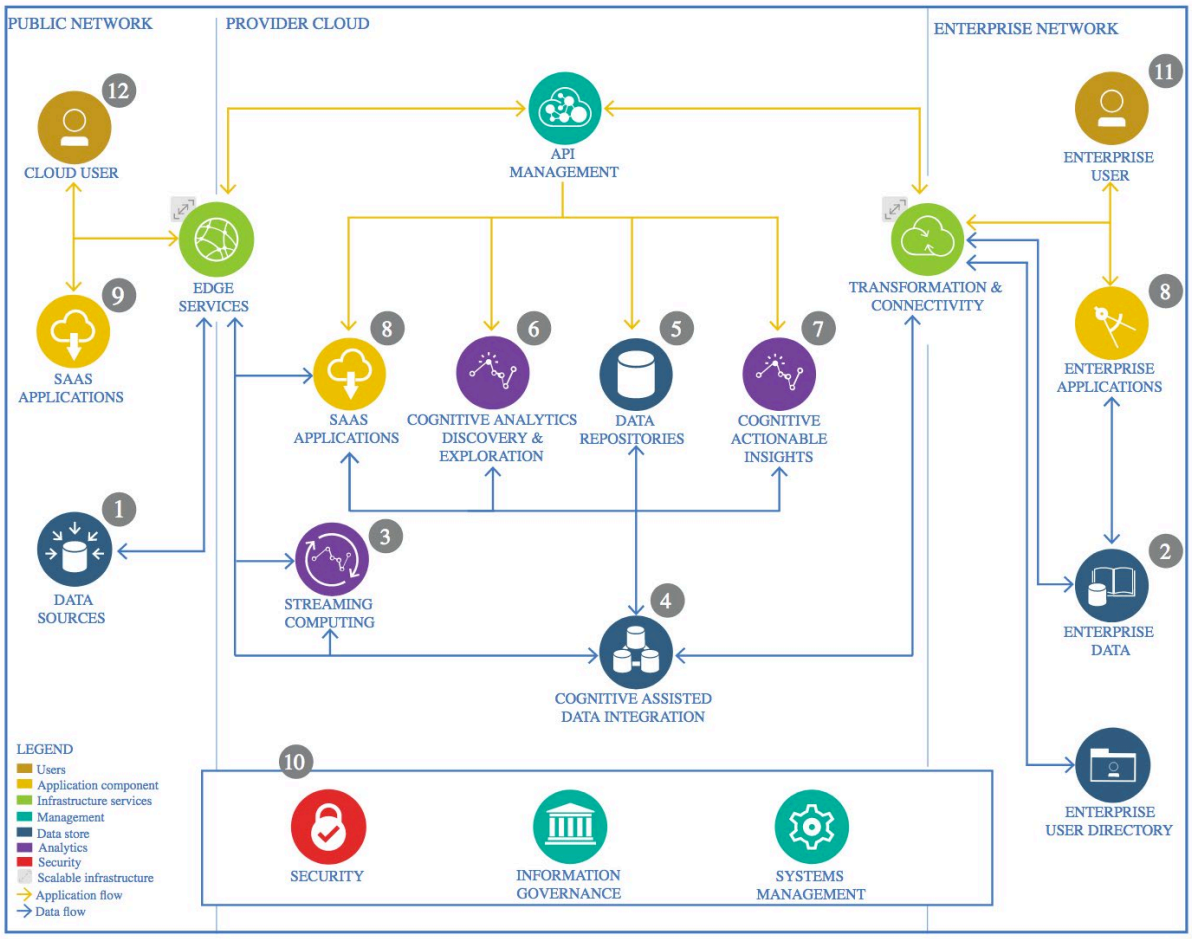
The Lightweight IBM Cloud Garage Method for Data Science

Architectural Decisions Document Template for Prediction of New York City Airbnb prices

1 Architectural Components Overview



IBM Data and Analytics Reference Architecture. Source: IBM Corporation

1.1 Data Source

1.1.1 Technology Choice

The dataset is obtained directly from the Airbnb website (http://data.insideairbnb.com/united-states/ny/new-york-city/2020-06-08/visualisations/listings.csv). Inside Airbnb is an independent, non-commercial organization that provides Airbnb data in cities around the world. Specifically, this dataset describes the listing activity and metrics in New York City as was in 8 June 2020. The columns include the following information: id number of the housing, name of the host, id number of the host, the neighborhood group that the housing is located (Manhattan, Brooklyn, Queens, Staten Island and Bronx), the specific neighborhood, latitude and longitude, type of room, price per night, the minimum nights allowed to book, the number of reviews, the date of last review, average reviews per month, the number of host listings, and the available days in 365 days. The aim of this case study is to identify the factors that affect the price affordability of renting an Airbnb property such as location, room-type and number of reviews. There will be also one additional feature added to the current dataset which is the distance to the nearest train stations.

1.1.2 Justification

Airbnb is a platform where millions of hosts and travelers list their space and book unique accommodations anywhere in the world. Airbnb affects the city's housing supply and affordability. In a modern, busy cosmopolitan like New York City, there are so many competitions for housing and huge demands for accommodations, Airbnb pricing is important to get right. If the price is too high the hosts might risk losing customers while if the price is too low, they might be missing out on potential benefits.

1.2 Enterprise Data

1.2.1 Technology Choice

The dataset used for this project is available for public access. Hence, there will be also one additional feature added to the current dataset which is the distance to the nearest train stations.

1.3 Data Integration

1.3.1 Technology Choice

The data sources are stored in my GitHub repository and imported into Jupyter Notebook.
All the files are in CSV format, hence there is no cloud storage required as the dataset is quite small.

1.3.2 Justification

All the analysis and modeling are done on Jupyter Notebook as well as IBM Watson Studio where the use of Spark is demonstrated for some parts of the analysis.

1.4 Data Repository

1.4.1 Technology Choice

Spark and Pandas data frames are used in importing and accessing the datasets. Hence, there are no database being utilized for this project. The data is stored inside the Github repository and imported into Jupyter Notebook.

1.4.2 Justification

The dataset is collected and analyzed based on the latest monthly update on the Airbnb website.

1.5 Discovery and Exploration

1.5.1 Technology Choice

We explore the data using the essential Python libraries such as Pandas, NumPy, Matplotlib and Seaborn to obtain the descriptive statistics and identify any missing values. After acquiring the dataset from the source, we can explore it to understand the various features present in the dataset and its characteristics. In Python, there is a special package to do this all exploration of data in to a single step known as pandas-profiling. The pandas-profiling Python package is a great tool to create HTML profiling reports. For a given dataset, it computes the following statistics:

- Essentials: type, unique values, missing values
- Quantile statistics like minimum value, Q1, media, Q3, maximum, range, interquartile range.
- Descriptive statistics like mean, mode, standard deviation, sum, median absolute deviation, coefficient of variation, kurtosis, skewness.
- Most frequent values.
- Histogram.
- Correlations highlighting of highly correlated variables, Spearman and Pearson matrices.

As the dataset consist geospatial information i.e. the latitude and longitude of the Airbnb, we also used geopy and ipyleaflet to derive a map of the locations of these Airbnb services.

1.5.2 Justification

These are the libraries commonly used in Python for data manipulation and visualization during the exploratory data analysis phase.

1.6 Actionable Insights

1.6.1 Technology Choice

The dataset illustrates the prices of the Airbnb properties in New York City.
Each `neighborhood group` (borough) has been drilled down into fine locations as `neighborhood` whose count is 222. Among the neighborhood groups:

- `Manhattan` have the least number of unique neighborhood locations (`32`) but at the same time have the highest number of Airbnb listings (`21.963` properties) in the whole New York City, which means that this is a popular site, thus more travelers and tourists and thus the large number of rental places. Therefore, we can expect that the average rental price to be higher than the other boroughs.
- `Brooklyn` have the second highest number of unique neighborhood locations (`48`) and also have the second highest number of Airbnb listings (`19.931` properties) in the whole New York City, which means that this site is also as popular as Manhattan. Therefore, we can expect that the average rental price will be higher than the other boroughs but not as expensive as 'Manhattan` borough is.
- `Queens` have the highest number of unique neighborhood locations (`51`) but a smaller number of Airbnb listings (`6068` properties) than `Manhattan` and `Brooklyn`, which indicates that the Queens is not much popular site therefore less tourists and less rental spaces. Hence, we can expect a low average rental price for this borough.
- `Bronx` have the same number of unique neighborhood locations (`48`) as `Brooklyn` but have few numbers of Airbnb listings (`1,198` properties), which means that this site is less popular. Hence, we can expect a low average rental price for this borough.
- `Staten Island` have the second smallest number of unique neighborhood locations (`43`) but have the least number of Airbnb listings (only `370` properties), which means that this site is not popular at all. Hence, we can expect a low average rental price for this borough.

We can see that the number of Airbnb Listings in each borough is what influences the price and there are more Airbnb listings in Manhattan and Brooklyn.

1.6.2 Justification

The use of data visualization allows important insights to be gathered thus influencing the outcome of the prediction model. By analyzing the price range of different neighborhoods and room types, we can able to identify the properties which are in demand for rental and short-stays.

1.7 Preprocessing and Feature Engineering

1.7.1 Technology Choice

For this dataset, I have identified plenty of outliers for the variables. Hence, one of the preprocessing done was to remove the outliers for the geolocation variables i.e. latitude and longitude. Secondary, I have to filter out the values under the “minimum nights” columns which have over 365 days.

I have also added a new variable to the existing dataset to calculate the distance (in km) of each Airbnb property to the nearest train station. Finally, I applied the using of Label Encoding using scikit-learn preprocessing model to the encode the categorical variables such as.

1.8 Model Architecture

1.8.1 Technology Choice

Using the Python Scikit-Learn libraries, I created eleven regression models to predict the prices of Airbnb rental spaces.

1.8.2 Justification

As the predictor variable consists of continuous values, I gave to the development of the model place a greater emphasis on flexibility rather than interpretability.

1.9 Model Training

1.9.1 Application of regression and learning algorithm

When training a model, the dataset is split into a training and a testing set. The training set is used to train the model whereas the test set is used to access the ability of the system to generalize.

Cross-validation of each model is also performed in predicting the error rate of each learning algorithm. Using 5-fold is used to obtain an optimal error estimate for each regressor. One key advantage of using a cross-validation is to help in minimizing the risk of selecting points near to the hyperplane.

We use the following models:

- ✓ ***Linear Regression:*** To capture the linear relationship between the dependent and the independent variable.
- ✓ ***Ridge Regression:*** Regularized linear model which penalized the higher coefficients by using alpha parameter to eliminate the overfitting issues.
- ✓ ***Lasso Regression:*** Least Absolute Shrinkage Selective Operator which penalizes the higher coefficients by using the parameter alpha to eliminate the overfitting issues and helps the feature selection.
- ✓ ***Decision Tree Regression:*** It uses a decision tree as a predictive model to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It breaks down a dataset into smaller and smaller subset while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes.
- ✓ ***KNN Regression:*** The KNN algorithm uses "features similarity" to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set.
- ✓ ***Random Forest Regression:*** Utilizes bagging (bootstrapping and aggregating) strategy with base eliminator as decision tree and creates a number of trees based on which the model predicts. Here all the trees are independent to each other.
- ✓ ***Gradient Boost Regression:*** Gradient Boosting Regressor in which all the weak learners will be combined to form a strong learner. Here, decision tree is the base estimator and all the estimators are dependent on each other (learns by residuals of the previous decision tree).
- ✓ ***Extreme Gradient Boosting (XG Boost):*** Extreme Gradient Regressor provides an efficient implementation of the Gradient Boosting algorithm. The main benefit of the XGBoost implementation is computational efficiency and often better performance.

- ✓ ***Light Gradient Boosting (Light GBM):*** Light Gradient Boosting Machine is a gradient boosting framework that uses tree based algorithm. It splits the tree leaf wise with the best fit. So, when growing on the same leaf, the lea-wise algorithm can reduce more loss than the level-wise algorithm (like XGBoost) and hence results in much better accuracy.
- ✓ ***Artificial Neural Network (ANN):*** Artificial Neural Network are computational algorithms. It intended to simulate the behavior of biological systems composed of "neurons" and inspired by a central nervous system. It is capable of Machine Learning as well as pattern recognition.
- ✓ ***K-Means Clustering:*** K-Means Clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

1.9.2 Model Improvements

We can improve the model performance by using decreasing the learning rate of the boosting algorithms as well as increase the maximum depth of training data learned. We can also re-train the model using lesser independent variables based on the feature importance.

1.10 Model Evaluation

As this is a mixed regression model, the metrics that were used to evaluate the performance of the model is based on the following scores:

Mean Absolute Error

Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of predictions, without considering their directions. It's the average over the test sample of the absolute difference between prediction and actual observation where all individual differences have equal weight.

Root Mean Square Error

Root Mean Squared Error (RMSE) is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation.

R^2 score

The R^2 score also known as the coefficient of determination measures the amount of variance of the predictive model.

Training Error

Training error is the error that we get when we run the trained model back on the training data. Remember that this data has already been used to train the model and this necessarily doesn't mean that the model once trained will accurately perform when applied back on the training data itself.

Test Error

Test error is the error that we get when we run the trained model on a set of data that it has previously never been exposed to. This data is often used to measure the accuracy of the model before it is shipped to production.