Supplier Price Forecasting for Aerodine: Executive Leadership

Benji D. Magnelli, CEO of Korrolating

Western Governors University

**Table of Contents**

[**Section A - Non-Technical Manager**](#_1fob9te) **3**

[**Proposal Overview**](#_5x6alh7i5i8j) **3**

[PROBLEM SUMMARY](#_yr9872umvf) 3

[IT SOLUTION](#_2et92p0) 4

[IMPLEMENTATION PLAN](#_tyjcwt) 5

[Project Rationale](#_4d34og8) 6

[Methodology](#_17dp8vu) 8

[Description of Data](#_x582ktn5xcp) 9

[OBJECTIVES AND HYPOTHESIS](#_cz8nlgb7p0gd) 10

[**Business Factors**](#_tvoois1olttn) **11**

[IMPACT ON STAKEHOLDERS](#_f1ehmnkmyc00) 11

[ETHICAL AND LEGAL CONSIDERATIONS](#_2s8eyo1) 12

[FUNDING REQUIREMENTS](#_pkf3k63aju2c) 13

[EXPERTISE](#_8nkc593qtph) **15**

[**Section B - Technical Executive Summary**](#_566ywkpmp58b) **17**

[Project Proposal](#_ucm4q6vr6bbk) 17

[PROBLEM STATEMENT](#_it0bpfbyl0nj) 17

[CUSTOMER SUMMARY](#_dzs7wc7hhtxy) 17

[EXISTING SYSTEM ANALYSIS](#_30k4ydmfnljq) 18

[DATA](#_pzeaxjtgpndp) 18

[PROJECT METHODOLOGY](#_fy1ybkk3wx66) 19

[PROJECT OUTCOMES](#_z75v3dvxk6gv) 20

[References](#_2jxsxqh) **27**

# Section A - Non-Technical Manager

## Proposal Overview

This proposal for Aerodine’s executive business leaders outlines the problem, our proposed solution, and ancaillty data to support our solutions efficacy.

### PROBLEM SUMMARY

Aerodine is a leading aerospace and defense contractor who has recently won a large contract from the United States government to build the next-generation fighter jet. Aerodine is one of the largest aerospace and defense contractors in the nation due to its exceptional technical ability to integrate software systems, hardware components, and airframe structures into a final product. This position as an integrator helps Aerodine deliver a product quickly and efficiently to their customer.

To build a product efficiently and quickly Aerodine needs to subcontract portions of the plane’s infrastructure to other companies. Managing the subcontracts profitably has proven difficult for Aerodine because forecasting the actual cost of the materials and labor for another is nebulous.

This problem has been compounded recently, due to schedule slips on the development side of the airframe. In response, Aerodine chose to outsource more components of the plane’s airframe build to more subcontractors to maintain the deadline that they have to the government.

The management of these new subcontracts, in addition to the profit owed, is projected to increase Aerodine’s costs. Thus, Aerodine needs to operate more efficiently to achieve financial success on this program. An accurate forecast of when suppliers will invoice Aerodine is critical to maintaining this financial success. Aerodine’s current process of human forecast and analysis has produced unreliable data for price prediction.

Aerodine has reached out to--our firm--Korrolating to build a software solution that can forecast suppliers and materials costs at a higher accuracy than the current process of human analysts.

### IT SOLUTION

Aerodine has contracted with Korrolating to build a software solution that can accurately predict material and supplier costs at scale. Korrolating proposes that artificial intelligence could be built to identify trends within the data that human analysts cannot see.

Time series forecasting—which is a subset of artificial intelligence—is the study of trends within time-series data that are then used to extrapolate predictions of future values. These trends are grouped and categorized within a machine learning model or a template for forecasting the future. The efficacy of machine learning models relies on the assumption that there are systematic movements in the data--even when they are hidden to the human observer. Korrolating proposes to use a deep learning Recurrent Neural Network (RNN) as the method to identify these systematic movements within the data and build this model.

RNNs are adept at identifying relationships between data points because they simulate the learning of the human brain, with interconnected series nodes. They efficiently synthesize accurate predictions from large datasets, making them a choice for many machine learning workloads (Ciaburro & Venkateswaran, 2017).

Korrolating recommends a deep learning neural network for Aerodine’s price prediction because of its high performance as a machine learning paradigm. Ortner demonstrates—in an independent investigation—that Neural Networks outperform other machine learning algorithms in terms of speed and accuracy (2020).

This speed will allow the AI to learn consistently and predict often without a drain on computing resources, thus performing at scale. The ability of deep learning to handle large data sets will be showcased when Aerodine’s supplier base grows further. An increase in price prediction accuracy will result in Aerodine’s increased ability to manage supplier and material costs which may result in a more profitable enterprise.

### IMPLEMENTATION PLAN

Accurate predictions for material prices are critical to the success of Aerodine’s business. To Korrolating achieves success through this paradigm: identify assumptions, exit criteria, collect data, build model, test, and deploy. Korrolating will start by predicting a single material price into the future to demonstrate the general applicability and scalability of the solution to other materials.

Korrolating will begin by meeting with Aerodine to discuss which material is the most appropriate for this first analysis. Korrolating recommends using a material that is generally viewed as unpredictable and that Aerodine currently spends a large amount of money on. If accurate predictions are produced successfully, Aerodine can assume that Korrolating’s model can give insight to Aerodine on all types of suppliers.

Korrolating works with Aerodine to determine exit criteria for the project. Aerodine will define the question ‘what are the metrics and the acceptable levels at which the model’s predictions produce actionable data for Aerodine.’ These meetings can be done in conjunction with the material discussions.

In the next phase, Korrolating builds the deep learning model based on the time-phased price history found in the previous section. Korrolating’s data science team collects a complete dataset for the chosen material’s price over time and begins the coding of the model. Korrolating will recursively test the model until it consistently meets the exit criteria set in previous steps.

Once the model reaches acceptance testing standards, correlating will deliver the product. Korrolating will send members of their data science team to Aerodine's headquarters to assist in the handoff of the new prediction tool. The data science team will train the IT professionals at Aerodine on how to operate, interpret and manage the price prediction machine learning tool. A major training component will be how to load new pricing histories into the tool.

## Project Rationale

As mentioned previously in the project summary, Aerodine's ability to forecast material and supplier costs has proved unsuccessful resulting in financial damage to the company. If left unchecked, Aerodine could see their generation fighter program go negative on profit. Aerodine can negotiate with their customer to remove certain items from the contract that reduce cost while still hitting scheduled delivery day. These items—part of a modernization effort—could be added to the build at a later time.

However, Aerodine is unclear on the actual cost of each component and supplier. This ambiguity increases the likelihood that Aerodine over allocates budget for certain suppliers and under allocating for others. One constraint is that the budget cannot be from one supplier to another to cover overruns. Under allocating a budget on non-negotiable items leads to negative profit for the company. And inversely, over-allocation budget on negotiable items can lead to a trapped budget which is essentially a waste.

Korrolating’s machine learning solution proposes to give an accurate price of suppliers and materials. So that Aerodine’s leaders can make the informed business decisions necessary to keep the company profitable while still delivering the plane on time.

Aerodine’s current analysts are unable to predict what the future cost the satisfaction of leadership. Korrolating’s machine learning model will be able to identify hidden trends within the time series data that can accurately predict the future price within a predetermined margin of error. These trends will be extrapolated into the future allowing artificial intelligence to accurately predict what future costs will be. These predictions can be used in business decisions by Aerodine’s leadership.

## Methodology

The SEMMA approach will be used to construct our machine learning models. SEMMA was chosen due to its suitability with the Agile project management approach, which we also used on this project. Because SEMMA eliminates the first planning phases that CRISP-DM considers necessary, we spend more time in the exploratory phase. We are convinced that this data investigation will provide richer data analysis in the material prices that we, as engineers, could not have conceived on our own. Rather than specifying a rigorous success criterion at the outset, we chose to navigate the characteristics of the data towards the project goal of accurate time series forecasting.

Sample: One data requirement is critical mass for the data set--to enable precise training and prediction. Critical mass is the reason for the second material type stipulations in the first section--the material must be a large portion of Aerodine’s costs. A material that is widely used in an industry will generally have a standard pricing history across vendors resulting in a scalable prediction. Sampling in time series prediction is simple, it is deleted by date. We select the first 70% for the training model and the last 30% of the data for testing.

Explore: We will load the training data into a python library called MatPlotLib for visualizations. MatPlotLib gives the data engineers the flexibility to manipulate the data quickly while saving their visualizations for future use. One aspect we will visualize is the percent change between data points, are they normally distributed or not.

Modify: Data quality is critical for any time series forecasting— especially so of deep learning models. We do foresee any gaps in data because of the material that we have chosen to research. Materials that are widely used have a full price in history that is accessible with minimal effort. However if any data discrepancies are identified in the exploration phase, we exclude those outliers this time.

Model: This phase is the most well-known of the SEMMA approach. Here our team feeds the cleaned data through our machine learning framework, which will then produce a model. Here the machine begins its Deep Learning analysis of the data--identifying recurring trends within the dataset.

Assess: Our engineers then load previously mentioned test data into this model and map its predictions to determine efficacy. How accurate are its predictions versus the actual value? Visual mapping of production values versus actual values gives the user a bearing on how close those two values are. But we will also use statistical frameworks to determine the exact delta between these two.

### DESCRIPTION OF DATA

As previously stated, this solution will demonstrate the efficacy of our software on a single material first, which will be used as the baseline of effectiveness for other material costs. Korrolating recommends using a material that is generally viewed as unpredictable, that Aerodine currently spends a large amount of money on and has available data. Korrolating states that oil fits all three of these requirements.

First oil prices are seen as generally unpredictable due to the few who control them. The Organization of the Petroleum Exporting Countries (OPEC+) produced more than 152.1 billion barrels of crude oil from 2009 to 2018 (OPEC, 2021). Compare this to non-OPEC countries where the production was 24.6 billion barrels over the same period (OPEC, 2021). OPEC+ produced six times greater oil than non-OPEC countries. In addition, OPEC has more than 82% of the world's oil reserves versus non-OPEC countries (OPEC, 2021).

This command on production and reserves gives OPEC the power to set prices at any level that they deem fit. Swings in price are often unpredictable and must be taken by the consumer as there are not many non-OPEC that cannot match production at a lower cost.

Secondly, an airplane is heavily reliant on crude oil for fuel—which is used in not only the testing of the airworthiness of the craft but the development of the plane. Much of large Aircraft manufacturing tools use gasoline for fuel, and also various types of oils for lubricants. Aircraft manufacturing is very heavily reliant on oil prices, thus knowing what oil price would be is beneficial in financial planning.

Finally, oil is used in almost every industry to produce almost every product that we consume in the United States. Since oil is so widely used, its price is transparent. Our data science team can easily collect historical data of oil prices over the past 30 years. This data is accurate and can be used to forecast the future.

### OBJECTIVES AND HYPOTHESIS

As stated previously, the goal of the project is to build a model that will accurately predict oil prices of the future within a degree of standard error. These predictions need to be timely and accurate so that business leaders can make informed business decisions.

Korrolating believes that there are underlying patterns within the data that a deep learning artificial intelligence can identify. These patterns of price changes are hidden to the human analyst but through many epochs–or iterations—of deep learning, our model can identify these successfully. These can be used to inform business decisions.

## Business Factors

### IMPACT ON STAKEHOLDERS

Korrolating foresees impacts to internal teams based on the adoption of this proposed supplier price prediction system. The engineering team and finance team may need to adjust their staffing mixes. The Executives should be briefed on the basics of AI/ML.

Aerodine’s software engineering team may need to hire data science professionals to manage this tool once deployed. It depends on the scale mix the Aerodine’s currently employs—which we do not know. We suggest Aerodine employ two full-time data science professionals who can manage this tool. Management activities include pulling in updates, data management, data security, running the model on new data sets, and the dissemination of predictions to the proper channels.

Aerodine's leadership team must understand that these predictions are being generated by a machine learning model. Inherently artificial intelligence models learn from past trends to generate predictions for the future. Correlating recommends that aerodynes executives be briefed on how artificial intelligence works at an elementary level to set proper expectations based on performance.

Aerodine’s finance team will also see changes. When correlating demonstrates the performance level of this price prediction artificial intelligence, manual predictions may prove obsolete. Aerodine may choose to re-deploy these many of its finance professionals into other areas of their business. However, some will still be needed to adjust price forecasts for the factor that is not taken into account by the model. These could be political administration changes, supply chain issues, natural disasters, etc.

### ETHICAL AND LEGAL CONSIDERATIONS

Korrolating no ethical or legal challenges with this purview of this project due to the availability of data. Oil prices are publicly available, without the need for licenses or other legal agreements to access. However, after our Data science team successfully builds models to predict future prices within acceptance criteria ethical and legal challenges arise. After delivery, Aerodine can load classified suppliers and other proprietary material into this model. Aerodine could face lawsuits, financial penalties, and even disbarment from future defense contracts if this future data were to get compromised via unauthorized users.

To protect against these potential consequences, Aerodine can operate the artificial intelligence locally on an intranet. There will be no direct internet connection from the deep learning model to Korrolating's servers resulting in increased security for this data. Any future updates will be delivered to Aerodine via a secure, one-way VPN channel to a staging environment—further restricting the possibility of data leaks.

### FUNDING REQUIREMENTS

| **Resource** | **Description** | **Cost** |
| --- | --- | --- |
| **Data Scientist (2)** | Execute the bulk of the product build: data preprocessing, data splitting,  data visualization, model training, model testing and model deployment | Annual salary 110k / year (Payscale, 2021)  x 3 months  x 2 people  = $55,000 |
| **Experienced DBA / Data Engineer** | Manage the transfer of data to our servers. Also manages ETL process | Annual salary 90k / year (Payscale, 2021)  x 3 months  = $22,500 |
| **TensorFlow** | An open source ML framework for data cleaning, data splitting, training of the models | Free |
| **Keras** | Open source deep learning framework built on top of Tensorflow. | Free |
| **Azure Data Share** | Used to move massive datasets between organizations | $730 /month (Microsoft, 2021)  x 3 months  = $ 2190 |
| **Azure Managed PostgreSQL2**  2 TiB General Storage  Hyperscale  Highly Available (node duplication) | General storage for massive datasets | 3,586/ month (Microsoft, 2021)  x 3 months  = $10,758 |
| **Azure Virtual Machines**  256 GB RAM & 512GB Temp storage | Scalable computing to run the model training without the expense or need to manage servers | $1635 / month (Microsoft, 2021) x 3 months = $4905 |
|  | **Total** | **$95,353** |

Above source is (AltexSoft, 2019) unless otherwise stated

### EXPERTISE

Korrolating is a leader in the application of artificial intelligence towards business predictions for over 10 years. Korrolating over 200 satisfied clients including Microsoft, Lyft, Snowflake, UPS, Starbucks, Florida Power & Light.

Our process focuses on bias-free data analysis, which leads to higher prediction accuracy. Though we build on open source technology that other firms use, we invest deeply in the education of our employees on these tools. Our teams have presented at both Tensorflow and Keras’s annual developer conferences – demonstrating our expertise in the field.

# Section B - Technical Executive Summary

## Project Proposal

### PROBLEM STATEMENT

Aerodynes current supplier forecasting techniques produce subpar results when compared to actual costs. By relying on human intuition alone, the predictions are skewed towards irrelevant factors leading to a poor prediction of the actual cost.

Human financial analysts can detect large trends, such as the general direction of a stock price over a 5 year period. But humans often fail to see small patterns that make up these large trends. Korrolating’s Recurrent Neural Network learns how each pattern affects the future state. By remembering certain patterns and their impact on the future state, Korrolating’s Recurrent Neural Network has historically predicted the actual cost of supplier invoices within a margin of +- 4%. This is far superior to aerodynes financial analyst performance of +- 36%.

### CUSTOMER SUMMARY

Aerodine’s overreliance on human intuition demonstrates its lack of robust data insights across the organization. Korrolating’s LSTM Recurrent Neural Network can neatly package the insights inherent in the data aerodyne already operates on.

Aerodine does not possess the technical expertise to stand up a similar project in-house. Hiring out the technical personnel—and those to manage the project—is expensive in time and dollars. Korrolating’s solution is robust enough for improvement customization, simple enough for newcomers to learn, and it's +- 4% accuracy is trustworthy for decision making. We do assume that Aerodine has—or plans to hire—the personnel to maintain this tool.

Korrolating’s proposed solution is simple to manage for experienced data science professionals. The LSTM RNN integrates with Aerodine’s current tech stack with ease as it operates as a self-contained unit--one can access insights simply via an API.

### EXISTING SYSTEM ANALYSIS

Aerodine currently lacks a software forecasting tool. Korrolating’s proposed supplier invoice prediction tool will replace Aerodine’s current process of financial analysts predicting manually.

### DATA

As previously stated, Korrolating’s software will demonstrate it’s efficacy on a single data set first which will be used as the baseline of effectiveness for other material costs. Korrolating recommends using crude oil prices because oil is generally viewed as unpredictable, Aerodine spends a large amount of money on and has a pricing history available.

Korrolating will use oil pricing history over the past 34 years originally published by the US Department of agriculture. Our data source is a Kaggle repository, where Abusalah formatted the data for easy processing (Abusalah, 2021).

This dataset is a daily time series containing date and closing price for a single barrel of crude oil. The dataset spans from May 20, 1987 to January 21, 2021. As originally published by the US Department of Agriculture, this data set is licensed under ‘US Government Works’. Thus, our selected oil pricing history is copyright-free and available for all commercial uses without need for acknowledgment.

This oil price dataset will be packaged within the Jupyter Notebook upon delivery for demonstration purposes. In the future, Aerodine will replace this oil pricing history with the pricing history from other materials and suppliers.

### PROJECT METHODOLOGY

For the course of this project, Korrolating will implement the agile software design methodology. We choose this for its flexibility, which is imperative for data science projects. We have our hypothesis—as stated in the previous section—for how the data will be structured, but there is uncertainty until our data science team truly begins the analysis of this data set.

The agile methodology demands meetings frequently with the customer to ensure that the software development team is incrementally nearing the requirements of the customer. Our meetings will be held weekly to ensure that our data product meets the quality standards that Aerodine sets. These meetings are an opportunity for Aerodine’s to shift our engineers if we have strayed too far from the approach Aerodine envisioned.

Each meeting will begin with correlating demoing the previous weeks' progress deliverable. The size of the deliverable will vary from week to week, but we commit to progress each week. Then we ask for feedback on the deliverable, against the acceptance criterion that we agreed upon. Feedback could be in the categories of functionally, usability, or any agreed-upon criterion. After the demo, Korrolating will share the current week’s sprints tasks as well as schedule any further meetings as necessary.

The agile methodology also allows for frequent change. Should requirements of Aerodine change, correlating can accommodate via this methodology. Scope changes we have seen in the past are the inclusion of additional data sets, the need for models that run with lower processing demand than deep learning (using statistical models such as AREMA), etc.

### PROJECT OUTCOMES

Upon delivery of this project, Aerodine’s data science team will be able to operate a material price prediction tool simply.

This tool provides a graphic user interface via a Python Jupyter Notebook accessible by the company’s intranet. Access control features will be included within Jupyter Notebook to increase the likelihood of data integrity. Everyone in the organization will be given read-only access, and the data science teams of Korrolating and Aerodine will be given write access.

As previously stated the delivered Jupyter notebook will include the oil prices data for demonstration purposes. For documentation purposes, Aerodine will be provided with user guides, installation instructions. The demo notebook will contain our example LSTM using oil prices—this also serves as documentation. Korrolating also will provide a project schedule summary with milestones objectives and deliverables.

### IMPLEMENTATION PLAN

The coding of this project will follow the bottom-up approach. This approach will prioritize the data source which is critical for this data science project. We will develop the database connection, transform the data for processing, train the LSTM, predict based on the model, and visualize our predictions.

First, we connect to our data source—the Brent oil prices dataset. The data file is a comma-separated values format, which is easy to read using the Python library Pandas. Pandas reads in the data as a data frame—which. Data frames make manipulations easy in the preprocessing stage.

Next, we prepare Pandas' data frame for processing. First, we will partition the data set into test and training sets. Our training set will be the first 70% of the whole price history and the training is the remaining 30%. We then scale the data down between zero and one using MinMaxScaler from the Python library sklearn because neural networks cannot accept inputs outside of those boundaries. It is unknown which algorithm is used by sklearn but scaling is frequently done using sigmoid or tanh.

Once the data is prepared for processing we will configure our model. The machine learning algorithm we will use is a Long Short Term Memory (LSTM) Recurrent Neural network. One major application of LSTMs is time series prediction due to their ability to recall past patterns. The goal of training an LSTM is for it to identify which trends need to be stored in long-term memory, and which trends are only relevant for the short term thus forgotten later. This specific type of neural network has two gates within each cell, a long-term and short-term gate. These gates determine if the current state remains in either of those states thereby getting passed on to the next cell. This process is completed one time per each of the 6 epochs we will run. We foresee that after each epoch we will see a more accurate LSTM model.

After the LSTM has been trained on the training data, it will predict the future values for the remaining 30% of the data set. These LSTM predictions will then be plotted against the test data's actual values to determine the model's efficacy. We would use qualitative analysis in the form of visualizations, and quantitative analysis in the form of Root Mean Standard Error.

### EVALUATION PLAN

The primary statistical metric that will validate our proposed LSTM is the root standard mean error (RSME). In data science, RMSE has a double purpose: To serve as a heuristic for training models and To evaluate trained models for usefulness/accuracy (Moody, 2019). RSME is used in relation to variance. An RSME that is less than 5% of the variance is acceptable for this project. We will plot the loss of the LSTM Neural Network over each epoch.

### RESOURCES AND COSTS

This Jupyter Notebook application needs a simple setup. The application can be run on any server with enough band and ram to support all the traffic. The Breton oil price dataset is self-contained within the Jupyter notebook, thus eliminating the need for databases. All of the libraries within this project are open source so they are free of any licensing costs. Total pricing is estimated to be ~95,000 with a complete breakdown in the FUNDING REQUIREMENTS section of the non-technical letter.

### TIMELINE AND MILESTONES

Development of this project is projected to take 60 business days. The main schedule constraint is the software engineering of the LSTM model. Project milestones are listed below.

| Task | Parent Task | Resource Assigned | Start | End | Task |
| --- | --- | --- | --- | --- | --- |
| 1 | - | CEO | Day 0 | Day 0 | Proposal Accepted |
| 2 | 1 | CEO | Day 1 | Day 1 | Kickoff meeting with Leadership |
| 3 | - | Data Science | Day 1 | Day 1 | Collect Data |
| 4 | 1 | Data Science | Day 2 | Day 4 | Requirements Gathering with Dev Team |
| 5 | 4 | Data Science | Day 5 | Day 9 | Acceptance Testing Standard Developed |
| 6 | 5 | Data Science/CEO | Day 7 | Day 7 | Weekly Status 1 with Demo - Acceptance Testing Standards |
| 7 | 6 | Data Science | Day 10 | Day 24 | Prediction Feature Development - LSTM Model Development 0.1 |
| 8 | 7 | Data Science/CEO | Day 14 | Day 14 | Weekly Status 2 with Demo |
| 9 | 7,8 | Data Science/CEO | Day 21 | Day 21 | Weekly Status 3 with Demo / Data Cleaning and Validation |
| 10 | 7 | Data Science | Day 24 | Day 31 | Models training on Training Data |
| 11 | 9 | Data Science/CEO | Day 28 | Day 28 | Weekly Status with Demo 4 |
| 12 | 7, 10 | Quality Assurance | Day 32 | Day 32 | Prediction Feature Testing w/ Fully trained Model 0.2 |
| 13 | 4 | Software Engineer | Day 33 | Day 38 | Continuous Improvement API Built 0.1 |
| 14 | 11 | Data Science/CEO | Day 35 | Day 35 | Weekly Status with Demo 5 |
| 15 | 4 | Quality Assurance | Day 39 | Day 43 | Continuous Improvement API Tested 0.1 |
| 16 | 14 | Data Science/CEO | Day 42 | Day 42 | Weekly Status with Demo 6 |
| 17 | 12 | Quality Assurance | Day 43 | Day 47 | Prediction Feature Finalized - 1.0 |
| 18 | 17 | Data Science | Day 48 | Day 50 | User Training |
| 19 | 17 | Data Science | Day 51 | Day 51 | LSTM Models 1.0 loaded to Prod Servers |
| 20 | 15 | Software Engineer / Quality Assurance | Day 55 | Day 55 | Continuous Improvement Launched |
| 21 | 16, 19 | Data Science/CEO | Day 42 | Day 42 | Weekly Status 7 with Feedback |
| 22 | 15, 19, 21 | CEO | Day 60 | Day 60 | Follow up with Leadership |

# **Section D - Documentation**

## Post Implementation Report

### BUSINESS VISION

The supplier price forecasting for Aerodine project's goal is to use machine learning to predict supplier’s future prices which stabilize forecasts for financial planning. Our machine learning algorithm of choice is a deep learning LSTM, a tool that specializes in recognizing linear patterns such as the time series problem we face here. The inputs to our algorithm are historical prices of the supplier, and the output will be a price prediction into the future. Our goal is to produce a machine-learning algorithm whose forecast accuracy is above the acceptable threshold for Fidelity. Surpassing the stress hold means our algorithm can be used to inform business decisions.

### DATA

Our oil price data set originated from the US Department of Agriculture is attached within the project. Traditionally time series data sets do not need to be cleaned as there is only a single relationship: date and price. Every day was accounted for so no cleaning was necessary. Raw data file ‘oil prices.csv’ is attached in the submission of this project.

### DATA PRODUCT CODE

To facilitate development and ease of use, our data product has been split up into five sections. Those sections load the data, describe data, prepare data, train the model, visualize results.

During the load the data section, we prepare the input for the algorithm. After our third-party libraries are imported successfully, the software prepares to focus on the data. For demonstration purposes, the ‘oil prices.csv’ data set is attached inside the Jupiter notebook, but in production, Aerodine will load many different supplier data sets into this tool. The data set is then read into a Pandas data frame for manipulation. Subsequently, the data frame is split into training and test data. The first 70% of the time series goes into the training data set and the final 30% is for testing.

The descriptive section gives context to the data for our engineers before processing. We use two statistical descriptive methods for this context: Standard deviation, and percent change plots. Standard deviation measures how dispersed the values are from the mean value. Standard deviation is compared to the mean of the data set. A large standard deviation versus the mean could indicate that the data set will perform poorly for prediction because values are too dispersed. Our Standard deviation is 46.35 compared to the arithmetic mean 31.16. Our standard deviation is within 30% of the mean, thus is heuristically acceptable.

Secondly, we calculate the percent change from each value to the next. The percent change values are then plotted on a histogram using MatPlotLib. As shown by the plot, most values are within the +10% or -10% change with some extreme changes of -92% and +37%. This illustrates the wide variance within the data- particularly the drops in price from 2008 & 2009.

After understanding the data, we prepare it for the model. The software uses the MinMaxScaler from the Python library sklearn to scale the values between 0 and 1. It is unknown which algorithm is used by sklearn but scaling is frequently done using sigmoid or tanh. Be prepared that it cannot be loaded into the model.

The model we have chosen is a long short-term memory (LSTM)—a deep learning framework that specializes in pattern storage. We configured the model to look at a rolling five-day window for a single feature: the price of oil. The model has 30 hidden layers of deep learning with a single final layer prediction node. The model is configured to train five times—or five epochs—on the dataset. Then the model predicts a price for each period in the test data set. The predictions are on the scale back to their US dollar equivalent for the visualization section.

In the visualization section, we plot the results of the LSTM models predictions. The first graph plots the loss—the amount of learning—at each epoch. We noticed a logarithmic growth, with large amounts of learning jumps at the beginning followed by demising returns. We also plot the LSTM predictions versus the actual values. The LSTM values are colored purple while the actual values are in blue.

### HYPOTHESIS EVALUATION

Our hypothesis for this project was that the deep learning model would be able to identify patterns within the data. These predictions need to be timely so that business leaders can make informed business decisions.

Our model accurately identifies patterns within the data. This is demonstrated by the training loss graph in the visualization section. Each epoch, the model identified the trends thus was to fit its hidden algorithm more precisely to the data set. The model's efficacy in using said patterns is further described in the ACCURACY ASSESSMENT section.

Our model runs in a timely manner. Even with running 6 epochs, The LSTM runs in approximately two minutes on the test environment. Our test environment is Google CoLab powered by Google Python 3 Compute Engine. We used 46.2 GB of disk space and 1.02 GB of RAM to run the whole machine learning application. These swift training sessions will allow for agile updates to the data set. Quick updates to the data set for lower business leaders to make informed decisions in a timely manner. Based on this assessment we can accept the hypothesis as factual.

### ACCURACY ASSESSMENT

Our goal for this project was to build a model that will accurately predict oil prices of the future within a degree of standard error.

In one run of the model, we achieved an RMSE Error: 2.807 versus a Variance of 1034.484. If the RMSE error is less than 5% of the variance the model is highly accurate. This RSME is 0.27% of the variance. In addition, the graphs in the visualization section show how accurate the model was versus the test data. The plotted pink LSTM line matches almost perfectly with the plotted blue test data line.

### EFFECTIVE VISUALIZATIONS AND REPORTING

Aerodine has communicated its acceptance criteria for a high fidelity model. Visualizations help humans give context to these statistical rates.

The descriptive section shows a histogram to plot the percent changes from value to value over time. This visualization is effective because it clearly shows the distribution of the data. The majority of oil prices move within plus or -10% intraday. And the outlier changes of -92% and +37% flatten the black bell curve line to be indistinguishable from the bottom of the graph. This illustrates the large variance in oil prices over the past 40 years.

The second descriptive visualization is a plot of the data’s autocorrelation. “Autocorrelation is a mathematical representation of the degree of similarity between a given time series and a lagged version of itself over successive time intervals” (Smith, 2021). Calculations for train and test data autocorrelation were astounding. The date it was incredibly auto correlated—nearly perfectly out to 15 period lags. Average autocorrelation for the training data is 0.994894346148506; testing data is slightly more correlated with an average autocorrelation factor of 99.61147852008113. Data with the autocorrelation factor close to 1 means past values or an accurate prediction of future state. With an autocorrelation factor of .99+ both data sets are candidates for time series prediction.

The Visualization Section displays our model's performance after training and predictions. The first visualization is a graph plotting the training loss after each epoch. This graph gives context to how quickly the LSTM improves its accuracy—more so than looking solely at the numbers. The curve of the graph follows a logarithmic distribution with leaps in learning followed by a flattening curve towards zero towards the end.

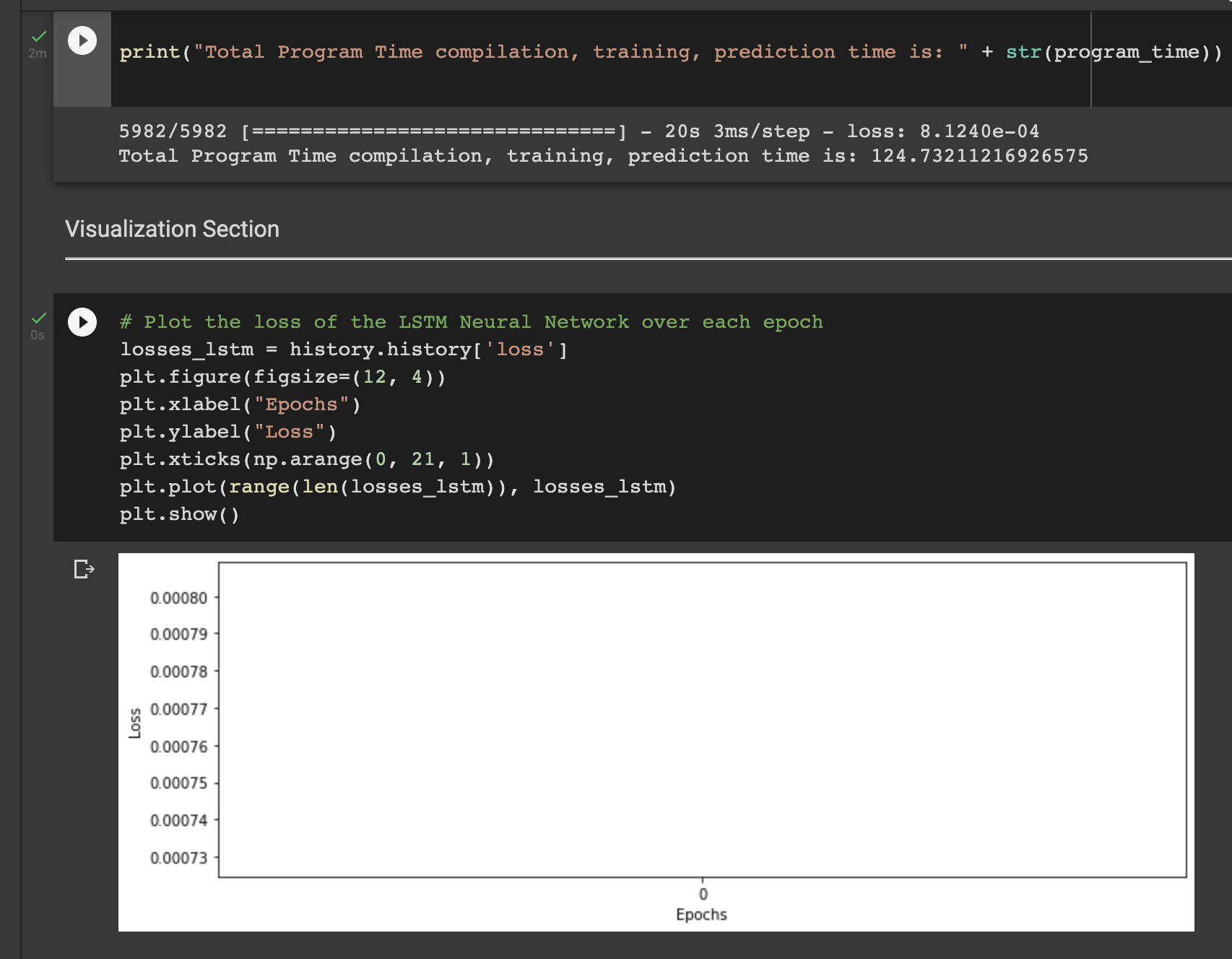
The second visualization is a two-line graph comparing the increase in time versus the decrease in training loss (better fitting model). Time increase is shown on the left Y-axis in green and the percent change in learning loss is on the right Y-axis in blue. Time increase for processing appears to scale upward linearly. This apparent linear aggression is intuitive because the average time for an epoch is roughly 19 seconds no matter how many we execute. After the training, the testing will take the same amount of time because the LSTM is predicting against the same number of testing values no matter how much it is trained.

Similar to the loss plotting graph, the percent change in loss rate decreases fast then approaches zero. This graph is another way to show where diminishing returns begin. Between 5 and 7 epochs the learning rate increases to ~0 and continues this way throughout subsequent epochs. This visualization shows that between five and seven epochs will produce the majority of the learning on this data set. Thus 5 to 7 epochs would be sufficient training should Aerodine have minimal computing resources available. But as previously mentioned 10 to 12 would be optimal.

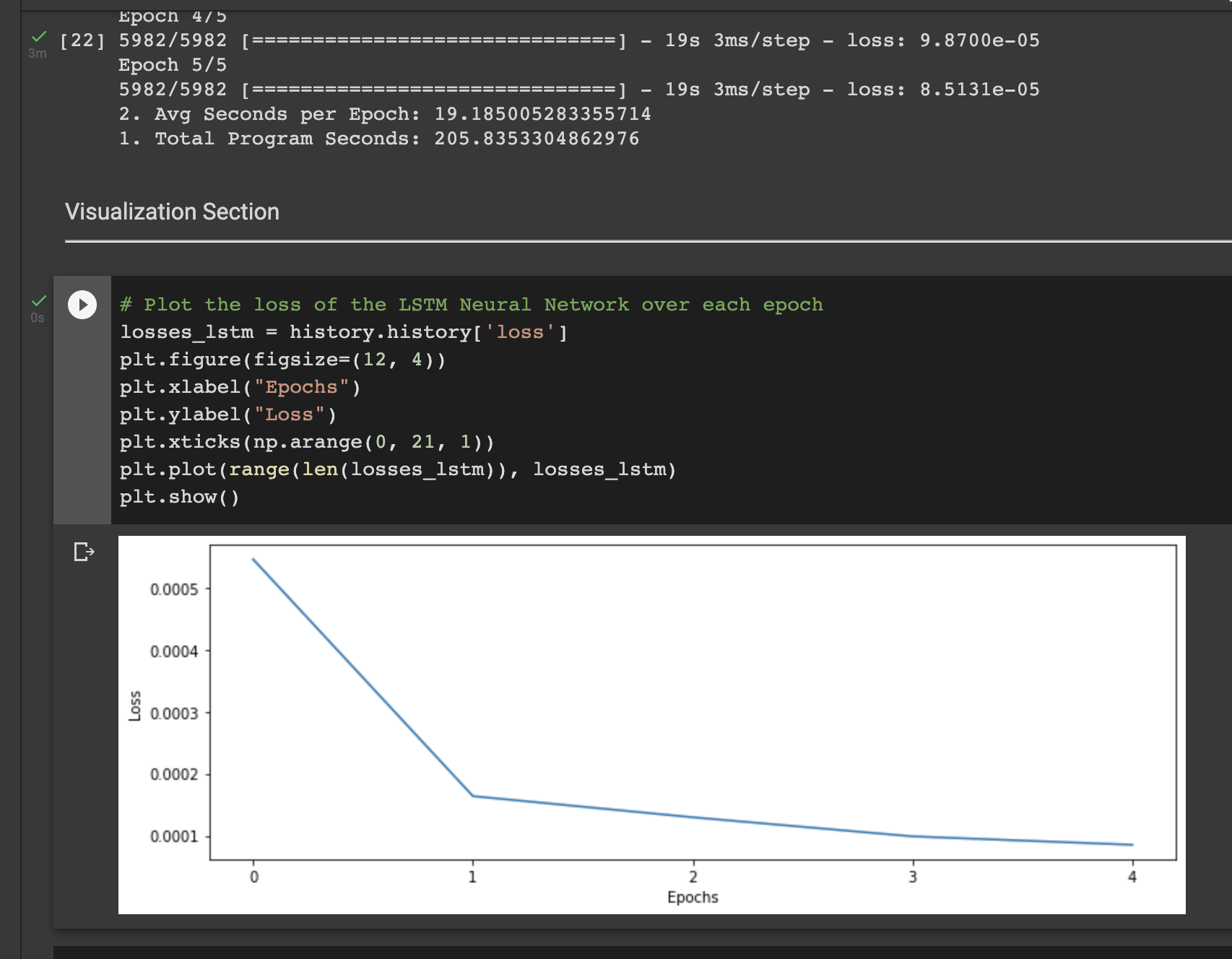
### APPLICATION OPTIMIZATION - EPOCHS

One optimization for this project is the number of epochs per training session. Our data science team balanced the processing time with diminishing returns of loss on each subsequent training. The total time metric includes the model configuration, training, and prediction which gives a holistic view of ML processing times. We tested 1, 5, and 10 epochs to attempt to find a clear intersection point of learning loss and total time.

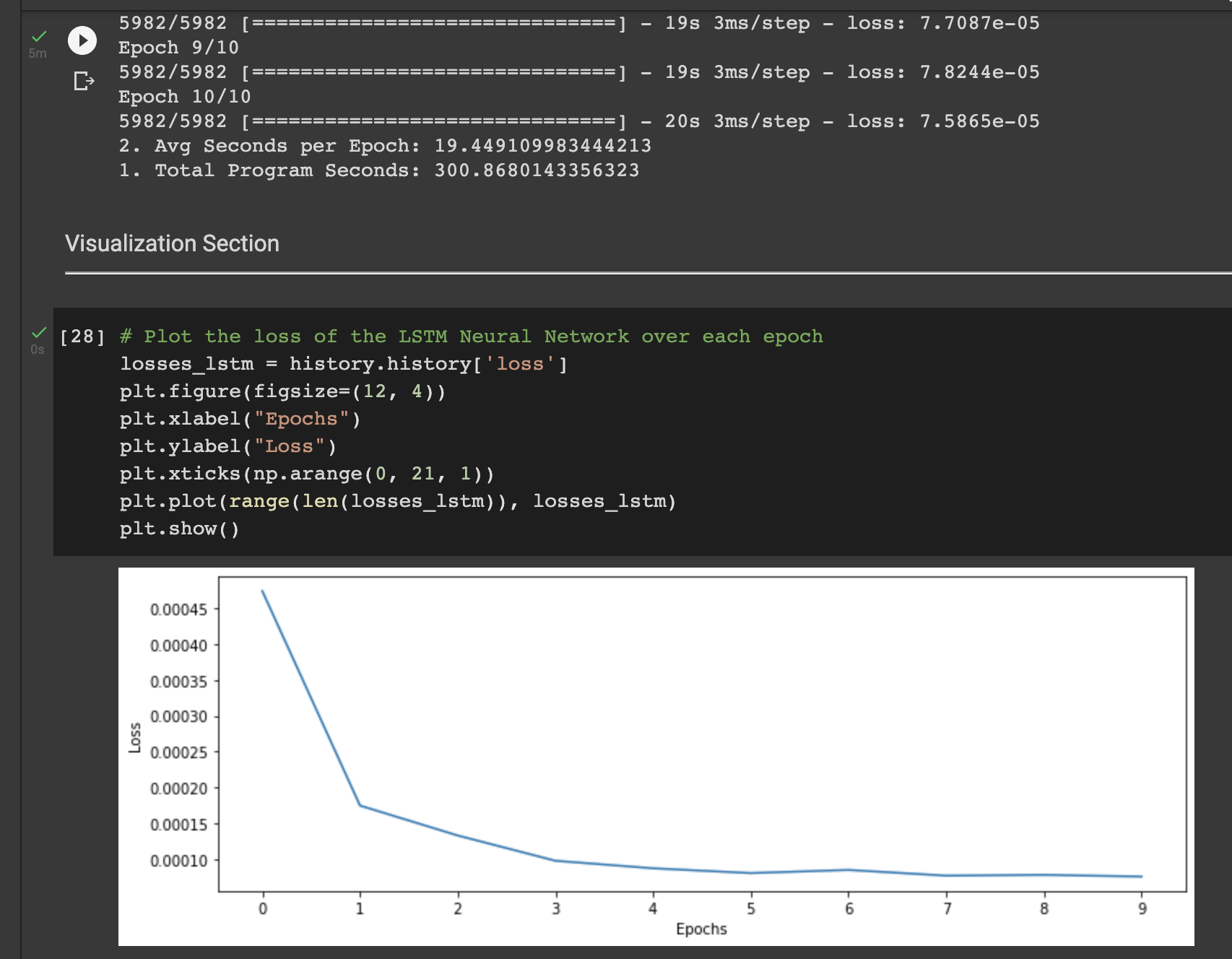
One epoch runs the fastest, but produces the highest training loss—or the model has learned the least. In this particular run, the loss was 7.6687e-04, 20 seconds per epoch, and took 124 seconds to run in total.



Five epochs significantly decreased training lost early and began to a logarithmic curve forming. For this particular run, the loss on the final epoch was 8.5131e-05. The model trained at a rate of 19.1 seconds per epoch and took 206 seconds to run in total.



Ten epochs decreased significantly over the first half of the session. Then the model actually got worse on the 6th epoch but resumed learning on the 7th. In this particular run, the loss on the final epoch was 7.5865e-05. The model trained at a rate of 19.45 seconds per epoch and took 300.1 seconds to run in total.

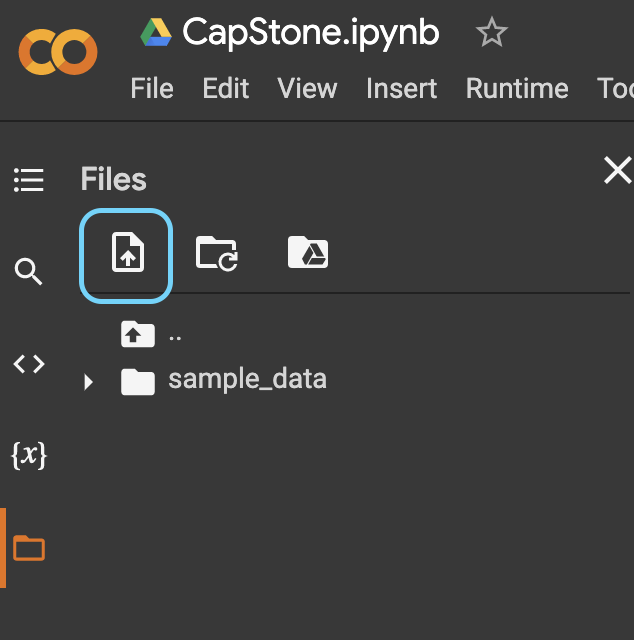


Twenty epochs decreased significantly visibly throughout the first 5 epochs. Then slightly over the next 5. Learning rate change approaches zero around epoch 12 for this data set. In this particular run, the loss on the final epoch was 6.8840e-05. The model trained at a rate of 18.16 seconds per epoch and took 491.28 seconds to run in total.



Our conclusion is that 10 epochs are the optimal number of iterations for this dataset. Less than 10 epochs do not achieve an optimal learning loss for the model. However, greater than 10 he parks produces diminishing returns in terms of learning loss that still demand a linearly growing amount of computing resource time.

### QUICK START GUIDE

* Navigate to Jupyter notebook
  + [Open link](https://colab.research.google.com/drive/1lYr9_HFMbaqbz_q4UeMb6H6xd58giU9B?usp=sharing)
  + Upload attached Jupyter notebook to Google CoLab
* Upload data
  + Upload the attached ‘oil prices.csv’ to the Google CoLab
  + 
* Run Code
  + Individually: Each Block can be run individually by clicking the play arrow
  + All: Runtime > Run all

#### 

#### 

### 

# References

Ortner, A. (2020, October 13). Top 10 binary classification ALGORITHMS [a beginner's guide]. Medium. https://medium.com/@alex.ortner.1982/top-10-binary-classification-algorithms-a-beginners-guide-feeacbd7a3e2.

Venkateswaran, B., & Ciaburro, G. (2017). Neural Networks with R. Packt. Packt. <https://subscription.packtpub.com/book/big-data-and-business-intelligence/9781788397872/1/ch01lvl1sec27/pros-and-cons-of-neural-networks>.

*OPEC share of World Crude Oil Reserves*. OPEC. (n.d.). Retrieved October 29, 2021, from https://www.opec.org/opec\_web/en/data\_graphs/330.htm.

Tensorflow. (n.d.). Tensorflow/Tensorflow: An open source machine learning framework for everyone. GitHub. https://github.com/tensorflow/tensorflow.

Keras-Team. (2021, August 9). Keras-Team/Keras: Deep learning for humans. https://github.com/keras-team/keras.

Trifacta. (2021, August 12). Pricing. Trifacta. https://www.trifacta.com/pricing/

Microsoft . (2021). Pricing calculator: Microsoft Azure. Pricing Calculator | Microsoft Azure. https://azure.microsoft.com/en-us/pricing/calculator/.

Tableau. (2021). Pricing for data people. Tableau Pricing for Organizations . https://www.tableau.com/pricing/teams-orgs.

PayScale. (2021). Average experienced Database ADMINISTRATOR (dba) Salary. Average Experienced Database Administrator (DBA) Salary. https://www.payscale.com/research/US/Job=Database\_Administrator\_(DBA)/Salary/ad9290f6/Experienced.

PayScale. (2021). Average mid-career data scientist salary. Average Mid-Career Data Scientist Salary. https://www.payscale.com/research/US/Job=Data\_Scientist/Salary/29e9d532/Mid-Career.

Abusalah, Mustafa. ([October, 2019]). Brent Oil Prices, version 5. Retrieved Oct 1, 2021 from <https://www.kaggle.com/mabusalah/brent-oil-prices>.

Moody, J. (2019, September 6). What does RMSE really mean? Medium. Retrieved November 3, 2021, from https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e.

Smith, T. (2021, November 11). What is autocorrelation? Investopedia. Retrieved November 11, 2021, from https://www.investopedia.com/terms/a/autocorrelation.asp.