Curriculum Vitae of Mr Luyanda Percival Dhlamini

☎ 0788523231 luyandadhlamini@gmail.com

Work Experience & Projects

Position	Organisation	Duration
Founder	**Learners License Goat**	**April 2020 – Present**
Founded a startup that sells learners license test preparation study guide. - Study guides sold direct to consumers over social media (Facebook). - Currently employs 2 people. - The study guide product is positioned as an alternative to K53, a popular learners license test preparation resource.		
Data Scientist	**Mr Price Group Limited**	**Jan 2019 – March 2020**
Projects - **Improved the product to store allocator system by 4%.** - **Built & deployed deep learning model to forecast daily sales at an individual store level.** - **Built time series model to forecast division daily sales using AWS Forecast.** - **Used rule based anomaly detection to flag orders & stock that needs attention.** - **Performed ad hoc analysis & reporting in Excel (E.g. product promotion analysis).** - **Created Tableau dashboards for planning teams.**		
Intern Data Scientist	**BCX**	**Jan 2018 – December 2018**
Projects - **Analysed the City of Cape Town's data & explain the major factors driving the water crisis.** - **Used NLP & Classification to predict student's personalities using 500-word essays as a data source.** - **Used Bayesian inference to predict the outcome of soccer matches (using the 2018 FIFA world cup as a case study).** - **Used Unsupervised Learning to learn about factors impacting Cape Town's traffic, with Tweets as a data source.**		

Key Competencies	Knowledge Areas:
Artificial Intelligence / Deep Learning	Prototyping, testing & deploying models in the Pytorch framework on AWS Sagemaker. Building Deep Neural Networks & Hyperparameter tuning. Theoretical Familiarity with: General Adversarial, Convolutional & Recurrent Networks.
Machine Learning	**Supervised Learning:** Linear Regression, Logistic Regression, Decision Trees, Random Forests, NLP **Unsupervised Learning:** K-Means & Hierarchal Clustering, Principal Component Analysis **Model Selection & Evaluation** Cross validation & Bootstrapping. MSE & F1-test evaluation, precision, recall interpretation
Python	Writing advanced Python programs & algorithms with clear documentation & reproducibility. **Packages:** Pandas, Numpy, Matplotlib, SciKit Learn, SQLAlchemy, Jupyter Notebooks, Selenium.
SQL	Writing advanced queries, stored procedures & events in SQL. **Platforms:** Microsoft SQL Server, MySQL, PostgreSQL & Hive (for Hadoop)
Amazon Web Services	Sagemaker, Forecast, IAM roles & policies, S3, EC2, Quicksight, RDS, DynamoDB, Lambda Functions & API Gateway
Data Visualisation & Communication	Dashboard building on Microsoft Power BI, Tableau & AWS Quicksight Using Microsoft Office for data analysis & presentation (Excel, Powerpoint, Word, OneDrive)

Education

Curriculum Vitae of Mr Luyanda Percival Dhlamini

☎ 0788523231 luyandadhlamini@gmail.com

Accreditation	Institution	Duration
AWS Data Scientist Specialist	AWS Training	July 2020 - Ongoing
Relevant courses include: - Developing Machine Learning Applications, Machine Learning Security, Math for Machine Learning.		
Data Science Certification	ExploreDataScienceAcademy	2018 – 2018
- Creating programs containing advanced algorithms using a procedural programming language (SAQA ID 115382) - Presenting information in a public setting (SAQA ID 115382)		
Bachelor of Commerce (Management Studies)	University of Cape Town	2013 – 2016
Relevant courses include: - Python & Java programming. - Applied Statistics & Mathematics. - Economics & Marketing.		
National Senior Certificate	Umlazi Comtech High School	2008 - 2012
- Attained seven distinctions in all my subjects with an average of 90%. - Relevant courses include: Mathematics (98%), Physical Sciences (97%) & English (89%).		

Relevant Certificates

Certificate	Provider	Certificate	Provider
AWS Machine Learning Security	AWS Training	Bayesian Machine Learning	Udemy
AWS Security Fundamentals		Cluster Analysis & Machine Learning	
Demystifying Machine Learning, Artificial Intelligence & Deep Learning		Microsoft Power BI	
Probability and Statistics	Udemy	Natural Language Processing	
Data Science & Machine Learning		SQL Database Design	
		PostgreSQL	

Link to Qualifications & Cerificates

LinkedIn Profile: https://www.linkedin.com/in/luyandadhlamini/

References are available on request.

Recent Projects Worked On

Built & deployed deep learning model to forecast daily sales at an individual store level.

Duration: 3 months (Dec 2019 to March 2020)	Type: Individual Project

An earlier project I had worked on had tried to predict daily sales across all stores. AWS Forecast was used as the machine learning platform but was found to not outperform predictions made by a group of internal senior executives.

While the previous project involved making a prediction for the entire division, the planning unit wanted to know if an automated model could be used to make highly accurate monthly sales predictions at an individual store level. The apparel division of Mr Price has over 500 stores, making it impossible to manually estimate daily sales values for each store. These predictions would then be used to feed into various systems that are used throughout the business.

Mr Price already has a large amount of historic daily sales & stock level data available at a store level. Each store is also different in its properties & location. The availability of large historical data meant that a Deep Learning model could be applied to this project.

Data Used

☎ 0788523231 luyandadhlamini@gmail.com

- Various data sources spanning a period of 5 years (Sales, Stock, Date & Store features).
- The data extracted from Hadoop cluster using SQL.
- Temporary tables were created to hold all necessary data.

- The extracted data was transformed into a form that the pytorch framework would understand. Part of this transformation included encoding variables & dealing with missing values.
- The input data was also converted from Pandas tables to NumPy ndarrays & finally a multidimensional tensor object.
- Pytorch's DataLoader & Dataset classes were used to split the input data into batches that could be parallelised during training.

Final Solution Architecture

- A Deep Neural Network (DNN) was built in Pytorch to estimate daily sales (a regression problem). The model was created to be able to train on either CPU or GPUs (using Pytorch's device feature). An advantage to using a DNN for this regression problem was that the network would be able to model more complex behaviour among the input parameters.
- A DNN model was constructed, with the following characteristics:

1. An input layer with 25 parameters (store ids, dates + 23 other variables).
2. 14 hidden layers, using ReLu activators.
3. A linear output layer with one output parameter (a daily sales estimate in Rands).

Solution Implementation

- Once a data processing pipeline & the model had been built on my local pc using a small sample of the data, the full dataset was uploaded to AWS S3 & made available to a Sagemaker Jupyter Notebook.
- This allowed me to train the model on an AWS Sagemaker instance with GPUs (ml.p3.2xlarge). Which significantly sped up training times. The dataset was shuffled and divided into 3 parts (training (60%), validation (20%) & test/hold out set (20%)).

Solution Evaluation

- The model was optimised using a Mean Squared Error (MSE) loss function & Adam optimiser.
- The aim was to achieve the lowest MSE possible.
- The model's predictions were also aggregated to a week level & compared to sales predictions produced by a existing business process. The DNN's predictions had a lower MSE & Mean Absolute Percentage Error when compared to predictions from an existing process.

Iterate & improve performance

- The final steps listed above were arrived at after various iterations.
- E.g. The input features to use, the way the input data was structured, the number of hidden layers used in the model.
- The model was continuously tweaked until additional changes did not lead to a lower MSE value.

Project Outcomes

- Once the head of Data Science & Planning Director were satisfied with the performance of the model, a dedicated host EC2 instance was used to make the model available for inference and to allow the IT department to integrate it into existing systems.
- Security & maintenance of the model's dedicated host is managed by the IT department.

Curriculum Vitae of Mr Luyanda Percival Dhlamini

☎ 0788523231 luyandadhlamini@gmail.com

Improved the product to store allocator system by 4%.	
Duration: 7 months (April 2019 to Nov 2019)	**Type:** Individual Project

Mr Price has a system that decides how much of each item category (e.g. Jeans, shirts) go to each individual store. This system derives its main inputs from what Merchandise & Location Planners have as sales targets for a week.

In production, this system ultimately tells our DC to pack this number of jeans/ shirts to each individual store. There are millions of possible product combinations across 500 stores, thousands of product categories & multiple weeks.

- Part of this system (product to store allocator) used one of two algorithms to allocate stock.

1. It either looked at how that category of product sold last year & used that same recipe to allocate stock this year, or
2. It looked at what the merchandiser's product plan was this year & used that to allocate (if 50% of purchases are jeans, all stores get 50% jeans as a product mix).

Each of these approaches had its disadvantages, especially when new categories are introduced, new stores are opened, or a store is an international store & has a delayed sales period.

Problem Statement:

- Is there a better formula for performing product to store allocations?

Data Used

- Historical inputs into the current product to store allocations system.
- Allocations made by the system based on current implementation (option 2 above)
- Subsequent sales based on allocations made, an internal formula was used to decide which sales could be attributed to allocations made by the system.

Final Solution Architecture

- An iterative method that considers both sales history & current product allocation budgets to determine how to allocate products across all stores.
- The method used indexes to express each store's preference for a particular product relative to all other stores.
- This solution had the advantage of combining both option 1 & 2 above when making allocations. It considered each store's history, as well as the product mix planned by the merchandiser for the current year.

Solution Implementation

- The solution was implemented as a series of SQL scripts.
- The solution was used to create predictions for allocations that would have been made in the past.
- Back testing was used to compare the solutions predictions against actual historical sales.
- Data was exported to Excel for performance evaluation against existing methods.

Solution Evaluation

Curriculum Vitae of Mr Luyanda Percival Dhlamini

☎ 0788523231 luyandadhlamini@gmail.com

- As a baseline, the existing method used by the business had an absolute error rate of 24%. This means that it would either over or under allocate stock to a store by 24% above or below what the store would sell.
- This presented significant margin for improvement, since it meant that the business was either wasting product that could have been better used elsewhere or insufficiently allocating product that could have sold at that store.
- The new solution mentioned above improved the existing store allocator system's absolute error rate from 24% to 20% across the entire division.
- Each percent point of error amounts to millions of Rands of stock that either gets discounted or that could have sold but was misallocated.
- Decreasing the product to store allocator's error rate down to 20% will result in many millions of Rands worth of savings or increased sales over the next few years.

Project Outcomes

- Detailed specification documentation was created so that the final solution could be translated from SQL scripts to Oracle RPAS syntax prior to implementation.
- The new solution was implemented on the Division's Oracle RPAS platform by IT.
- I am very proud of having been able to work on this project from conceptualisation to production deployment.

Built time series model to forecast the division's daily sales using AWS Forecast.

Duration: 2 months part time (August 2019 to September 2019)	**Type:** Team Project

Mr Price uses budgets, targets & forecasts to manage stock purchases, discounts, inventory & incentives. One of these forecasts is done on a monthly bases & used to forecast the next 30 days' worth of sales at a division level. This forecast is done by pooling the sales predictions of a number of experienced executives & arriving at a common daily number for the division.

This process has historically been accurate, but it is time intensive. This project was worked on by a team of 4 Data Scientists as a part time project. I was responsible for implementing the solution on AWS Forecast.

Problem Statement

- The DS team was tasked with finding an automated time series model for performing the daily sales forecasts that could either beat or be as good as the existing manual process.

Data Used

- We used aggregated daily sales data for the Apparel division, going back 5 years.
- Date specific data was also created, this data was used to flag whether it was a holiday, weekends, month ends etc.
- All training data was uploaded to S3.

☎ 0788523231 luyandadhlamini@gmail.com

Final Solution Architecture

- The model was implemented in Amazon Forecast a self-service machine learning service by AWS.
- We had an option of 3 time-series models:

1. ARIMA (Auto Regressive Integrated Moving Averages) model.
2. The Prophet time series model developed by Facebook
3. The DeepAR+ model developed by Amazon.

Solution Implementation

- When training & comparing the 3 time series model types (ARIMA, Prophet & DeepAR+) we found DeepAR+ to have the lowest RMSE (Root Mean Squared Error) against our evaluation dataset at the 50^{th} percentile point. DeepAR+ was then used as the default model.
- Another advantage that the DeepAR+ model had was that it could take in additional time series & categorical features in addition to the target variable. This means that you are able to enrich this model with info such as whether the day being predicted was a holiday or not or what the previous day's closing stock level was.

Solution Evaluation

- After several iterations & hyperparameter tuning, we found that the predictions produced by the DeepAR+ model, we never as accurate as the predictions made by the group of SMEs.
- The model consistently under performed on important dates, such as pay day or special holidays.

Project Outcomes

- It was decided to not adopt the predictions of the model into any existing workflows.
- We also found that there were inherent disadvantages to using the AWS Forecast architecture such as the need to train a new model from stretch when there is new data.

Used rule-based anomaly detection to flag orders & stock that needs attention.

Duration: 3 months (January 2019 to March 2019)	**Type:** Individual Project

Mr Price's Distribution Centre packing system requires that certain minimum pick values are filled in order for inventory items to get picked & sent to stores. This filling is the responsibility of Location planners, who are responsible setting these minimum pack values.

Due to the high volume of orders & SKUs that flow through the DC each month, a small fraction of order sometimes does not get pick values assigned to them & as a result do not get pushed out to stores.
I was assigned with creating scripts that would be able to flag unassigned orders so that they can have their values entered by Location Planners.

This project was an anomaly detection task, since only a small fraction of items were affected by this problem. I had to investigate what key factors led to items not being assigned pick values & use these as rules in my script to flag such items.

Problem Statement

- Can items without minimum pick values be identified using an automated script?

Data Used

- Store category grading tables that determine if items can go to certain stores.
- Purchase order & current stock data used by the DC to carry out pick instructions.
- Minimum pick tables that reference all items that can be picked in the DC.
- Data was retrieved from SQL server & Hadoop databases.

Final Solution Architecture

- A Python script was created that can automatically flag items that have missing minimum pick values.
- Using this script, an Excel document was created & shared with internal staff so that the missing data can be added.
- The data was automatically added to a Tableau dashboard, eliminating the need for an Excel report (subsequent implementation).

Solution Implementation

- Bash scripts were created to automatically call SQL & Python scripts, done on a recurring weekly schedule.
- SQL scripts were used to query the relevant databases & dump their output data to a shared network drive.
- Python was used to ingest & manipulate the data so that items of interest could be identified.
- Python was also used to create an Excel report as well as store the data in a table that is connected to a Tableau database.

Solution Evaluation

- The script's output was manually tested against known rules that help identify items of interest.

Project Outcomes

- The division now has an efficient automated way of flagging items that needed attention in order to get picked to stores.
- This will lead to a more efficient supply chain, with all items being dispersed to their destinations from the DC as intended.

Performed ad hoc analysis & reporting in Excel for planning teams.

Duration: As needed	**Type:** Individual Project

The Data Science team performed analysis for different units of the Mr Price Apparel division on an ongoing basis. These reports are prepared in addition to whatever main projects I was working on at the time.

Examples of the analysis I performed includes:

1. Analysing the effectiveness of a product promotion.

Curriculum Vitae of Mr Luyanda Percival Dhlamini

☎ 0788523231 luyandadhlamini@gmail.com

- This was done after a merchandise team had run a promotion on a clothing line & wanted to find out if it had led to an increase in volume sales or an increase in the average order value of customers who bought into the promotion.

2. Assessing whether incoming product order breakdowns were in line with how that product was already selling.
- This was done to assess whether any adjustments needed to be made on incoming orders so that they align with customer demand. As an example, a shoe might have been on sale for a period of 3 weeks, a merchandiser might want to know which size is selling out differently to what had been expected. They might then adjust their future order to be in line with the market trend.

3. Preparing a excel file that allows the planning team to view product sales down to an individual store level. Members of the planning unit used this file to easily see which profile of stores was selling a product well.

Created Tableau dashboards for planning teams.	
Duration: As needed	**Type:** Individual Project
I was responsible for creating dashboards that would be used by the planning team for the projects that I worked on. These dashboards were created on Tableau Server. The data source for the dashboards would be database tables stored in Hadoop. Tableau schedules would read these tables on a daily basis so that the data displayed was updated.	