UWA Data Analytics Bootcamp - Project 4

Predicting AIDS Progression With Data Insights

Shiva Bajelan, Linh Doan, Rahul Gade, Jeremy Hooper

Introduction

HIV/AIDS remains a significant global health challenge, and early detection of progression from HIV to AIDS is crucial for effective treatment and management. In this project, we aim to develop a machine learning model capable of accurately predicting HIV progression to AIDS.

The AIDS Clinical Group of the National Institute of Allergy and Infectious Diseases – General Research Center funded a study (175) in the early 1990s called "*A trial comparing nucleoside monotherapy with combination therapy in HIV-infected adults with CD4 cell counts from 200 to 500 per cubic millimetre".* The study was *a controlled trial of two nucleoside analogues plus indinavir in persons with human immunodeficiency virus infection and CD4 cell counts of 200 per cubic millimetre or less.* The purpose of the study was to examine the performance of two different types of AIDS treatments. The study produced A public dataset called AIDS Classification containing 2139 rows and 23 columns. We used this dataset in our project.

The first thing we did was read about the study, its purpose, and its findings, which made it easy to understand the dataset's features.

The Study

- **Participants -** There are 2139 participants. All participants who come into the survey are HIV positive but do not have AIDS. They all have CD4 cell counts of between 200 and 500 per cubic millimetre. Many of them have a history of some treatment for HIV. None of them have AIDS.
- **Related History –** The related history of each participant is included in the following columns:

Features/Columns

	Censoring Indicator:	Indicate whether the event of interest (for example, infection with AIDS) was observed (0) or not(1).
treatment	Treatment indicator 0=ZDV only, 1= ZCV + ddl,	ZDV – Zidovudine, drug for the treatment of HIV.

	2=ZDV + Zal, 3=ddl only	Zal – Zalcitabine – antiretroviral medication used to treat HIV. dll – Didanosine – NRTI and is used to help control HIV infection
ZDV_only_treatment	treatment indicator (0=ZDV only, 1=others)	
off_treatment	indicator of off-treatment before 96 +/-5 weeks (0=no,1=yes)	
time	time to failure or censoring	Time to Failure: cd4 cell < 50% or infected.
age	age years at baseline	Age of participants at baseline
weight	weight (kg) at baseline	Weight of the participants at baseline
hemophilia	hemophilia (0=no, 1=yes)	Participants suffered from haemophilia before or during the study?
Homosexual_actvity	homosexual activity (0=no, 1=yes)	I assume it means that the participant has undertaken any homosexual activity before or during the study.
drugs	history of IV drug use (0=no, 1=yes)	I assume it means that the participant has taken any "recreational drugs before or during the study.
race	race (0=White, 1=non-white)	
gender	0=F, 1=M	
karnofsky_score	Karnofsky score (on a scale of 0 (dead) -100(OK)	The Karnofsky score in AIDS patients helps monitor how the disease or associated complications affect their daily living capabilities.
therapy_prior	Non-ZDV antiretroviral therapy pre-study (0=no, 1=yes)	
ZDV_last_30days	z30: ZDV in the 30 days before 175 (0=no, 1=yes)	

prior_ART_length	Days pre-study anti-retroviral therapy	The number of days before the 175 study anti-retroviral therapy.
ART_history	antiretroviral history (0=naive, 1=experienced)	Naive = never previously treated with ART drugs
ART_history_stratified	antiretroviral history stratification (1='Antiretroviral Naive', 2='> 1 but <= 52 weeks of prior antiretroviral therapy, 3='> 52 weeks)	
symptom	symptomatic indicator (0=asymp, 1=symp)	
	indicator of off-trt before 96 +/-5 weeks (0=no,1=yes)	
cd4_base	CD4 at baseline in cells per cubic millimetre. Higher cell count, higher immunity.	Disease Monitoring: The CD4 count is a key indicator of immune function in individuals with HIV.
cd4_20wks	CD4 at 20 +/-5 weeks	
cd8_base	CD8 at baseline in cells per cubic millimetre. Complex action. Will remove.	
cd8_20wks	CD8 at 20 +/-5 weeks	

Target Column

infected	Is infected with AIDS 0=No, 1=Yes	‘infected ‘ became the ‘target’ in our ML models

Here, we detail the data cleaning and transformation, data exploration to determine the relationship between the various features and 'target' of the data set, implementation and optimization of five predictive data models, and discussion of the results.

Data Cleaning and Transformation

We cleaned, normalised and standardised the dataset before exploring, visualising, and modelling. Cleaning data from an acquired CSV file is a crucial step in data analysis to ensure accuracy and reliability. The process involved several steps:

- Firstly, we reviewed the AIDS_classified.csv dataset in a Jupyter Notebook using the .nunique(), .info(), and .describe() functions to identify duplicate rows, missing values, or formatting errors. We noted that no missing values were present in the dataset.
- We created a series of Box Plots to see if there were any obvious outliers in the continuous data sets.
- We decided to rename many features to clarify their roles in the study.
- Before this step, we noticed no references to the influence of the data stored in the cd8_base and cd8_20wks features. We decided to drop these two columns. We reviewed this decision later and found that dropping the columns didn't significantly change the results.
- Later, we added two columns to the dataset: cd4_propdif (proportional difference between cd4_20wks and cd4_base) and cd4_numerical_change (numerical difference between cd4_20wks and cd4_base). We found that cd4_base had some zero values, and this calculation introduced some infinity values. We removed these values.

Our cleaned and normalised data set comprised 21 features and 1 target column (infected).

Data Exploration and Visualisation

Data were explored to gain insights into the distribution of features and relationships between variables:

- **Descriptive Statistics**: The describe() method provided an overview of the dataset, presenting key summary statistics including count, mean, standard deviation, and range for numerical features.
- **Unique Values**: Iterating through each column revealed unique values, offering insights into the dataset's cardinality and range of categorical variables.
- **Bar Charts for Categorical Variables:** Bar charts, created using sns.countplot(), visually depicted the distribution of categorical variables such as treatment, risk factors (e.g., homosexual activity, drugs), and demographic variables (e.g., race, gender).
- **Histograms for Numerical Variables**: Histograms, generated with sns.histplot(), visually represented the distribution of numerical features like age, weight, and CD4 counts, aiding in understanding the data distribution and identifying potential outliers.

- **Pie Chart for Infected vs Not Infected:** A pie chart visually represents the distribution between infected and non-infected samples, offering an overview of the target variable's distribution within the dataset.
- **Box Plots for outliers:** Box Plots of continuous numerical data alerted us to potential outliers.

Conclusion

The data exploration and visualisation techniques provided valuable insights into the dataset's characteristics, facilitating a better understanding of the variables and their distributions. These visualisations serve as a foundation for further analysis and decision-making in addressing HIV infection patterns and treatment strategies.

Tableau Analysis and Visualisations

Key Insights:

- **Gender Disparity**: Male patients exhibit a higher infection rate compared to females.
- **Sexual Activity and Infection**: No significant correlation between sexual activity and infection rate; non-infection rates surpass infection rates.
- **Age and Recovery**: Younger patients demonstrate a higher recovery rate than older patients.
- **Ethnicity and Infection**: Both white and non-white patients show a higher infection rate.
- **Effective Treatments**: Patients living longer without infection receive a combination of ZDV, ddl, Zal, and ddl; ZDV alone proves ineffective.
- **Antiretroviral Drug Exposure**: Patients previously treated with antiretroviral drugs exhibit higher infection rates and reduced response to standard treatment.
- **Treatment Efficacy**: Combination treatments improve patients' immune systems, as evidenced by increased CD4 count.
- **Impact of ART**: Patients never treated with ART show a lower infection rate than those exposed to antiretroviral drugs.

Conclusion

Tableau visualisations provide crucial insights into various factors influencing HIV infection rates, treatment effectiveness, and patient demographics. These insights inform healthcare strategies and highlight the importance of tailored treatments based on patient characteristics and medical history.

Data Exploration

We noted that the purpose of this original study was to look at the effectiveness of the various treatments in slowing down the transition from an HIV condition to an AIDS infection. Two of the event measures were a reduction in the CD4 cell concentration by at least 50% and becoming infected with AIDS. We could learn more about how the process worked by examining the relationships between the participant's time in the study, the treatment, changes in CD4 cell concentration, and infection status.

The visualisations referred to here are included in our presentation:

The Histogram of proportional CD4 Count Change by Treatment showed that Treatment 1 was the most effective in boosting CD4 levels, thereby improving immunity. We also see that CD4 cell count has decreased by more than 50% in several cases.

The histogram of the Number of Participants Meeting Fail Conditions by Treatment shows that Treatment 0 was the least effective. The output of this calculation included a list of participant numbers, including the numbers who had failed the two events of reduction in CD4 cell count and becoming infected.

- Treatment 0:
- Total Participants: 528
- Participants with CD4_propdif <= -0.5: 27
- Participants Infected: 181

- Treatment 1:
- Total Participants: 516
- Participants with CD4_propdif <= -0.5: 10
- Participants Infected: 103

- Treatment 2:
- Total Participants: 522
- Participants with CD4_propdif <= -0.5: 5
- Participants Infected: 109

- Treatment 3:
- Total Participants: 554
- Participants with CD4_propdif <= -0.5: 12
- Participants Infected: 127

The bar chart of Total Participants vs. Participants Meeting Failing Conditions by Treatment shows the relationship between the total participants in each treatment and those who meet failing conditions. Again, this shows that Treatment 0 is the least effective.

Machine Learning Models

1. **Logistic Regression:**
   - Methodology: Estimates the probability of a binary outcome using a logistic function.
   - Advantages: Efficient for linearly separable data, provides probabilities for outcomes.
   - Disadvantages: Assumes linear relationship between features and target variable, unsuitable for complex relationships.
2. **K-Nearest Neighbors (KNN):**
   - Methodology: Classifies data points based on the majority class of their k nearest neighbours.
   - Advantages: Simple to understand and implement, no training phase.
   - Disadvantages: Computationally expensive for large datasets, sensitive to irrelevant features and the choice of k.
3. **Support Vector Machine (SVM):**
   - Methodology: Finds the hyperplane that best separates classes in high-dimensional space.
   - Advantages: Effective in high-dimensional spaces, memory efficient.
   - Disadvantages: Limited to binary classification, sensitive to the choice of kernel function and parameters.

Methodology:
We utilised three traditional classification algorithms: Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM).
The dataset was split into training and testing sets using different train-test split ratios: 70:30, 50:50, and 30:70.
Evaluation metrics included accuracy, precision, recall, F1-score, and support.
Logistic Regression:
Achieved an accuracy of approximately 85% across all train-test split ratios.
Displayed balanced precision and recall for the "not infected" class, indicating robust performance in correctly identifying individuals not infected with the AIDS virus.
However, precision and recall for the "infected" class were relatively lower, suggesting challenges in correctly classifying individuals infected with the virus.

K-Nearest Neighbors (KNN):
Produced an accuracy ranging from 81% to 82%.
Demonstrated relatively lower precision and recall for both classes compared to Logistic Regression, indicating a less optimal performance in classification.

Support Vector Machine (SVM):
Attained an accuracy like Logistic Regression, ranging from 84% to 85%.
Showed balanced precision and recall for both classes, indicating effective classification performance.

4. **Random Forest:**

- Methodology: Ensemble learning method that builds multiple decision trees and combines their predictions.
- Advantages: Robust to overfitting, handles missing values and maintains accuracy with large datasets.
- Disadvantages: Less interpretable than single decision trees, can be computationally expensive during training.
- **Performance Metrics:**
  - Accuracy: 0.8925
  - Precision: 0.8481
  - Recall: 0.6633
  - F-1-score: 0.7444
- **Observations and Recommendations:**

Accuracy (0.89): The model demonstrates high overall correctness in its predictions, suggesting that it effectively classifies most instances correctly.

Precision (0.84): The precision indicates that when the model predicts a positive outcome, it is correct approximately 84.81% of the time. This suggests that the model has a relatively low rate of false positives.

Recall (0.66): The recall suggests that the model correctly identifies about 66.33% of all positive instances. While this is lower than precision and accuracy, it still signifies a reasonable ability to capture positive instances.

F1-score (0.74): The F1-score balances precision and recall. It suggests that the model balances between minimising false positives and false negatives.

Overall, these results suggest that the Random Forest model performs well, with high accuracy and precision, although there is room for improvement in recall. This could indicate that the model may need further tuning or optimisation to capture positive instances better.

5. **Deep Neural Network (DNN):**
   - Methodology: Utilizes multiple hidden layers to learn complex patterns in data.
   - Advantages: High learning capacity allows for automatic learning of feature representations.
   - Disadvantages: It requires large amounts of data and computational resources and is prone to overfitting.

We made four attempts to optimise this model:

Model_1:

- **Model Architecture:**
  - Activation Function: Relu
  - Hidden Layer 1: 80 neurons
  - Hidden Layer 2: 30 neurons
  - Output Layer: 1 unit with Sigmoid activation function
- **Performance Metrics:**
  - Training Accuracy: 0.99
  - Testing Accuracy: 0.87
  - Precision: 0.78
  - Recall: 0.73
- **Observations and Recommendations:**
  - There is a notable disparity of approximately 12% between the accuracy scores of the training and testing datasets. This discrepancy suggests that the model is overfitting to the training data.

Model_2

- **Model Architecture:**
  - Activation Function: ReLU
  - Hidden Layer 1: 14 neurons
  - Hidden Layer 2: 7 neurons
  - Output Layer: 1 unit with Sigmoid activation function
- **Performance Metrics:**
  - Training Accuracy: 0.94
  - Testing Accuracy: 0.89
  - Precision: 0.82
  - Recall: 0.76
- **Observations:**
  - Compared to the initial model (Model_1), Model_2 demonstrates improvements in both testing accuracy and recall.
  - The testing accuracy has increased from 0.87 to 0.89, indicating enhanced performance on unseen data.
  - Additionally, the recall score has improved from 0.73 to 0.76, indicating better sensitivity in identifying positive cases.

Model_3

- **Model Architecture:**
  - Optimized Structure using Keras Tuner
- **Performance Metrics:**
  - Training Accuracy: 0.94

- Testing Accuracy: 0.88
- Precision: 0.80
- Recall: 0.74

Model_4

- **Model Architecture:**
  - Optimized Structure using Keras Tuner
  - This attempt kept the 7 most important features as visualised in the Random Forest Importances function and dropped the remainder.
- **Performance Metrics:**
  - Training Accuracy: 0.9
  - Testing Accuracy: 0.9
  - Precision: 0.86
  - Recall: 0.76

- **Observations:**
- Model_3 and Model_4 were optimised using the Keras Tuner, resulting in the following performance metrics:
  - Training Accuracy: 0.9
  - Testing Accuracy: 0.9
  - Precision: 0.86
  - Recall: 0.76
- **Comparison with Other Models:**
  - Model_2 has a higher testing accuracy (0.89) than Model_3 (0.88).
  - Model_2 also exhibited a slightly higher recall (0.76) than Model_3 (0.74).
- **Conclusion:**
  - Despite optimising the neural network structure using the Keras Tuner in Model_3, the performance did not surpass that of Model_2.

These statistical metrics provide insights into the performance of a Deep Neural Network (DNN) model:

Training Accuracy (0.9): Training Accuracy measures model accuracy on training data, indicating the proportion of correctly classified instances. A score of 94% suggests accurate predictions.

Testing Accuracy (0.9): Testing Accuracy assesses model accuracy on unseen data, showing the proportion of correctly classified instances. A value of 0.88 suggests accurate predictions.

Precision (0.86): Precision assesses positive prediction accuracy by comparing the ratio of true positives to the total positives predicted. 0.80 means 80% were true positives.

Recall (0.76): Recall, or sensitivity, gauges the model's ability to identify positive instances. It is the ratio of true positives to actual positives. 0.74 implies 74% accuracy.

Predictive Machine Learning Models Discussion

Handling Class Imbalance

The code used in our predictive learning models indicates that our classes are imbalanced. Handling class imbalance is crucial in machine learning when one class has significantly more samples than others. To handle this issue, we have used the following strategies:

- **Using algorithmic techniques**, adjusting the model's class weights to penalise misclassifications of the minority class more than the majority class. This can be achieved by setting the class weight parameter to 'balanced' in some models, such as Logistic Regression and Support Vector Machine2.
- **Using ensemble methods,** such as Random Forest, which inherently handles class imbalance by combining multiple weak learners.
- **Evaluation Metrics,** using appropriate evaluation metrics less sensitive to class imbalance, such as precision and recall rather than accuracy.

Conclusion

Overall, we conclude that:

1. **Neural Network Performance:** Neural Networks achieve high accuracy (89%) and precision (85%) and slightly lower recall (71%) than Random Forests and Support Vector Machines (SVM), indicating their potential for accurate predictions but with some trade-offs. The Neural Network model, in its fourth attempt, shows promising performance with a high accuracy_train (%90), high accuracy_test (%89) and precision(%88), and recall for the infected class(class 1) %78.
2. **Optimal Prioritization:** If maximising accuracy and precision is paramount, Neural Network is the top contender due to its impressive performance in these metrics.
3. **Recall Priority:** When prioritising recall, the Support Vector Machine (SVM) with 77% recall stands out, making it the preferred choice for scenarios where maximising recall is crucial.
4. **Random Forest Analysis:** Random Forest yields high accuracy (89%) and precision (85%) but exhibits a lower recall (66%), suggesting its potential for precise predictions but with limitations in capturing all relevant instances.
5. **Alternative Options:** Logistic Regression offers 85% accuracy and 76% recall, while KNN provides 80% accuracy and 76% recall. Both provide viable alternatives with balanced performance metrics for analysis consideration.
6. **Given the concept of our medical dataset** and the importance of predicting individuals infected with AIDS to get timely treatment, accuracy and recall for the infected

class(class 1) are considered the most important metrics for choosing the best model in this project. So, the Neural Network is the best model, with an accuracy of 89% and a recall of 78%.