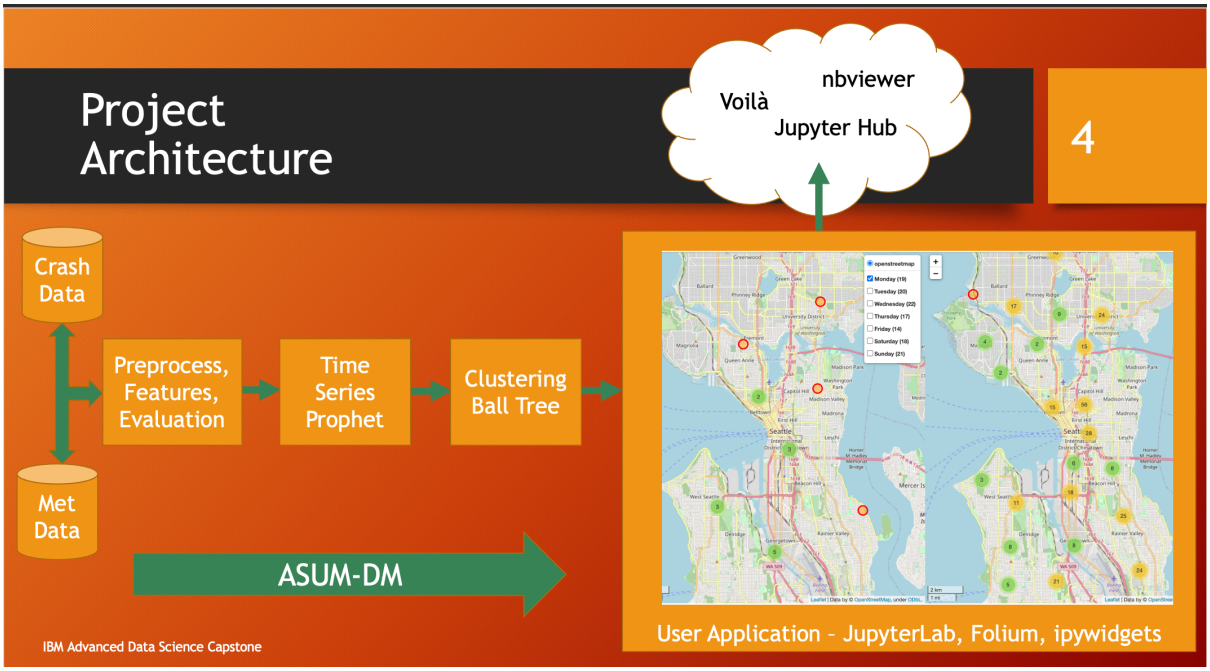
Seattle Collision Analysis Program (SCAP)

Architectural Decisions Document Template

Overview: SCAP is a fictious data product that could be used by Seattle government officials to analyze collision data to gain actionable intelligence based using data science tools and technologies. The data product will provide a map-based interface that will present actual and forecasted collision locations. The application will be available via Jupyter Notebook Viewer.

Note Data Science specific requirements for the Capstone are included at the end of this document starting on page four.

1 Architectural Components Overview



1.1 Data Source

The following data sources will be used by SCAP -

Seattle Department of Transportation (SDOT) collision data hosted by SDOT GIS Division and curated by the SDOT Traffic Division where collisions are collected from Seattle Police Department after a collision is reported.

https://data-seattlecitygis.opendata.arcgis.com/datasets/5b5c745e0f1f48e7a53acec63a0022ab_0?geometry=-123.310%2C47.452%2C-121.352%2C47.776

General characteristics of the subset -
- Format: CSV
- Timeframe: January 2004 to October 2020
- Columns: 38, 37 are unique
- Rows: 194,673
- Bounding Coordinates
-- West Bounding Coordinate: -122.4754
-- East Bounding Coordinate: -122.2008
-- North Bounding Coordinate: 47.7582
-- South Bounding Coordinate: 47.4814

For details on all attributes visit the following link -
https://www.seattle.gov/Documents/Departments/SDOT/GIS/Collisions_OD.pdf

ESRI Metadata -
https://www.arcgis.com/sharing/rest/content/items/5b5c745e0f1f48e7a53acec63a0022ab/info/metadata/metadata.xml?format=default&output=html

1.1.1 Technology Choice

SCAP will be developed using Project Jupyter and Open Source tools and technologies including –

- JupyterLab – application and development environment
- Facebook Prophet – for time series analysis
- Scikit Learn – for clustering
- Jupyter nbviewer – for
- JupyterHub – not for this version but using hub would enable more users to actively engage with SCAP, add to feature and functionality and

1.1.2 Justification

A specific goal of SCAP is to be built on 100% open source tools and technologies rather than COTS tools. While there are many data science tools and technologies available every effort will be made to keep SCAP free of any COTS. This will ensure SCAP has a low TOC and ongoing support from an active community. Jupyter tools and technologies are rapidly evolving and providing end users with tools and technologies that can be easily deployed and managed. While SDOT makes heavy use of ESRI tools an open source platform is cost effective, widely supported by an activity community. SCAP could easily be ported to a web application if the time and resources where available. SCAP will be developed using Python and standards-based technologies that are supported across all popular platforms.

1.2 Enterprise Data

SCAP will not use any enterprise data although SDOT does manage and curate the existing dataset. However it would be extremely advantageous to integrate with Seattle government datasets. This integration could be facilitated via API however the appropriate security controls would need to be applied to integrate with type of data.

1.2.1 Technology Choice

N/A

1.2.2 Justification

The SDOT data set contains a vast amount of information that can be consumed by SCAP. Subsequent versions of SCAP could be integrated with enterprise data at a later time.

1.3 Streaming analytics

SCAP will not use streaming analytics.

1.3.1 Technology Choice

N/A

1.3.2 Justification

N/A

1.4 Data Integration

SCAP will not integrate with any data sources. The SDOT data will be incorporated into the SCAP application.

1.4.1 Technology Choice

N/A

1.4.2 Justification

N/A

1.5 Data Repository

1.5.1 Technology Choice

SCAP will not use a data repository for this iteration. However, it would be beneficial in the future to move to a cloud platform such IBM Cloud to take advantage of data repository offered there.

1.5.2 Justification

N/A

1.6 Discovery and Exploration

1.6.1 Technology Choice

Data exploration was conducted using a Jupyter technologies, specifically JupyterLab. No other tools are required. JupyterLab and IBM Watson Studio as required. Most work can be done on standalone PCs without additional tools and technologies.

1.6.2 Justification

See above.

1.7 Actionable Insights

SCAP will provide analytics to SDOT and Seattle government stakeholders including

1.7.1 Technology Choice

N/A

1.7.2 Justification

SCAP's objective is to provide insights into collision data. Although the data is not frequently updated users will be able to take advantage interface to browse and understand complex collision data.

1.8 Applications / Data Products

SCAP will be built on Jupyter technologies.

1.8.1 Technology Choice

See above.

1.8.2 Justification

SCAP's objective is to offer a low-cost, open source solution for analysis and GIS. While not as full featured as other COTS offerings it provides all of the key functionality needed. However, in the future it would be beneficial to integrate with third-party COTS platform such as ESRI ArcGIS Pro.

1.9 Security, Information Governance and Systems Management

1.9.1 Technology Choice

SCAP is built on publicly available data and will be open source therefore security risks are low. Also, SCAP will be a standalone application therefore no servers are required. It is possible to integrate Jupyter security features such as JupyterLab Credential Store however these are out of scope for this version of SCAP.

1.9.2 Justification

Even though SCAP will be an open source, standalone application any appropriate security protocols and policies will be followed.

1.10 Data Science Specific

1.10.1 Data Quality Assessment

SCAP data quality will be centered around –

- Validating the existing SDOT data set, there are many issues with the data set that need to be addressed during the feature engineering phase. All relevant attributes will be examined during the EDA phase to ensure issues are addressed early on in the process.
- Weather data – the application will use weather data that is known for quality issues. Gaps and issues with the data will be performed to ensure the measurements used are effective. This will be performed during the ETL phase and feature engineering.

1.10.2 Feature Engineering Methods

SCAP will utilize the following methods –

- Numerical and Categorical Imputation
- Categorical data encoding
- Scaling
- Outlier Detection – via visualization, percentiles

1.10.3 Selected Algorithm(s)

SCAP will utilize the following algorithms –

- Ball Tree - a space partitioning data structure for organizing points in a multi-dimensional space. The ball tree gets its name from the fact that it partitions data points into a nested set of hyperspheres known as "balls". The resulting data structure has characteristics that make it useful for a number of applications, most notably nearest neighbor search. Provide by Scikit-Learn.
- Stan - a probabilistic programming language for statistical inference written in C++. The Stan language is used to specify a (Bayesian) statistical model with an imperative program calculating the log probability density function. Provided by Facebook Prophet
- Density-Based Clustering - clusters are defined as areas of higher density than the remainder of the data set. Objects in sparse areas - that are required to separate clusters - are usually considered to be noise and border points. Utilized by DBSCAN
- Haversine Formula - determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles. Distance measure used in clustering, available in Scikit Learn
- Geohash - a public domain geocode system that encodes a geographic location into a short string of letters and digits. It is a hierarchical spatial data structure which

subdivides space into buckets of grid shape, which is one of the many applications of what is known as a Z-order curve, and generally space-filling curves.

- Zero-inflated Poisson (ZIP) - this model mixes two zero generating processes. The first process generates zeros. The second process is governed by a Poisson distribution that generates counts, some of which may be zero. Provided by Statsmodels.

1.10.4 Framework

SCAP will utilize the following frameworks –

- Scikit-Learn - Scikit-Learn is a free ML library and is a Python Machine Learning framework. It is designed to leverage Python's numerical and scientific libraries, namely, NumPy, SciPy, and Matplotlib. It is open-source, reusable and has a broad range of tools and technologies to support data science and machine learning tasks. Specifically, SCAP will use Scikit-Lean clustering libraries
- TensorFlow (and Keras) – future consideration, currently SCAP does not use these frameworks but will be compatible as needed

1.10.5 Model Performance Measures

SCAP will utilize the following performance measures –

- MAE – Mean Absolute Error will be used to evaluate time series models within SCAP
- RSME – in conjunction with MAE, RSME will be used to assess the quality of time series analytics within SCAP
- Silhouette Score - used to understand clustering effectiveness within SCAP
- Custom tool integrated with the SCAP to review cluster and mapping data
- Cross Validation – where applicable to CV will be used to validate the performance of the SCAP analytics