Machine Learning

### SSID - 2193396

***Link to Git Repository:*** [***https://github.com/binhudas/SSID-2193396-Report.doc.git***](https://github.com/binhudas/SSID-2193396-Report.doc.git)

# Introduction, Background, Description of Problem:

Aims of project, Literary References

The healthcare data science landscape is rapidly evolving, and one of the most drastic changes is the integration of machine learning techniques. These machine learning techniques find patterns in data that offer greater understanding. Through this, they have become a transformative force, offering solutions to various challenges within the industry.

There is one such challenge in healthcare, which is especially pressing, revolving around optimising the management of patient admissions and discharges. Research into disruption to NHS services shows “39% of the 1,084 people had their care postponed or cancelled on two or more occasions.” (Healthwatch.co.uk, 2024). This kind of disruption has a serious negative impact on the patient and the organisation. Patients who go through cancellations report issues such as “ongoing pain, worsening mental health, and worsening symptoms,” and the health organisation struggles to maintain trust with its patients.

Delays in treatment can arise from several factors, but often they are due to a lack of resources. Frequently, patients who require a bed cannot acquire one since a previous patient's discharge has been postponed. Therefore, we can see that one way to ensure timely patient care is by efficiently scheduling and coordinating patient discharges. This process is crucial for proper bed utilisation, reducing strain on resources, and ultimately ensuring timely access to healthcare services. Accurately predicting the duration of stay for a patient is the business requirement of this project.

There are several difficulties with this process, which can be because of the complexity of the data. However, several organisations are offering solutions. Amazon Web Services states they can use “machine learning (AI/ML) and natural language processing (NLP) on de-identified, longitudinal patient data to discover patterns in a patient’s medical history” to help with “data-driven care management” (Amazon Web Services, Inc., 2024). This machine-learning project aims to create a similar solution through pattern recognition.

The machine learning model will be supervised as the dataset is labelled. We will use the labels to learn the relationship between the input features (patient information, services provided) and the target variable. The nature of the target variable will be a continuous variable, and the task of predicting a continuous outcome aligns with the principles of a regression model. Therefore, to summarise, the aim of this project is to use supervised machine learning to create a regression model that can predict the duration of a stay.

The workflow will match that of key machine learning frameworks such as the industry standard Cross-Industry Standard Process for Data Mining (CRISP-DM). The project began with a clear understanding of the business problem: accurately predicting the duration of stay of patients in healthcare facilities. This was identified with the requirements of possible stakeholders: improved resource allocation and enhanced patient outcomes.

# 

# Describe Dataset:

How it Was Obtained, Data Exploration, Ethical Considerations

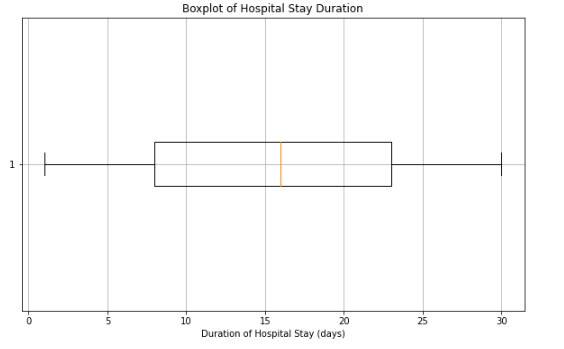
Another aspect of the evolution of the data landscape in the healthcare industry is the mass influx of ‘big data driven by rapid advances in medical technology’. This ‘big data’ involves large amounts of information such that it defies traditional storage solutions. There are lots of possible data sources available in the healthcare industry, including clinical trial data, data collected from wearable devices, and electronic health records (EHR), the last of which is the key focus of this project.

EHR data is obtained at a point of care, such as a hospital. These records often contain information that offers granular insights into a patient's background, as they can encompass a diverse range of information about a patient, from demographics and medical history to outcomes. This is why EHRs will be the primary data source for this project. They contain information that can offer insight into the different factors behind a patient’s stay in hospital. There is also administrative data available that offers similar information regarding the dates of patient care, but it lacks the necessary background medical information. Which is why I have decided to source data solely from EHRs.

Ethical concerns arise from working with any medical data; however, EHRs are especially sensitive due to their very personal nature. Due to this, they are subject to strict regulations such as the GDPR, so handling EHRs requires careful consideration. Access to an actual database of EHRs would be difficult, and the effort required would be unproductive. To avoid these issues, a synthetic dataset was employed. However, the same considerations for sensitive data were applied. These considerations took place in exploratory data analysis (EDA) through the process of anonymisation.

The EDA that took place gave some high-level insights into the data. For example: some were quickly identified for the anonymization process (name, doctor, hospital, and insurance provider). Several other variables show promise to be valuable to the machine learning model due to their possible relevance to the target variable: age, medical condition, medication, and test results. The next part of the EDA involves taking a deeper dive into the types of data. This allows us to plan the pre-processing steps required.

Then some more high-level insights into the data take place, which involve seeing the trends of different variables. For example, most patients were admitted due to asthma. The median stay is around 16 days, with most being contained under 23 days. During this EDA, we created the ‘Hospital Stay Duration’ variable by subtracting the date of discharge from the date of admission. This variable will be our target variable.



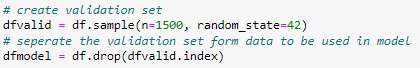
Now that the data has been explored, we can now anonymise. The specific method used is hashing. Hashing ensures the anonymisation that takes place is irreversible, ensuring complete protection and security of personal identifiable information (PII). The hashing method is a pharmaceutical industry-wide standard procedure due to the fact that it is secure but does not compromise the data’s validity or reliability. The hashed values are truncated to a length of 5 for efficiency.

To summarise this part of the workflow in this project, relevant data sources were identified that included the necessary information, admission details, and medical conditions. Exploratory data analysis (EDA) was used to better comprehend the dataset's structure, quality, and linkages. A greater understanding of the business problem was shown through the anonymization process, which upheld the confidentiality of patient data. Now the data has been anonymised pre-processing can take place.

# Workflow:

Pre-processing, Algorithms and Fine Tuning Process, Evaluation

Before any machine learning pre-processing takes place, the first step will be to implement a validation set. This is a portion of the overall data, and its purpose is to evaluate the performance of the machine learning model by being a dataset to create predictions. We will explore its purpose later; however, we create the validation set through the following code:

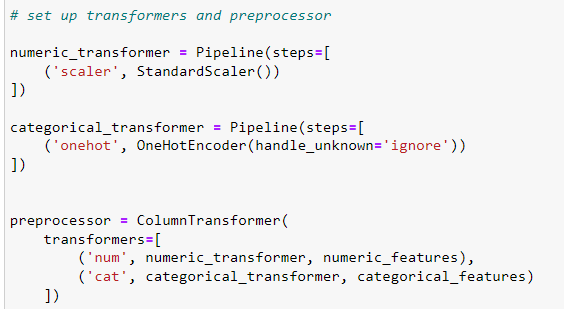


Now that the validation set has been created, pre-processing can begin on the remaining data. One of the first parts of pre-processing involves separating the features and the target variable of the dataset. The target variable is defined by y, while the features are defined by X. Some of the variables have been dropped as they have been determined to have little relevance. Having fewer variables makes the code more efficient and reduces runtime. The drop() function was used alongside the definition of the features variable.



The data can now be split into training and testing sets. The training set is 70% of the actual dataset, and the other 15% is used to test the model. As we don’t have access to real-world data, this step is essential; it allows us to ensure we can validate our model on unseen data. Despite the fact that the data is split randomly, we can use the random\_state parameter to ensure the reproducibility of the code. By setting it to 42, the data split will now remain consistent. We can also begin to categorise our features; this is simply done by using the data types previously discovered in the EDA.

Through our EDA, we also previously established that there are no missing values, so there is no need for any imputations. We can now use the pipeline function to create the numeric\_transformer and categorical\_transformers to prepare the numerical and categorical variables, respectively. The numerical variables, which in this case are just age, can be scaled to have a mean of 0 and a standard deviation of 1. This standardised the data to be on a similar scale, which improved the performance of the algorithm. For the categorical variables, OneHotEncoder was used. This converts them into binary vectors, where each category becomes a binary feature—a format that, again, can be used more effectively by machine learning algorithms. Both of these transformers are designed to increase the efficiency of the code.

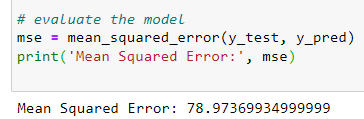


Data pre-processing techniques were implemented to clean and transform the raw data into a format appropriate for machine learning. With the data processed, the design of the model can begin.

As previously mentioned, the prediction model will implement regression to fulfil the aim of the project. It is based on a Random Forest Regressor model with 100 estimators. Random Forest is a common choice of supervised machine learning algorithms for regression tasks such as predicting continuous variables because it can handle complex relationships while avoiding overfitting. The specific random forest model used for this will have 100 decision trees. We again use the random state to ensure we are using the same random split previously used in our code.

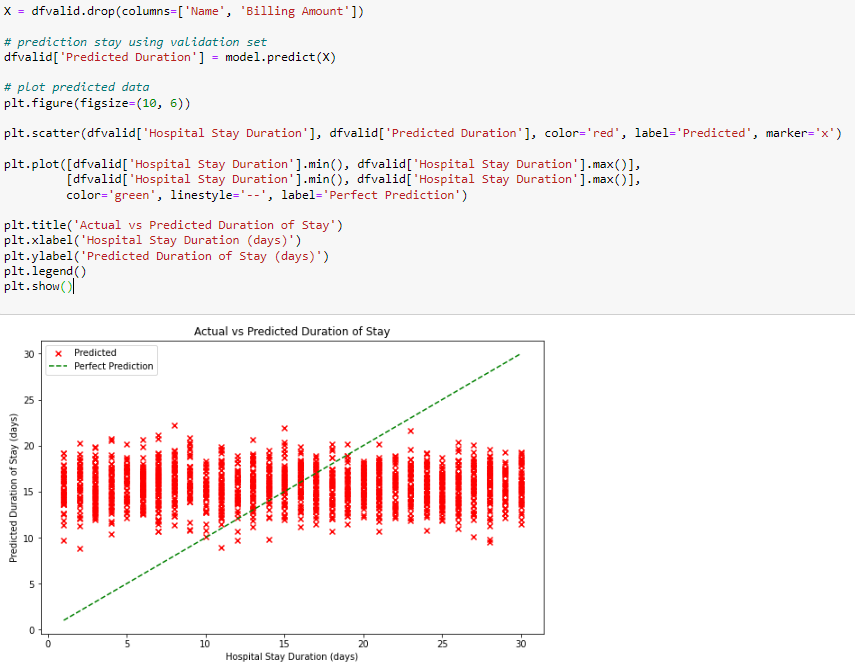
Now that the model has been set, we run the model on the pre-processed training data (X\_train and y\_train) to then create predictions (y\_pred).  
  
As this is taking place, we can also begin to create some form of measurement of the model’s prediction accuracy. For this, we calculate the mean squared error (MSE). The MSE is a metric that quantifies the difference between the predicted duration of stay and the actual duration of stay.

The MSE provides some quick insights into the model’s performance, and from our first run of the model, the MSE is calculated to be:



A good model will have an MSE close to 0 so we can see this model requires improvement.

Another way to analyse the model’s performance in a more visual way is through a scatter plot. To complete this, we will utilise the validation set from prior. We input the variables in the validation set into the defined model in a similar way to what was previously done. We then plot the predicted duration against the actual duration to create the below scatter plot.



This scatter plot provides a qualitative assessment of the model’s performance.

The model was trained on the pre-processed data, with features such as patient demographics, medical conditions, and admission details used to predict the target variable.

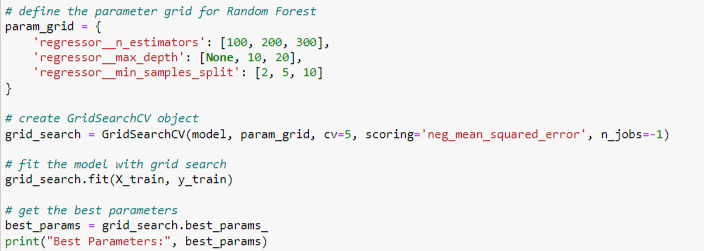
With the first run of the model completed along with some basic evaluation, we now need to see how we can adjust the code to improve the performance of the model.

We will utilise the GridSearchCV method from scikit-learn to perform hyperparameter tuning. The principle behind this is to search through a grid of specified parameters and perform cross-validation. The method begins by defining the grid of parameters (param\_grid), which contains different values for the parameters of the regression model we created.

These parameters relate to the decision trees in the random forest tree algorithm: the number of decision trees in the random forest (n\_estimators), the maximum depth of these decision trees (max\_depth), and the minimum number of samples needed to separate an internal node in a decision tree (min\_samples\_split).

Each of these parameters is crucial for controlling the complexity of the model, and fine-tuning them can optimise its performance on unseen data, such as the validation set. Increasing the parameters to have a greater number of decision trees with greater depth can improve the performance. While reducing the min\_samples\_split, it will produce more splits, resulting in more complex trees. However, these parameters can also have a negative impact on the model. A higher n\_estimator increases computational cost, and both max\_depth and min\_sample\_split can lead to overfitting.

These factors have been considered and are reflected in the code below.

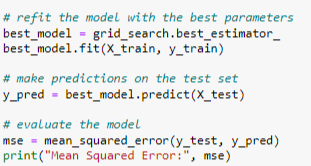


A GridSearchCV object has been created named grid\_search by passing in our model, the parameter grid we created, the number of partitions into which the dataset is split (5), and the scoring metric we use to evaluate the performance during hyperparameter tuning (negative mean squared error). We also potentially increase efficiency in the computation process by utilising available CPU cores with the n\_jobs=1 statement.

The training data is then passed through the grid\_search object so cross validation is performed on this training data to calculate the best combination of parameters.



With this information on the best parameters, we can code to refit our model according to them.



The code above refits the model pipeline to a new model called best\_model with the parameters found during the grid search. This best model is then tested with the test dataset, and its performance is evaluated again with the calculation of the MSE.



The MSE is smaller which suggests an increase in performance confirming an iterative approach in the workflow and that In the fine tuning section of the workflow; hyperparameter tuning techniques, such as GridSearchCV, were successfully employed to optimise the model's performance. However, it is important to properly interpret these readings to critically evaluate the performance and its real world impact. The final part of this project will focus on this and extensively analyse the model and our workflow.

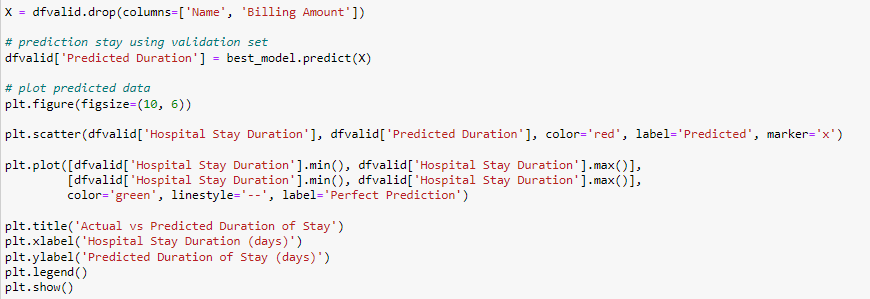
# 

# Evaluation and Interpretations:

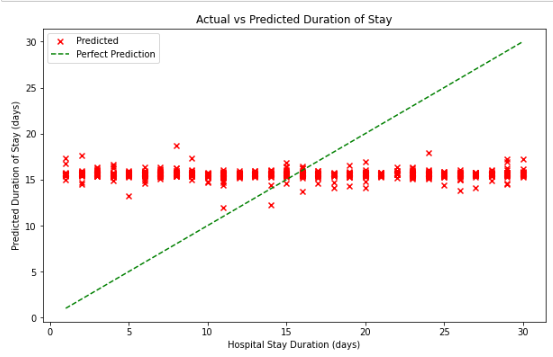
Critical Analysis, Scope for Improvements, Ethical Considerations

The MSE, although a powerful metric, is only a single numerical value and so does not provide detailed insights into the nature of the prediction error or overall performance of the model, lacking interpretability. It is also insensitive to distributional differences, as it treats all prediction errors equally. These factors make it difficult to understand the real-world impact.

Visualisations such as the scatter plots offer the opportunity for model diagnostics by highlighting areas of strength and weakness for the model. That is why another scatter plot has been created: to gain a better understanding of the impact of the hyperparameter tuning and to critically analyse the model in summary.



The code used is very similar to the code previously written as we are using the same validation dataset which in context is unseen data for both models. The scatter plot output produced is below:



We can see instantly that the predictions of the model have been impacted by the hyperparameter tuning. The predictions are less varied and are more streamlined. However, we can see that towards the ends of the data, away from the median, the predictions stray further away from the perfect prediction line. This is an important discovery and gives us valuable insight into the model.

This is evidence that within the model that there is bias towards the central tendency of the target variable, as it is optimised to predict values around the median of the target variable but struggles to accurately predict in other regions of the data. What this suggests is that the model may lack generalisation capabilities and will not perform as well with data that has a varied distribution.

This bias can result from a variety of reasons, but specifically in this project, it is likely it arose from hyperparameter tuning, as the large number of parameters led to a prioritisation of fitting the data while sacrificing accuracy in fewer represented regions. In terms of addressing this issue, an approach would likely take the form of simplifying the model using techniques such as increasing the min\_samples\_split. Thus the evaluation process of the workflow took several forms. Through appropriate evaluation metrics, such as the MSE and the interpretation of scatter plots.

In the context of real-world implementations, this model would be useful for use with a general population that is likely to follow the majority or be close to the median. However, several significant improvements would need to be implemented. This project lacked stakeholder feedback, however, in a real-world scenario. This would be an integral part of the evaluation and would help with more iteration through feedback loops, leading to a better model that is aligned to the business problem. Experimenting with different machine learning algorithms may provide a better and more effective model; even some tuning with the current parameters could prove effective. Greater CPU capabilities would be required, and unfortunately, that is out of the scope of this project.

If greater resources were available, additional features could be included such as socioeconomic status and patient lifestyle factors to provide a more holistic view. Greater resources would also allow better evaluation through techniques such as feature importance graphs. Were the model to be implemented, model monitoring, maintenance and updating would need to be addressed to ensure continued effectiveness. Moreover, bias mitigation strategies would have be implemented to prevent discriminatory outcomes, particularly concerning sensitive attributes such as race, gender, and socioeconomic status.

By adhering to the CRISP-DM workflow, this project effectively guided the development and deployment of a machine learning solution for predicting discharge dates in healthcare facilities, ultimately addressing the identified business problem and setting the foundations for a suitable solution.

# 

# References:

*Amazon Web Services, Inc. (2024). Patient Outcome Prediction | AWS Solutions for Healthcare, Life Sciences, and Genomics | AWS Solutions Library. [online] Available at: https://aws.amazon.com/solutions/health/patient-outcome-prediction/ [Accessed 29 Mar. 2024].*

*Healthwatch.co.uk. (2024). Delays to NHS care: Understanding the impact. [online] Available at: https://www.healthwatch.co.uk/report/2023-07-27/delays-nhs-care-understanding-impact [Accessed 29 Mar. 2024].*

*Senil, K. (n.d.). Pharmaceutical Data Protection in Supply Chain Using Hashing. [online] Available at: https://www.ijcse.net/docs/IJCSE19-08-02-027.pdf [Accessed 31 Mar. 2024].*

*The King’s Fund. (2023). The Hidden Problems Behind Delayed Discharges | The King’s Fund. [online] Available at: https://www.kingsfund.org.uk/insight-and-analysis/blogs/hidden-problems-delayed-discharges [Accessed 29 Mar. 2024].*