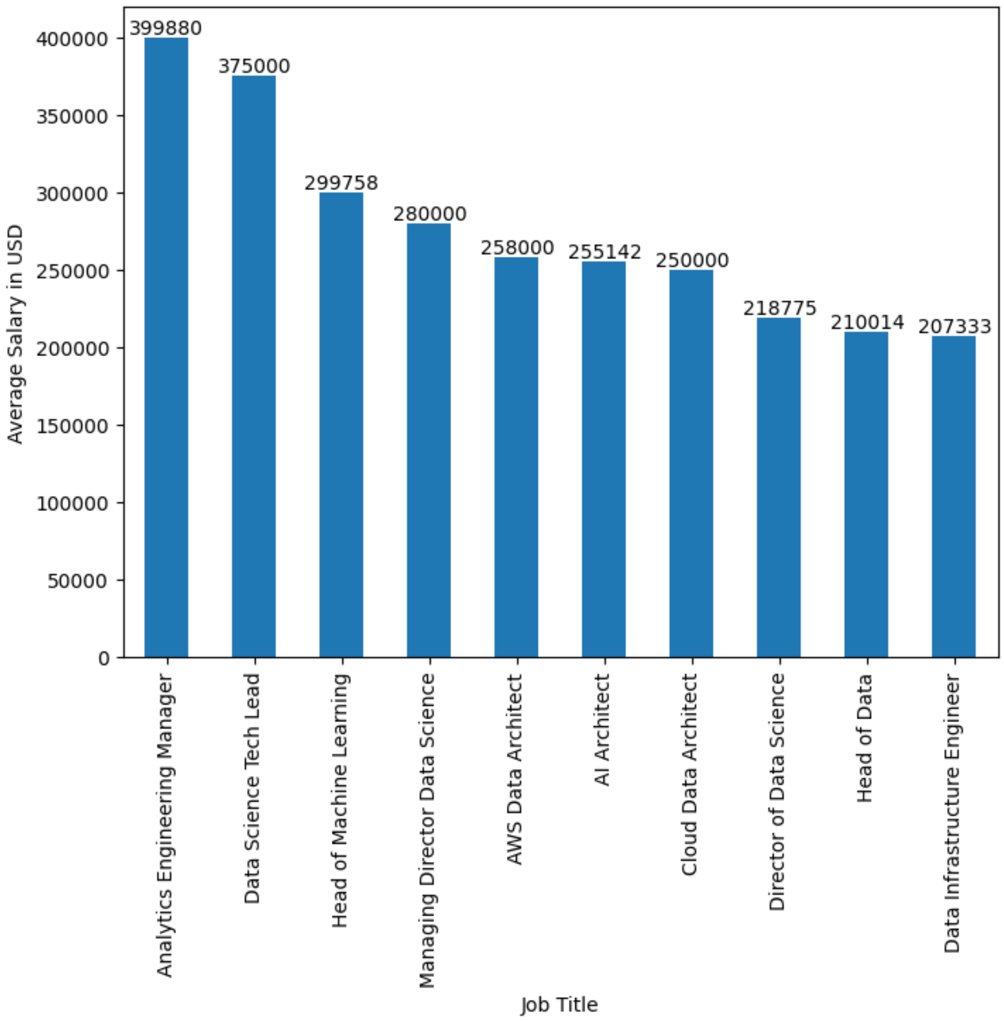
399880
375000
299758
280000
258000
255142
250000
218775
210014
207333
400000
350000
300000
250000
200000
150000
100000
50000
0
Average Salary in USD
Analytics Engineering Manager
Data Science Tech Lead
Head of Machine Learning
Managing Director Data Science
AWS Data Architect
AI Architect
Cloud Data Architect
Director of Data Science
Head of Data
Data Infrastructure Engineer
Job Title

**Fig.2**

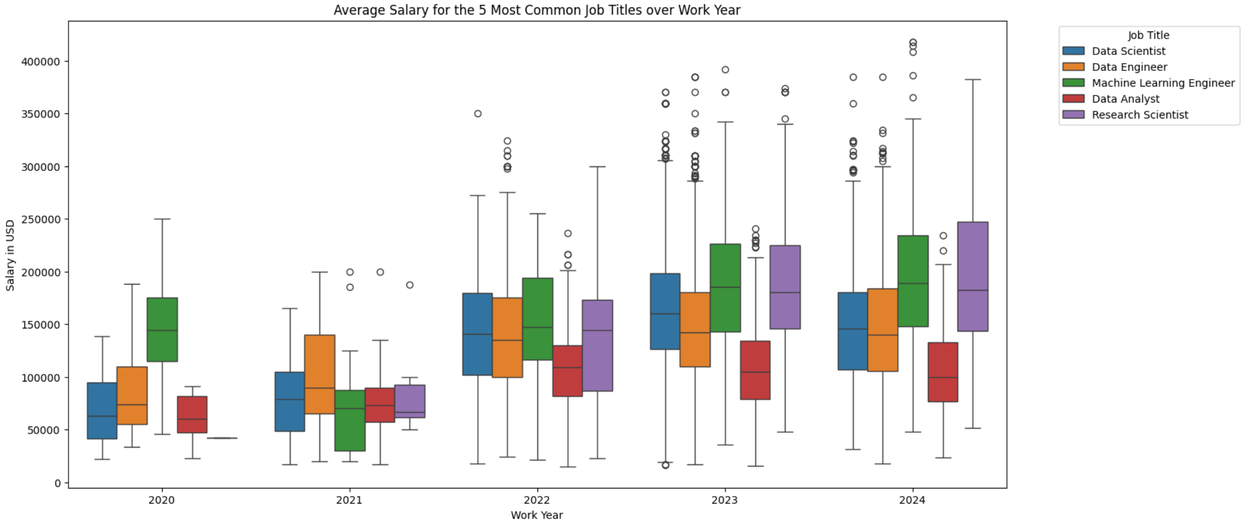
Average Salary for the 5 Most Common Job Titles over Work Year
400000
350000
300000
250000
200000
150000
100000
50000
0
Salary in USD
2020
2021
2022
2023
2024
Work Year
Job Title
Data Scientist
Data Engineer
Machine Learning Engineer
Data Analyst
Research Scientist

**Fig.3**

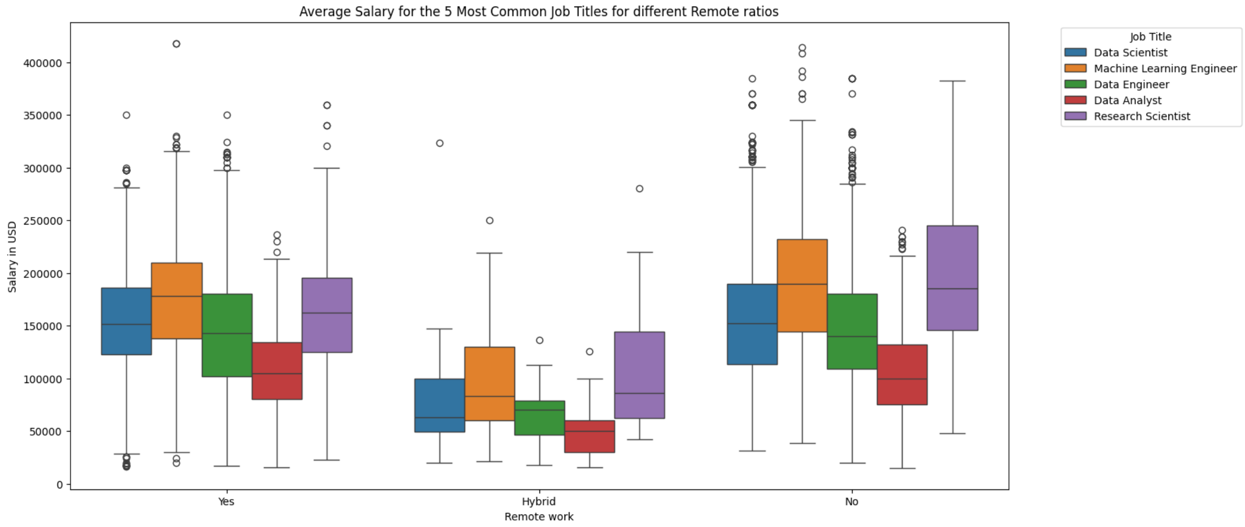

**Fig.4**

## Decision Tree (B)

In order to predict the salary rank (low-medium-high), the first model developed for the classification consists in a machine learning approach with Decision Tree algorithm. As the name suggests, the decision is performed according to a tree-like structure, where each node corresponds to a decision based on one of the 7 attributes in our dataset. The advantages of this algorithm are its simplicity and human interpretability, still capturing non-linear relationships of the attributes. On the other hand, a well known limitation of DT consists in their tendency to overfitting. To overcome and evaluate this problem, we developed two classification models with different depth control: none and 2. This decision also allows a better insight on the model's robustness, generalization and explainability. After, we implemented in the model the Explainable Boosting Machines (EBMs), a ML model that provides clear insights in interpretability, particularly in understanding the most important features in the decision-make process.

## Neural Network (B)

The second model we developed a Neural Network (NN) consisting of multiple layers and nodes. This kind of method tends to perform better with complex datasets, where the relationship between the features is non-linear and hardly depictable. For the NN, we have been using the normalized dataset, and the predictions were encoded into categorical values. In the development of the architecture first different depths (number of layers) and then widths (number of neurons) have been tested. After assessing the loss and accuracy of each architecture, we noticed that deeper architectures exhibited poorer performance compared to smaller ones. The final chosen architecture was kept to include only one hidden layer, along with the input and output layers, with 100, 50 and 3 neurons, respectively. The activation functions for the output layer have been evaluated between ReLU, Sigmoid, Softmax and TanH. Surprisingly, the best performance has been obtained with the Sigmoid function, normally used for binary classification problems.

Following the tuning of hyperparameters through a grid search, the following parameters were tested:

```
optimizers = ['rmsprop', 'adam', 'sgd']
batches = [1, 16, 32, 64, 128]
inits = ['normal', 'uniform']
epochs = [200]
```

Notably, higher batch sizes (256, 512, 1000, and 10000) were deliberately omitted from the grid search due to their prior testing. Additionally, early stopping was applied during the training process.

The final model has been interpreted using LIME and SHAP algorithms, in order to offer complementary insights into the workings of the NN. Briefly, LIME stands for Local Interpretable Model-agnostic Explanations that, as the name suggests, explains the prediction of the model by performing perturbations around a local prediction and measuring the effects of the original dataset's features. The SHAP algorithm, that stands for SHapley Additive exPlanations, assigns to each feature a score for the prediction, with an approach based on gaming theory. These two methods have been applied on the developed NN to evaluate its reliability and trustiness.

## Results and Explainability discussion (B)

The decision tree without depth control yielded an accuracy of 68.43%, while restricting the depth to 2 resulted in a slightly improved accuracy of 68.67%. Employing the EBM algorithm further boosted accuracy to 69.68%.

The decision tree without depth control is more intricate, as shown in Fig.5, containing a much larger number of nodes in its architecture. For this reason, its interpretability is more challenging. In contrast, the decision tree with a maximum depth of 2 (Fig.7) is considerably easier to interpret. Before that, a maximum depth of 3 (as shown in Fig.6) has been tested, which did not significantly enhance accuracy compared to the first model. Among the three models tested, the higher accuracy achieved by the model coincided with improved explainability.

Figure 7 illustrates the classification logic employed by the model with maximum depth 2. According to the dataset, an experience rating of 1 indicates mid-level, 2 corresponds to senior-level, and 0 signifies entry-level. Instances scoring below 1.5, thus below senior level, are categorized as medium income. Similarly, considering employee location, instances below 1.5 are classified under the medium income category. Furthermore, for job titles, instances above 96.5 are classified as high income. Notably, the model with a maximum depth of 2 does not classify any jobs as low income, suggesting potential bias in the classification process when compared to the model with a maximum depth of 3.

By performing EBM analysis, we find that the most influential feature identified is the job title, followed by the company's location.

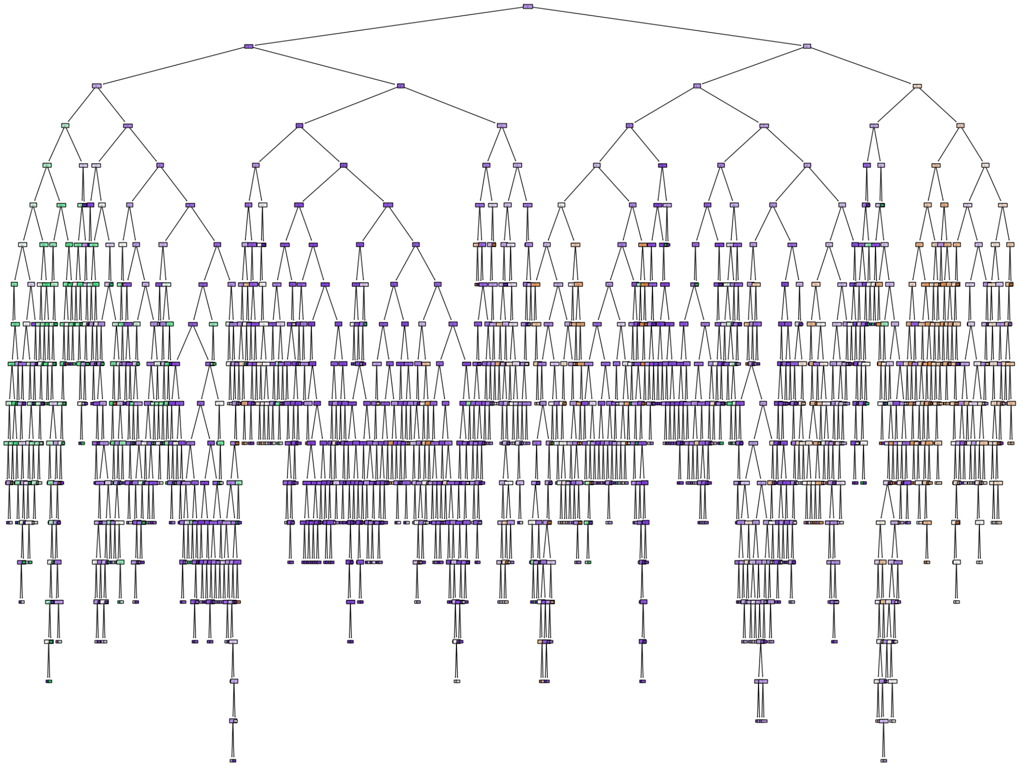

**Fig.5**

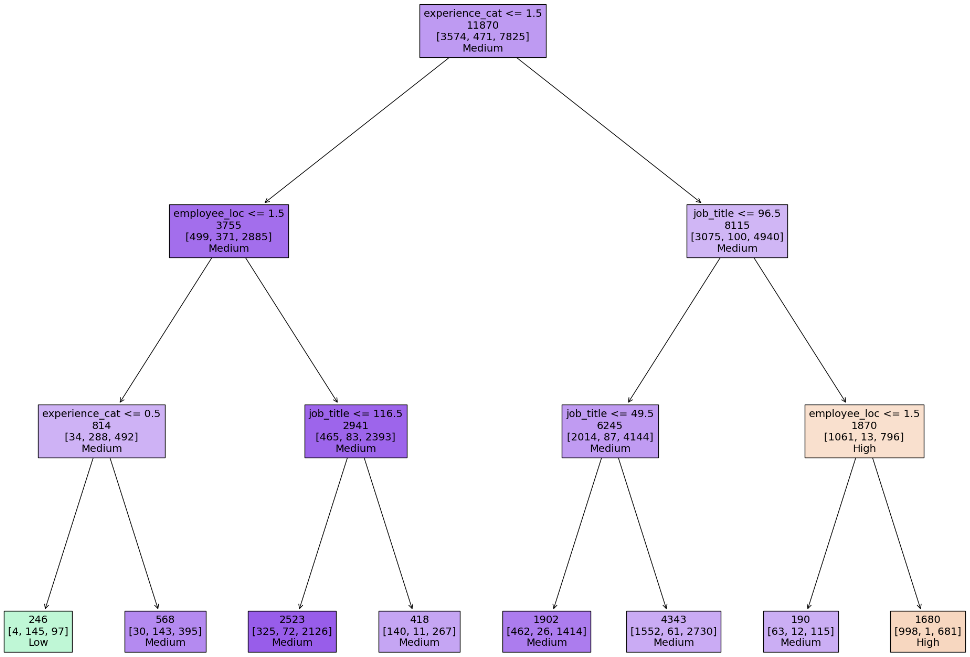
experience_cat <= 1.5
11870
[3574, 471, 7825]
Medium
employee_loc <= 1.5
3755
[499, 371, 2885]
Medium
job_title <= 96.5
8115
[3075, 100, 4940]
Medium
experience_cat <= 0.5
814
[34, 288, 492]
Medium
job_title <= 116.5
2941
[465, 83, 2393]
Medium
job_title <= 49.5
6245
[2014, 87, 4144]
Medium
employee_loc <= 1.5
1870
[1061, 13, 796]
High
246
[4, 145, 97]
Low
568
[30, 143, 395]
Medium
2523
[325, 72, 2126]
Medium
418
[140, 11, 267]
Medium
1902
[462, 26, 1414]
Medium
4343
[1552, 61, 2730]
Medium
190
[63, 12, 115]
Medium
1680
[998, 1, 681]
High

**Fig.6**

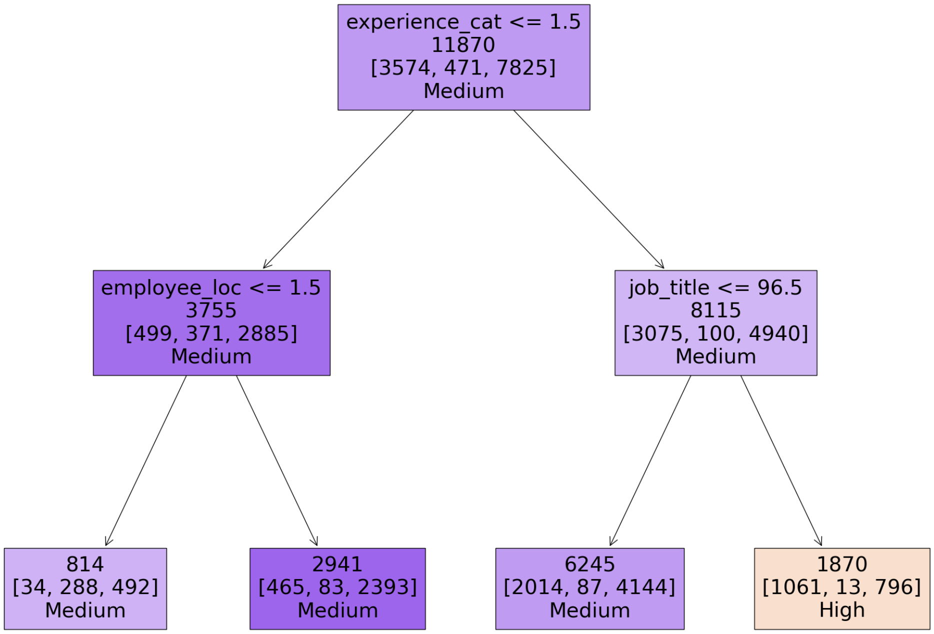
experience_cat <= 1.5
11870
[3574, 471, 7825]
Medium
employee_loc <= 1.5
3755
[499, 371, 2885]
Medium
job_title <= 96.5
8115
[3075, 100, 4940]
Medium
814
[34, 288, 492]
Medium
2941
[465, 83, 2393]
Medium
6245
[2014, 87, 4144]
Medium
1870
[1061, 13, 796]
High

**Fig.7**

The neural network exhibited a consistent performance, yielding an accuracy of 67% prior to hyperparameter tuning. However, the results from the subsequent grid search were inconclusive, as multiple parameter combinations produced identical accuracies. Notably, the best accuracy of 66.19% was achieved with `{'batch_size': 1, 'epochs': 200, 'initIn': 'normal', 'optimizerIn': 'rmsprop'}`, but also:

```
0.661870 (0.378488) with: {'batch_size': 1, 'epochs': 200,
'initIn': 'normal', 'optimizerIn': 'rmsprop'}
0.661870 (0.378488) with: {'batch_size': 1, 'epochs': 200,
'initIn': 'normal', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 1, 'epochs': 200,
'initIn': 'uniform', 'optimizerIn': 'rmsprop'}
0.661870 (0.378488) with: {'batch_size': 1, 'epochs': 200,
'initIn': 'uniform', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 16, 'epochs': 200,
'initIn': 'normal', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 16, 'epochs': 200,
'initIn': 'uniform', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 32, 'epochs': 200,
'initIn': 'normal', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 32, 'epochs': 200,
'initIn': 'uniform', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 64, 'epochs': 200,
'initIn': 'normal', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 64, 'epochs': 200,
'initIn': 'uniform', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 128, 'epochs': 200,
'initIn': 'normal', 'optimizerIn': 'sgd'}
0.661870 (0.378488) with: {'batch_size': 128, 'epochs': 200,
'initIn': 'uniform', 'optimizerIn': 'sgd'}
```

indicating potential limitations of the model.

This lack of improvement may suggest that the model has reached its capacity to learn from the available data. Such limitations were further underscored by the LIME analysis, which revealed consistently low prediction probabilities (~20%) for randomly selected predictions. This suggests that the model's predictive power may be inherently constrained, potentially due to factors such as data quality or pre-processing.

Moreover, the SHAP analysis is in good agreement with the findings from the EBM analysis, highlighting job title as the most influential predictor, followed by company size.

Despite efforts to optimize the model, the final accuracy remained at 67.3%, consistent with the performance of DT models.

# Conclusion (B)

The Data Science income dataset served as the basis for testing various machine learning models and explainability algorithms. Despite employing decision trees (DT) and neural networks (NN) with varying depths and architectures/hyperparameters, both models yielded remarkably similar accuracies. This consistency across models suggests a plateau in achievable accuracy, indicating that further enhancements may necessitate alternative dataset preprocessing or consideration of additional features.

Analysis of the DT models revealed a notable bias towards categorizing predictions as medium income, implying potential imbalance in the representation of low and high income classes. The EBM and SHAP analyses reinforced the importance of job title and company size as primary predictors, while other features exhibited minimal influence on predictions, consistent with observations from the DT architecture.

The NN, while comparable in accuracy to the DT models, incurred higher computational costs during both training and architectural design and hyperparameter tuning. Additionally, the LIME analysis of the NN indicated a lack of robustness in the model's predictions.

In summary, two key observations emerge:

1. Regardless of the model choice, prediction accuracy remained consistent, suggesting inherent limitations in the dataset's representational capacity.
2. Persistent bias towards one class (low income) across all models and predominant reliance on one or two features for classification underscore the dataset's processing limitations.

These observations underscore the importance of explainability techniques in addressing dataset-related challenges and guiding model development, highlighting the need for further exploration and refinement in dataset processing to overcome existing limitations.