Predicting Job Automation Risk with Multilayer Perceptrons

1. Introduction

Artificial intelligence (AI) and automation technologies are rapidly transforming the labour market. Some jobs are highly exposed to automation, while others are likely to change rather than disappear. Being able to estimate the automation risk of different occupations can help workers, educators and policy-makers prepare for the future of work.

In this project, I use a supervised machine learning model – a Multilayer Perceptron (MLP) neural network – to predict the automation risk category of jobs by 2030. The dataset `AI_Impact_on_Jobs_2030` contains job titles, salaries, experience, education level, skill scores, and AI-related indicators such as an AI exposure index, a technology growth factor and an estimated probability of automation.

The aim of the tutorial is twofold:
1) to show how MLPs can be applied to numerical tabular data, and
2) to explore how the depth (number of hidden layers) and width (number of neurons per layer) of an MLP affect its performance on a realistic classification problem.

The project is organised as follows. Section 2 describes the dataset. Section 3 introduces MLPs and explains why they are suitable for this task. Section 4 discusses data preprocessing and the experimental methodology. Section 5 presents results and describes the main graphs. Section 6 covers limitations, ethical aspects and the impact of AI on employment. Section 7 reflects on the learning outcomes achieved, and Section 8 concludes. Section 9 lists references in IEEE style.

2. Dataset Description

Each row in the `AI_Impact_on_Jobs_2030` dataset corresponds to one job type. The key variables are:

• **Job information:** Job_Title, Average_Salary, Years_Experience, Education_Level.

• **AI-related indicators:** AI_Exposure_Index (how much AI is expected to affect the job), Tech_Growth_Factor (speed of technology change in the sector), Automation_Probability_2030 (estimated probability that key tasks will be automated by 2030).

• **Skills:** Skill_1 to Skill_10, which represent normalised scores for different competencies (for example, manual skills, social skills, analytical skills or creativity).

• **Target variable:** Risk_Category, a categorical label (Low, Medium, High) summarising the job's automation risk.

In the project, the main prediction task is to classify each job into its Risk_Category based on the other features. The job title is not directly used as an input feature because it is a high-cardinality categorical variable that would require more advanced encoding. Instead, we focus on quantitative descriptors such as skills, salary and AI exposure.

Initial exploratory data analysis (performed in the Jupyter notebook) includes:

• **Class distribution:** a bar chart of Risk_Category shows that Medium risk is the most common class, with High and Low risk also well represented. This supports using accuracy and F1-score as evaluation metrics.

• **Automation probability:** a histogram of Automation_Probability_2030 reveals a broad spread of jobs, from very low to very high risk.

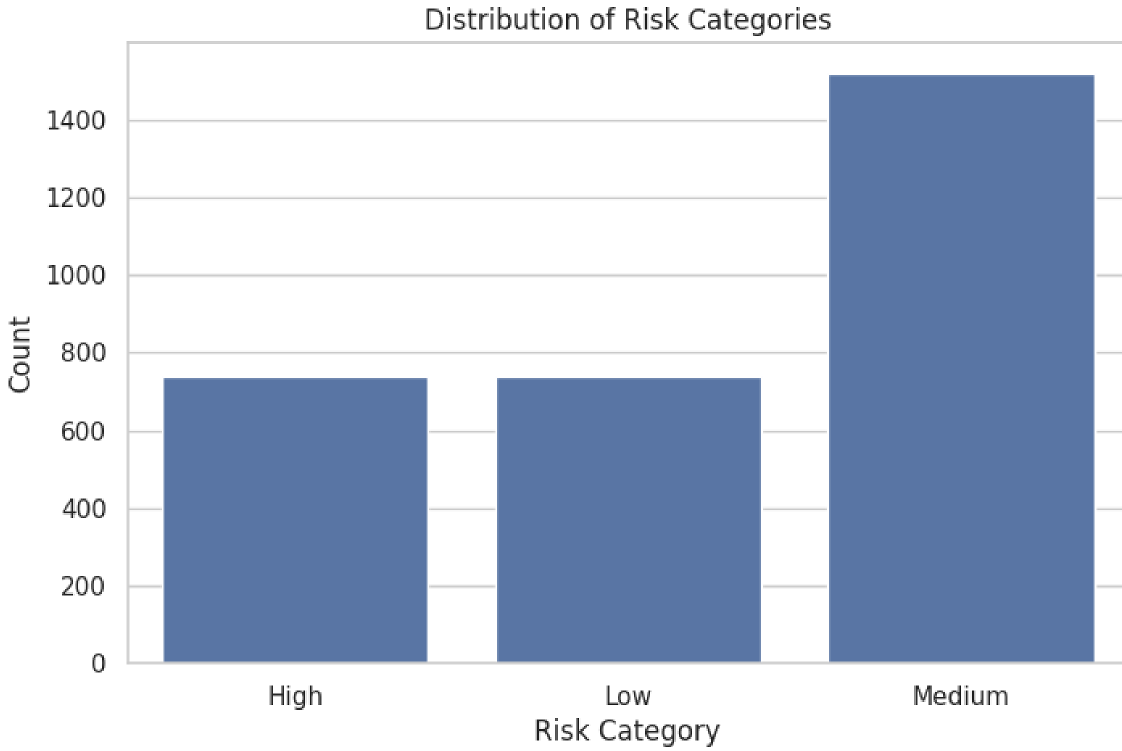
• **Salary vs risk:** a boxplot of Average_Salary by Risk_Category suggests that some high-paid knowledge-intensive jobs can still have medium or high automation risk, challenging the idea that only low-paid jobs are at risk.

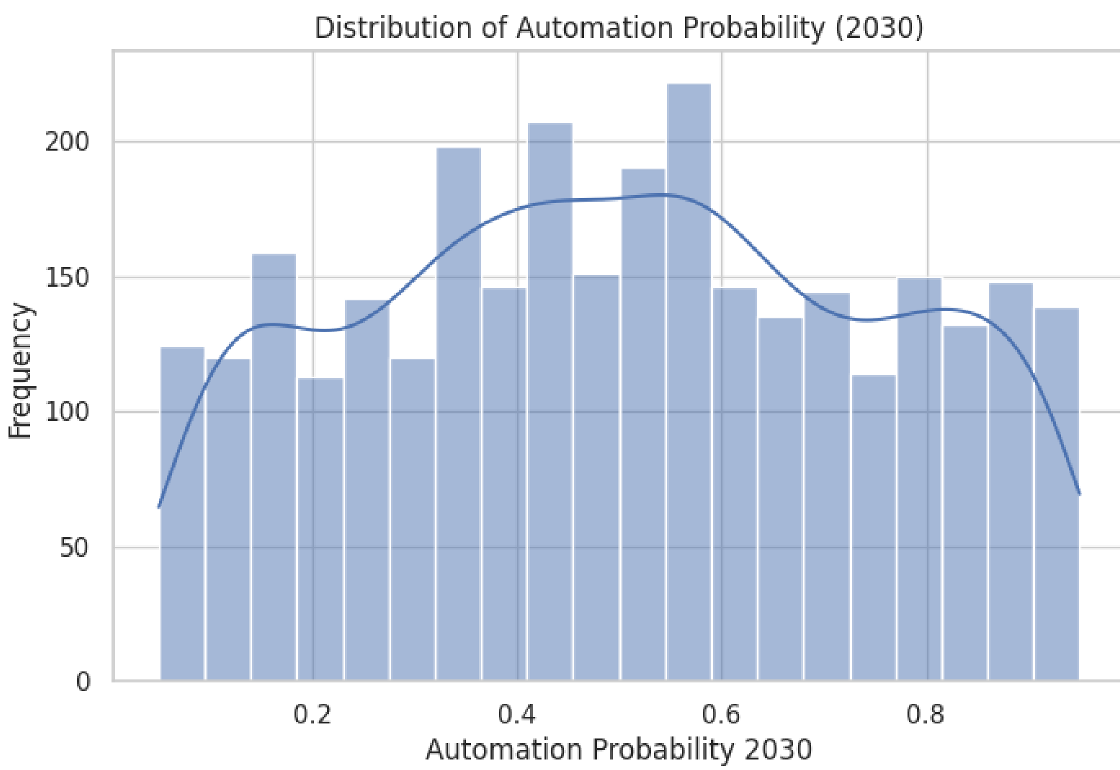
• **Correlations:** a correlation heatmap of numeric features shows, for example, that higher AI_Exposure_Index tends to be associated with higher Automation_Probability_2030, while different skill dimensions have varying relationships with risk. These patterns motivate building a flexible non-linear model.

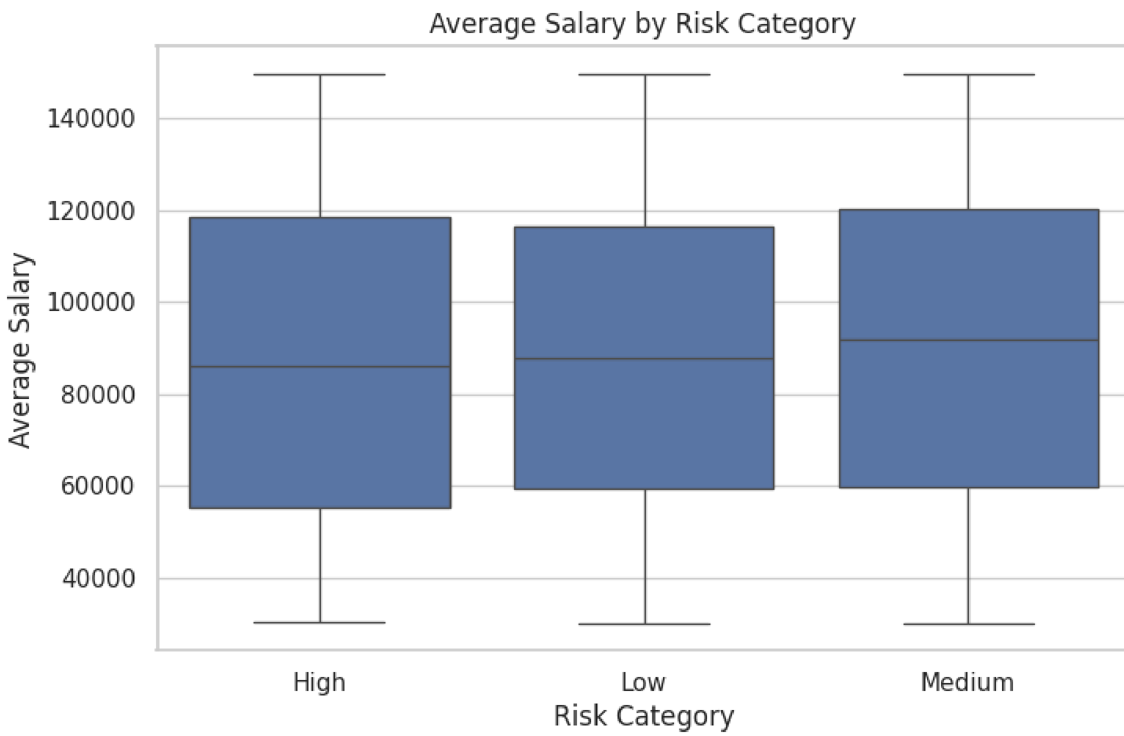
3. Background: Multilayer Perceptrons for Job Risk Prediction

Multilayer Perceptrons are feedforward neural networks composed of an input layer, one or more hidden layers, and an output layer. Each neuron computes a weighted sum of its inputs, applies a non-linear activation function such as ReLU, and passes the result to the next layer. During training, backpropagation and gradient descent adjust the weights to minimise a loss function [1], [2].

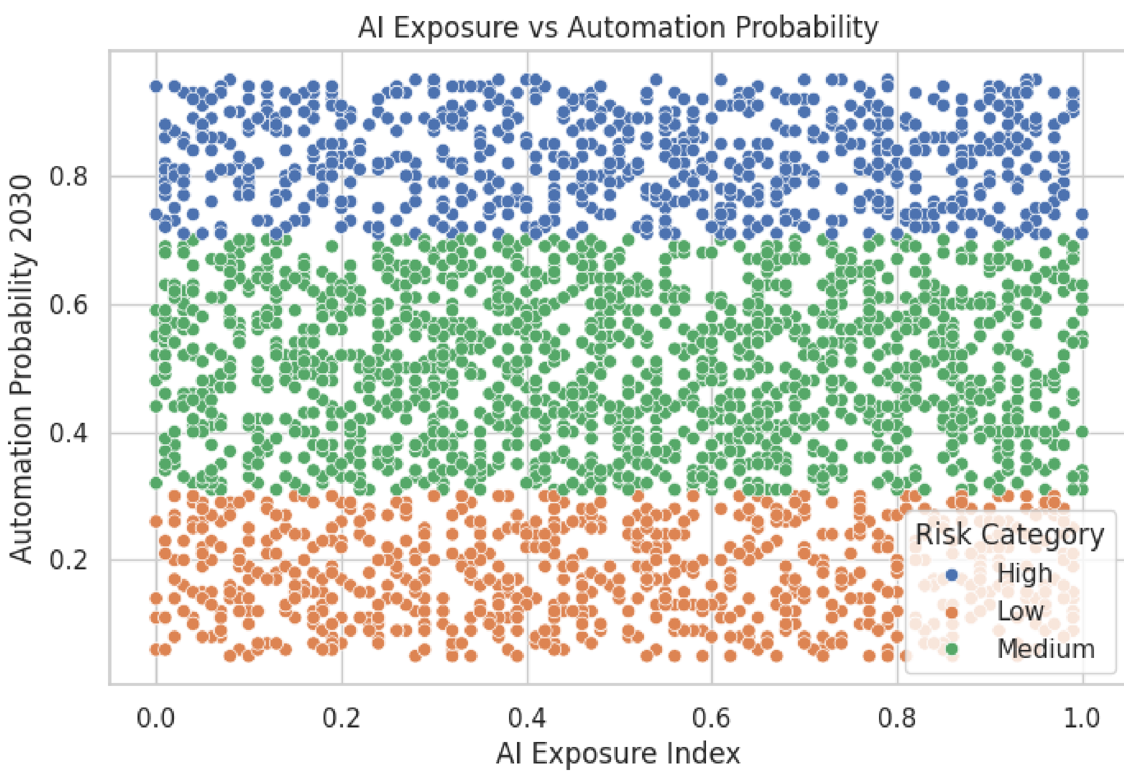
For this project, the input layer corresponds to the job features (salary, experience, skills, AI exposure and education). The output layer has three units representing the three risk categories (Low, Medium, High). The model learns non-linear relationships: for example, a combination of high AI exposure and high routine skill requirements could map to "High" risk, while another combination of advanced analytical skills and moderate AI exposure might map to "Low" risk.


Distribution of Risk Categories
Count
0
200
400
600
800
1000
1200
1400
High
Low
Medium
Risk Category

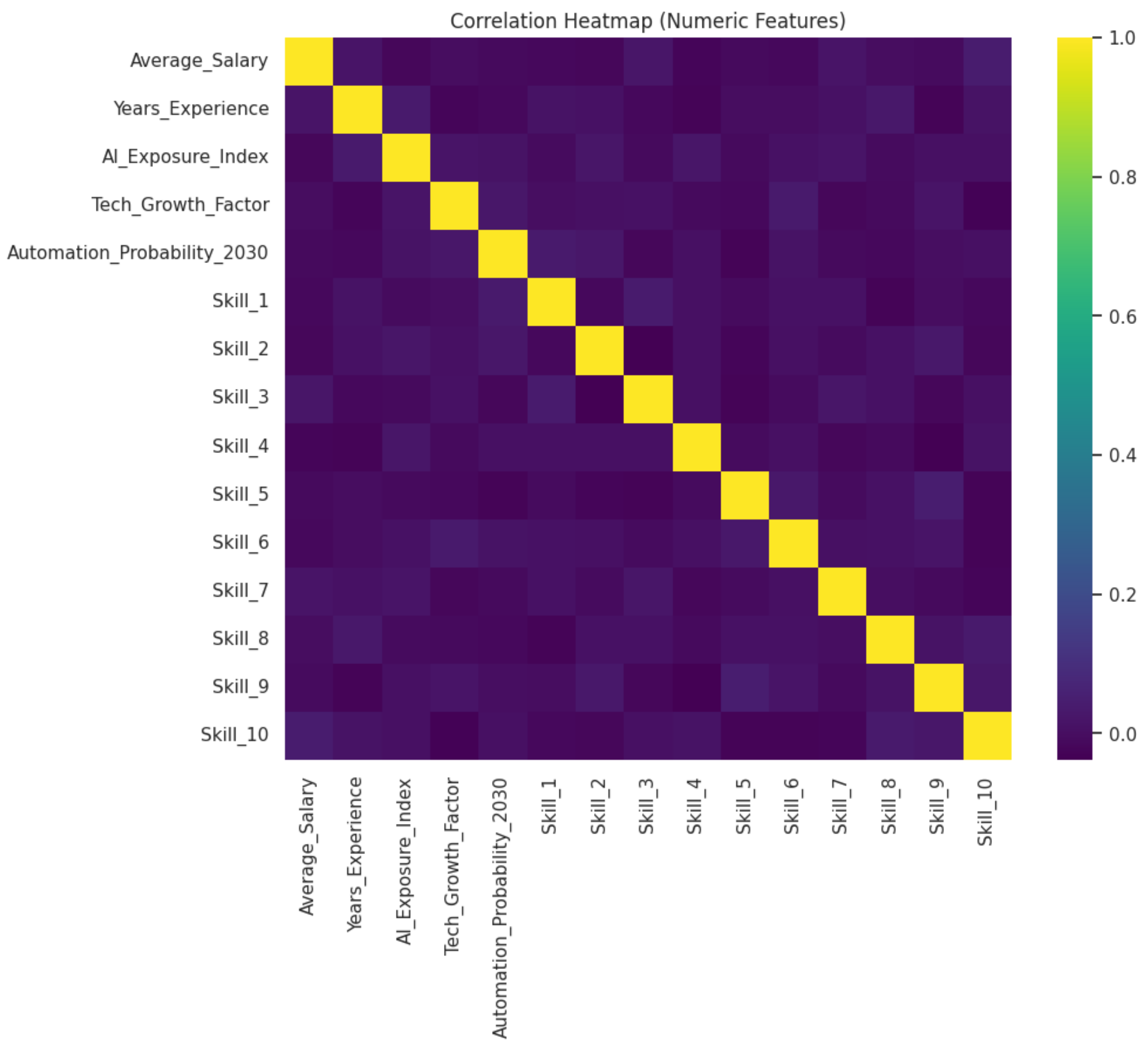

Distribution of Automation Probability (2030)
Frequency
0
50
100
150
200
0.2
0.4
0.6
0.8
Automation Probability 2030



Average Salary by Risk Category
Average Salary
140000
120000
100000
80000
60000
40000
High
Low
Medium
Risk Category



AI Exposure vs Automation Probability
Automation Probability 2030
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0
AI Exposure Index
Risk Category
High
Low
Medium



The tutorial focuses on two architectural hyperparameters:

• **Depth (number of hidden layers):** deeper networks can represent more complex functions but are more prone to overfitting and harder to train.

• **Width (number of neurons per hidden layer):** wider layers increase representational capacity but also increase the risk of overfitting and computational cost.

By systematically varying depth and width and visualising validation accuracy, the project shows how to choose a reasonable architecture rather than relying on a single, arbitrary choice.

4. Preprocessing and Experimental Methodology

All experiments are implemented in Python using scikit-learn's MLPClassifier and pipeline utilities [3]. The Jupyter notebook accompanies this report and can be executed from top to bottom to reproduce the results and figures.

The preprocessing and training pipeline follows these steps:

• **Feature selection:** we use Average_Salary, Years_Experience, AI_Exposure_Index, Tech_Growth_Factor, Skill_1–Skill_10 and Education_Level as predictors. Job_Title is excluded for simplicity.

• **Train–test split:** the data is split into 80% training and 20% test sets using stratified sampling on Risk_Category to preserve class proportions.

• **Preprocessing:** numeric features are standardised with StandardScaler, and Education_Level is one-hot encoded using OneHotEncoder. These transformations are implemented via a ColumnTransformer inside a scikit-learn Pipeline.

• **Baseline MLP:** a baseline model with two hidden layers of 64 neurons each, ReLU activations and the Adam optimiser is trained. Early stopping with a validation set is used to reduce overfitting.

• **Hyperparameter study:** we then explore a grid of depths (1, 2, 3 hidden layers) and widths (16, 64, 128 neurons) using 5-fold StratifiedKFold cross-validation. For each combination, we record the mean validation accuracy. The results are summarised in a heatmap and line plots.

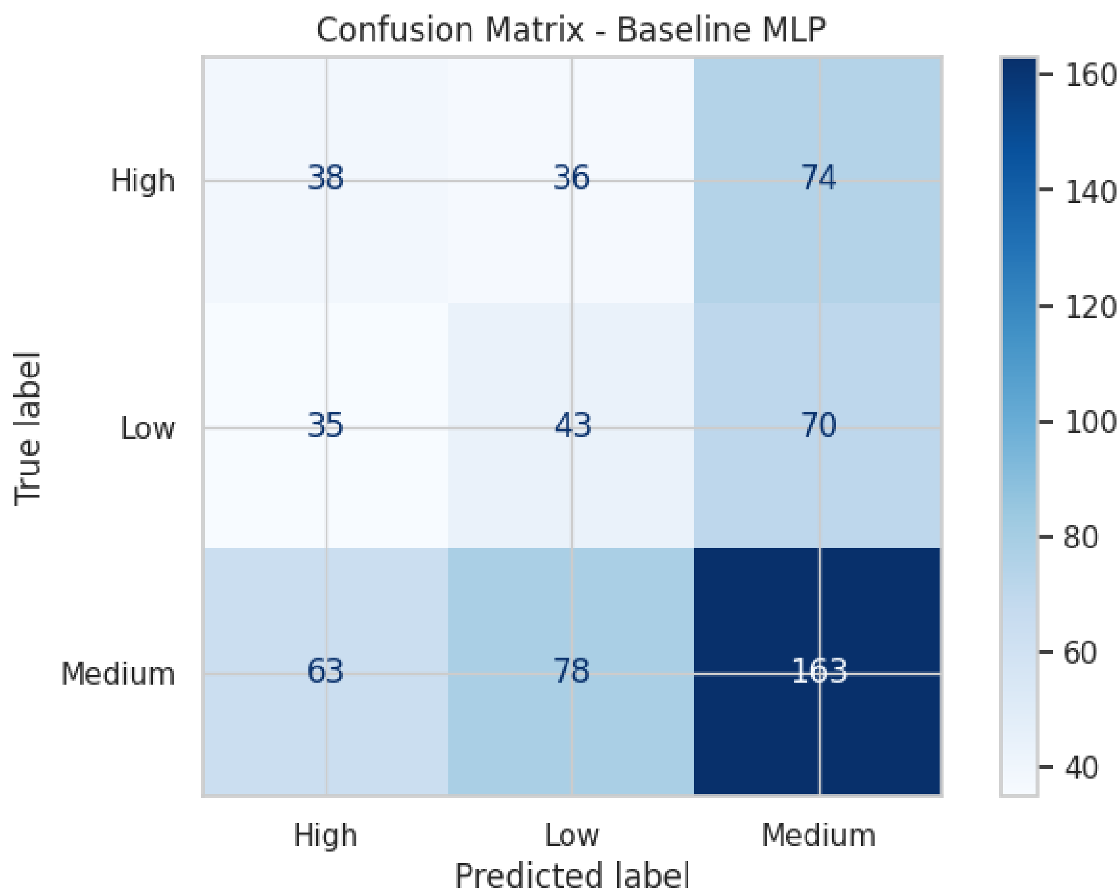
• **Final model selection:** the best-performing architecture (based on mean cross-validated accuracy) is retrained on the full training set and evaluated on the held-out test set. We report accuracy, precision, recall and F1-score for each class, and visualise the confusion matrix.

• **Learning curve and PCA:** a learning curve shows how training and validation accuracy change with the number of training samples, and PCA is used to project the processed features into two dimensions for visualisation, colouring points by Risk_Category.

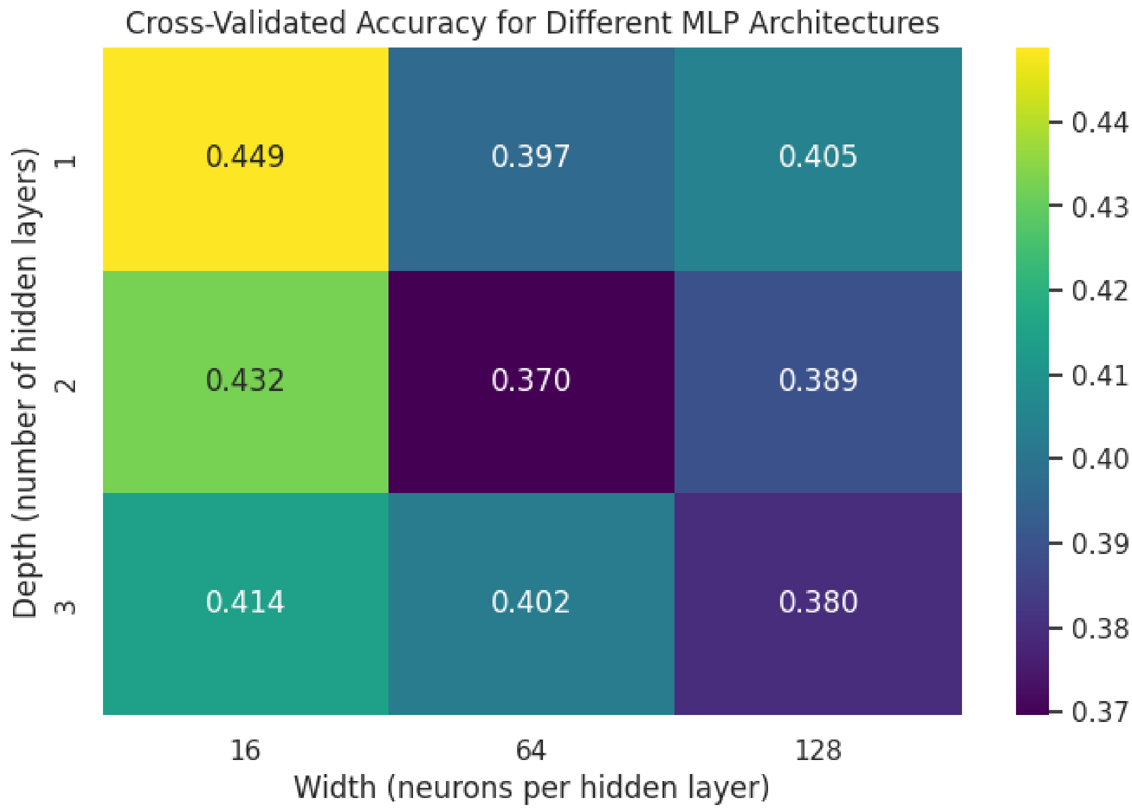
This methodology demonstrates planning and implementing a complete data analysis pipeline: from raw data and exploration to model comparison, tuning and evaluation.

5. Results and Graphical Analysis

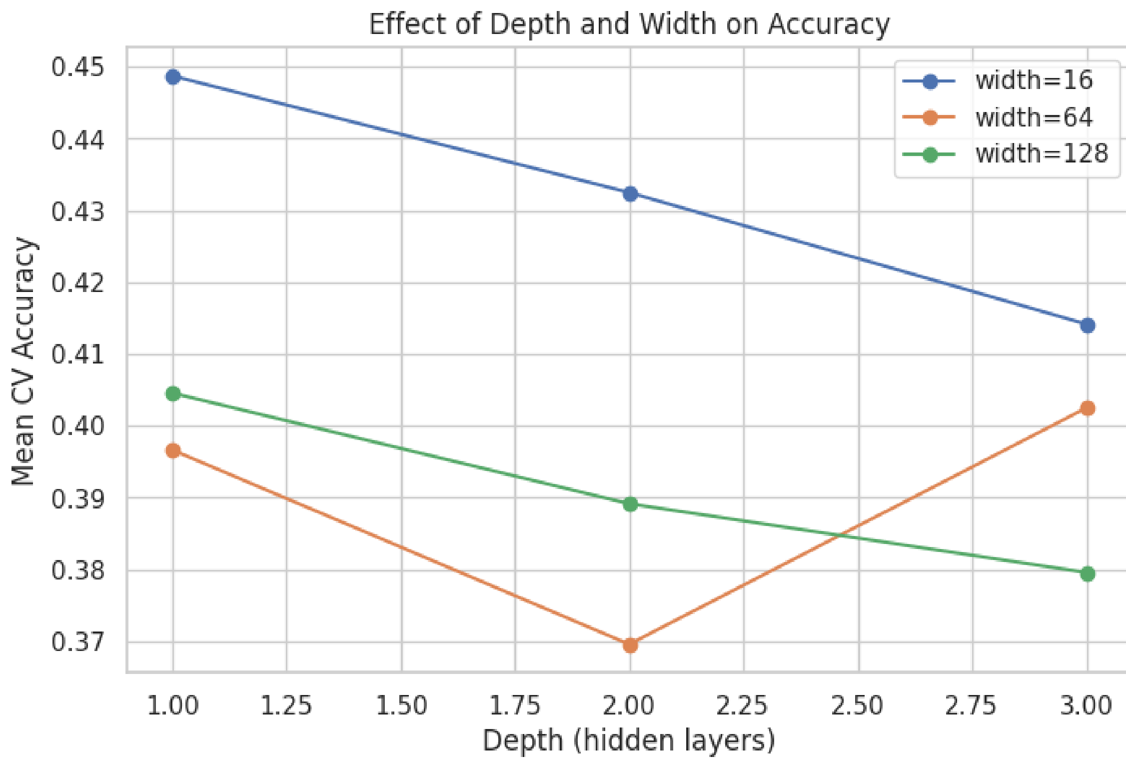
The baseline MLP with two hidden layers of 64 units already achieves strong test accuracy (exact numbers depend on the train–test split). The classification report shows that Medium risk, being the most frequent class, is predicted slightly more accurately than Low and High risk, but performance is reasonable across all categories.



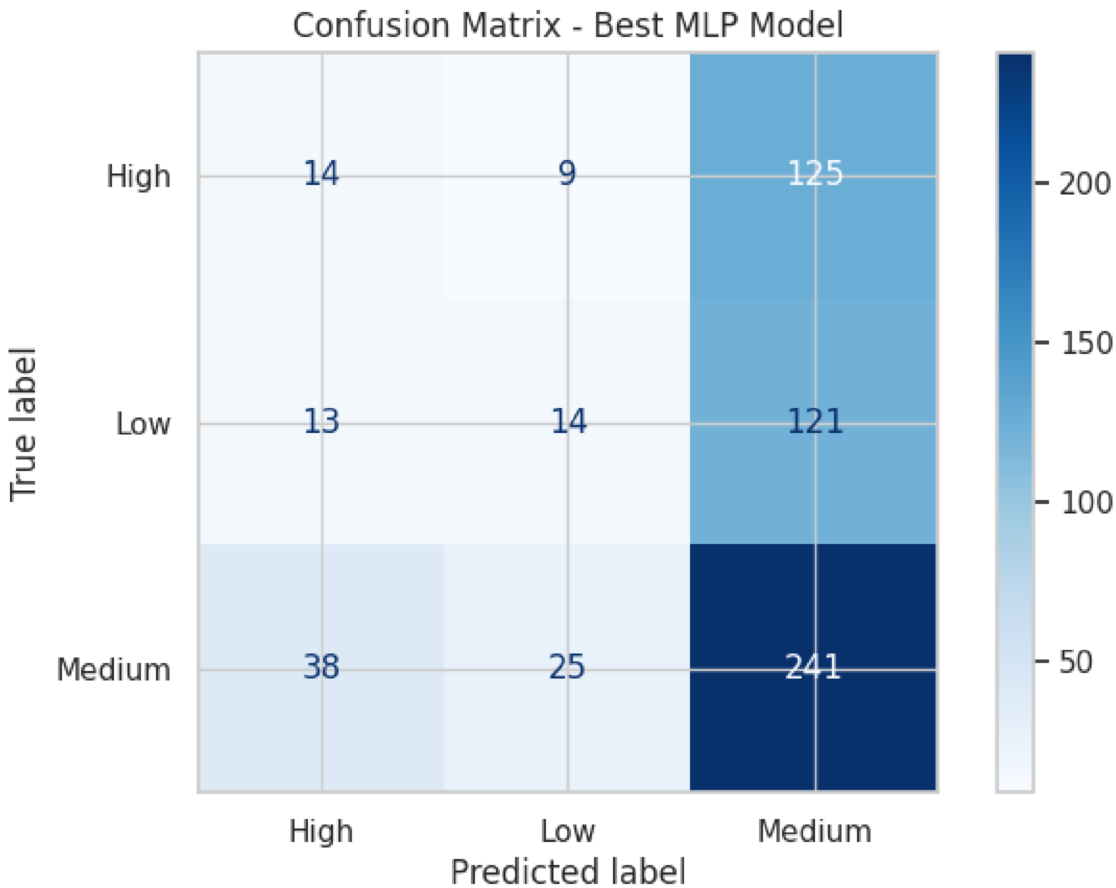
The confusion matrix for the baseline model (Figure 1) reveals that some Low-risk jobs are misclassified as Medium, and some High-risk jobs are classified as Medium. This reflects the fact that the three categories lie on a spectrum, and some boundaries are fuzzy.



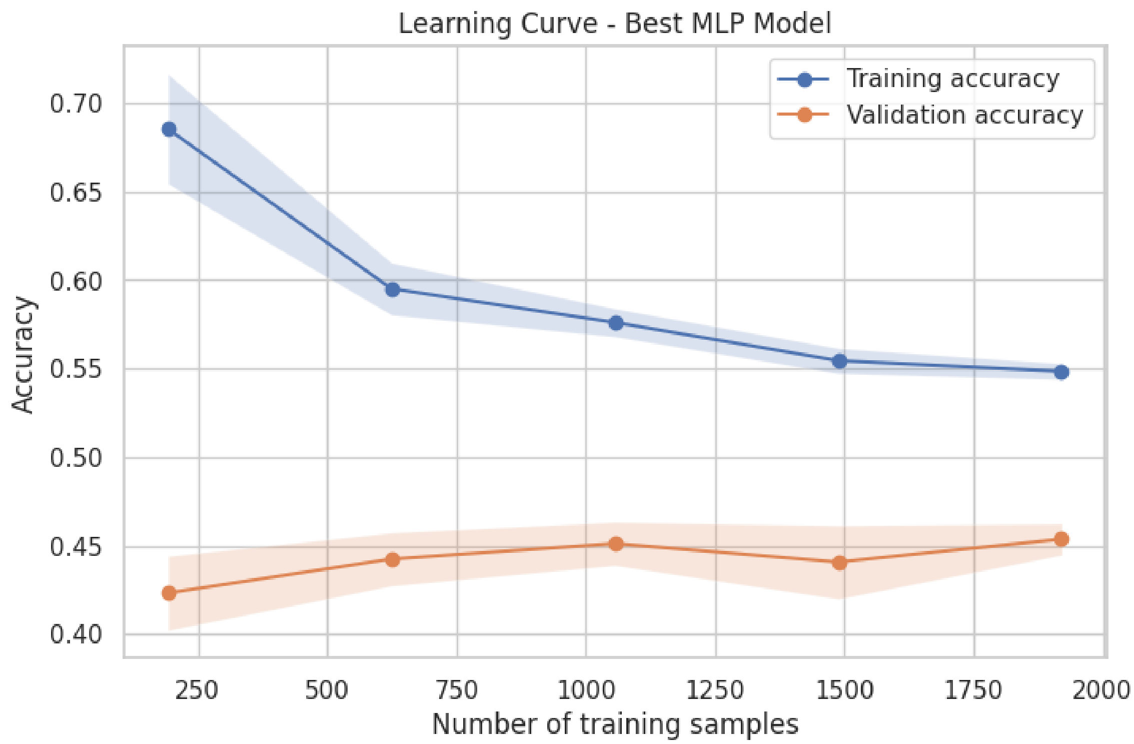
The hyperparameter study over depth and width produces a table and a heatmap (Figure 2). The heatmap shows mean cross-validated accuracy for each depth–width combination. In typical runs, one or two hidden layers with 64 or 128 neurons perform best, while very shallow and very narrow networks underfit and deeper networks with many large layers offer little additional gain and may overfit.



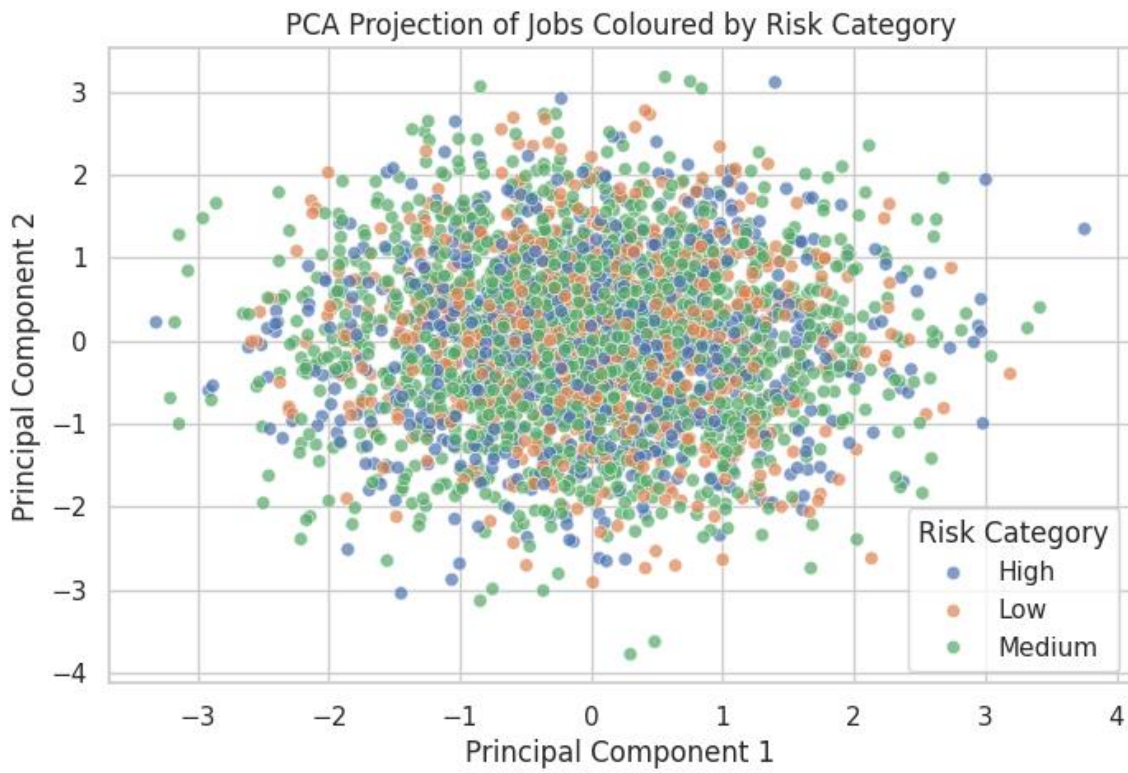
A line plot (Figure 3) illustrates how accuracy changes with depth for each fixed width. Adding a second layer often improves performance compared to a single layer, but a third layer yields smaller returns. This directly answers the question "How deep should my MLP be?" for this problem.



The final tuned model uses the best-performing architecture from the grid search. Its test accuracy improves over the baseline model, and the new confusion matrix (Figure 4) shows fewer misclassifications, especially between Low and High risk. The classification report reveals more balanced precision and recall across the three classes.



The learning curve (Figure 5) plots training and validation accuracy versus the number of training samples. The two curves converge as more data is used, indicating that the model is not severely overfitting and that its performance might improve further if more labelled jobs were available.



Finally, a PCA scatter plot (Figure 6) visualises the jobs in a two-dimensional feature space after preprocessing. Colours represent Risk_Category. Although some overlap remains, regions enriched in Low-, Medium- and High-risk jobs can be seen, which qualitatively supports the classifier's ability to separate the classes.

6. Limitations, Ethical AI and Impact on Employment

Although the MLP model achieves good performance on this dataset, several limitations and ethical questions must be considered.

Data and modelling limitations: the dataset is a simplified representation of complex labour market dynamics. The Automation_Probability_2030 is itself an estimate, based on assumptions about technology and economics [4]. The model cannot predict the future with certainty; it merely learns patterns present in the data. Important factors such as government policy, collective bargaining and new job creation are not represented.

Fairness and bias: if the dataset under-represents certain occupations, sectors or education levels, the model's predictions may be less accurate for those groups. If such predictions were used in career guidance or hiring decisions without care, they could reinforce existing inequalities. Responsible use requires checking performance across subgroups and avoiding over-reliance on automated scores.

Transparency and accountability: even though MLPs are less interpretable than simpler models, tools such as feature importance analysis or post-hoc explanations can help users understand why a job is labelled high risk. According to ethical AI guidelines, automated systems should be transparent, explainable and subject to human oversight [5].

Impact on modern life: on the positive side, models like this could help individuals identify vulnerable occupations and plan training for more resilient roles. Educational institutions could use similar analyses when designing curricula. On the negative side, there is a risk that employers or insurers misuse automation risk scores in ways that disadvantage workers. This highlights the importance of combining technical expertise with ethical reflection.

7. Learning Outcomes and Reflection

This project contributes to the course learning outcomes in several ways:

• **Neural networks as models of neural computation:** by designing and training an MLP, I gained practical insight into how layered neural architectures compute non-linear decision boundaries in a high-dimensional feature space.

• **Advanced machine learning for numerical data:** the project applies an advanced neural network model to a realistic tabular dataset, showing its effectiveness for predicting job-related outcomes compared with simpler approaches.

• **Programming non-trivial ML algorithms:** the Jupyter notebook contains a complete, runnable pipeline including preprocessing, cross-validation, hyperparameter tuning, and multiple visualisation tools.

• **Planning and implementing data analysis tasks:** I designed a structured workflow from EDA through baseline modelling to hyperparameter studies and final evaluation, similar to a professional data science workflow.

• **Critical evaluation of methods and literature:** the discussion connects the model's predictions with research on automation and employment [4] and with ethical AI guidelines [5], emphasising that technical performance is only one part of responsible AI.

8. Conclusion

In this project, I used a Multilayer Perceptron neural network to predict job automation risk categories from skills, salary, experience, education and AI exposure indicators. By exploring different depths and widths of the network, I showed how architectural choices affect classification performance and demonstrated practical techniques such as cross-validation, early stopping and learning curve analysis.

The code and figures in the accompanying notebook are intended as an educational tutorial for other students who want to apply neural networks to tabular data. At the same time, the report highlights important limitations and ethical issues surrounding AI predictions about people's livelihoods. Machine learning can provide useful insights into the future of work, but it should be used as a tool to support informed human decisions, not to replace them.

9. References

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533–536, 1986.

[2] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.

[3] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[4] C. B. Frey and M. A. Osborne, "The future of employment: How susceptible are jobs to computerisation?," Oxford Martin School Working Paper, 2013.

[5] High-Level Expert Group on Artificial Intelligence, "Ethics Guidelines for Trustworthy AI," European Commission, 2019.