Adrian Ng MSc.

Graduate Data Engineer

Email: contact@adrian.ng
Mobile: +447766336972

Summary

Computer Science graduate seeking junior-level Data Engineering roles. In particular: opportunities that apply and grow my *Java* skill-set, which has been utilised throughout my postgraduate studies and in solo projects since then. Prior to postgraduate study, my technical expertise was in *SQL development*. My recent role as a *Data Analyst* was a brief but beneficial learning experience: I flourish when writing code and solving technical problems of that nature. Thus naturally follows a career in programming.

Education

- **Master of Science in Data Science and Analytics** — with Distinction
  Department of Computer Science, Royal Holloway, University of London — *Sept. 2016 – Dec. 2017*
- **Bachelor of Engineering in Mechanical Engineering** — Upper Second Class with Honours
  School of Engineering, King's College London, University of London — *Sept. 2007 – July 2010*

Java Projects

- **Dissertation: Implementation of Value at Risk (VaR) measures in Java**
  (https://adrian.ng/java/var/) (https://github.com/Adrian-Ng/VaR)
  This project implements various approaches to estimating *VaR*, a measure of risk. These are: *Model Building*, *Historical Simulation*, and *Monte Carlo Simulation*. In addition, the following approaches to estimating market variance were implemented: *Equal Weighted*, *Exponentially Weighted Moving Average*, and *GARCH(1,1)*.
  - **Object Oriented Design**
    As we have a number of approaches to estimating both *VaR* and *variance*, abstract methods `getVariance()` and `getVaR()` were defined in abstract classes but implemented in child classes. Instantiation of these child classes is handled via the *factory method*.
  - **Concurrency**
    The *Monte Carlo* approach generates a large number of random walks, which can take a long time to fully execute. Therefore execution is parallelized using `java.util.concurrent` for which `Callable` and `Future` lambdas were implemented.
  - **Data Ingress**
    Real-world market data was sourced using the *Yahoo Finance API*, which returns an iterable collection of `BigDecimal`. These daily closing prices are transformed into daily price changes of type `Double[]`. Thus sacrificing accuracy for computational performance.
    A hypothetical portfolio with a number of stocks and options was defined locally.
- **OptionPricer**
  (https://adrian.ng/java/options/) (https://github.com/Adrian-Ng/OptionPricer)
  This project implements three approaches to estimating option prices in Java: *Monte Carlo simulation*, *Black-Scholes equations*, and*Binomial Trees*.
  Option prices for each of these approaches were retrieved through the interface methods `getCall()` and `getPut()`. Abstract classes were also used to facilitate the implementation of various real-world option types, which each differ in how *pay-off* is calculated.
- **Data Mining with Hadoop MapReduce**
  (https://github.com/Adrian-Ng/HadoopEnron)
  A number of *MapReduce* applications were written in Java with a variety of purposes: e.g. extracting the communications network from the *Enron Corpus* or aggregation of Twitter data in *JSON* format.
  These applications extended `Mapper` and `Reduce` classes. Methods `map()` and `reduce()` were overridden with bespoke implementation.
  Java artifacts were exported and ran on Hadoop clusters (both single node and distributed). Input/Output data were stored in HDFS and accessed via `hadoop fs` syntax.
  A subsequent exercise was undertaken to minimise the verbosity of these *Hadoop MapReduce* applications by translating them to *Scala* for use in a *Spark REPL*.
- **Mini Project: Java 8 Streams with financial data**
  (https://adrian.ng/java/yahoofinance/#stream)
  A small exercise involving the use of *Java 8 Streams*. Processing real-world financial data to return *mean* and *equal-weighted variance* of some market asset by defining input-functions to `map()` and `reduce()` methods.

Professional Experience

- **Manchester City Football Club** — London
  Data Analyst — *Jan. - July 2018*
  - **NYCFC Project**
    This data engineering project involved the implementation of SQL *stored procedures* to automate the ingress of data from external sources (transactional data from Ticketmaster, customer data from NYCFC).
    Conference calls with stakeholders (New York City FC) and partners (Major League Soccer) were regularly hosted/attended to understand the *data situation*.
  - **GDPR Customer Preferences**
    This project involved the creation of a number of automated processes to merge GDPR preference data with the analytical database. To accomplish this, SQL DML such as `merge` were utilised.
  - **Tableau Dashboard Automation/Optimisation**
    Implemented *Data Cubes* to pre-aggregate data along all possible subset of categorical fields. That is, every possible drill-down and roll-up was computed in advance. As a result, front-end dashboards retained their exploratory flexibility but removed real-time computational burden. Thus improving user-experience.
  - **Guiding and Mentoring**
    Instructing junior colleagues on SQL Server Management Studio best practices and fundamentals. E.g. understanding DDL & DML for writing SQL queries and creating database objects; when to return a *product join* vs *semi-join*; making use of `information_schema`; utilising *SQL Agent* to schedule jobs. Both regular and ad-hoc workshop sessions were held in order to provide this instruction.
- **ITG Creator** — Westminster
  Senior CRM Campaign Executive — *Dec. 2013 - Sept. 2016*
  - **Virgin Media Processes**
    Built an automated segmentation process using SQL stored procedures for other members of the team to utilise. Recipient data were imported via `BULK INSERT`, stored in database tables and indexed (clustered). Segmentation data was output and linked to HTML content to be broadcast to recipients.
  - **Recursion**
    https://adrian.ng/SQL/cte/Recursion/
    Used recursive queries (CTEs) to clean data e.g. removing n-number of leading zeros from mobile phone numbers in order to prefix with dialling codes; or splitting strings and mapping into relational format.
  - **Query Execution**
    (https://adrian.ng/SQL/misc/openquery-xml)
    Improved cross-server query execution speeds by using `OPENQUERY`, which transmits a SQL query in string format to the remote server. Dynamic SQL was utilised to include XML data in the string. Using a *CTE*, this XML could be represented as relational object. Effectively this allowed us to join and filter on the remote server using data from the local sever. As a result, only a small data set was returned via the `OPENQUERY`.
  - **Soft Skills**
    Attended inter-departmental work assessment groups and advised on work specifications.
    As senior team member, served as point of contact for clients and colleagues looking to resource our team.
    On occasion I held responsibility for resourcing and managing the team's workload in Jira.

- **Languages:** Java 8, T-SQL
- **Software:** IntelliJ IDEA, SQL Server Management Studio, Sublime, Git, Jira, Maven
- **GitHub:** https://github.com/Adrian-Ng
- **Website:** https://adrian.ng