Danny Ki [Data Scientist]

CONTACT

Address
139 Fairway Dr.
Newnan, GA 30265

Mobile
(678)-850-4240

Email
kish1919@gmail.com

Website
datavoyagerdanny.com

GitHub
github.com/kish191919

Linkedin
Linkedin.com/in/danny-sunghwan-ki-52381116

SKILLS

Data Science[Python]
- Numpy/Pandas.
- Sklearn/Statsmodel
- Seaborn/ggplot
- Tensorflow
- Keras

Languages
- Python
- R

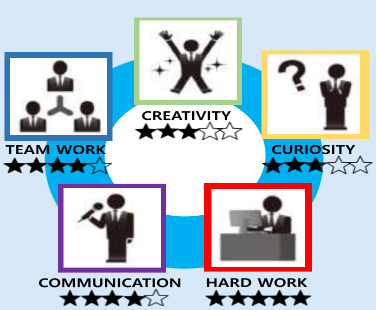
RDBMS and NoSQL
- MySQL
- MongoDB

Big Data
- Hadoop
- Spark
- Hive

Framework
- Flask

Cloud
- AWS

PERSONAL



Overall Highlight

Solid data science background with data engineer skill.
Developed intricate algorithms based on deep-dive statistical analysis and predictive data modeling. Skilled in machine learning, statistics, problem solving and programming.

Relevant Experience

- **Fast Campus** | Data Science Intensive — JAN 2018 - MAY 2018
  - Completed intensive, five-month course on data science
  - Learned industry best practices and practical data science standards by collaborating with senior data scientists

- **Udacity** | Intro to Programming Nanodegree — JUN 2017 - AUG 2017
  - Project-based programming credential
  - Learned the basics of programming through Python and Data analysis

Projects

Portfolio Website: http://datavoyagerdanny.com

- **Predict Used-Car Price in Georgia** [Service Website : http://dannyki.ga/]
  - Crawled on cars.com to collect data and store it in AWS's mysql
  - After preprocessing stored data, put it in machine learning model to predict used car price
  - Models learned on AWS server implemented as web services using Flask web framework

- **[Kaggle] Predict House Prices** | OLS Regression | Rank: 1042 / 4548 (22.9%)
  - Developed and applied OLS algorithms to predict house prices in Ames, Iowa
  - Built a linear regression model with R-square of 0.945 and a model with most significant variables of house size and quality using python

- **[Kaggle] Spooky Author Identification** | Naive Bayes | Rank: 793 / 1,244 (63.7%)
  - As a text analysis project, it was a problem seeing which authors wrote articles. After vectorizing words, classified them via machine learning with Naive Bayes Classification.

- **[Kaggle] Titanic Machine Learning** | Voting Classifier | Rank: 4,304 / 10,676 (40.3%)
  - Predicted survival on the Titanic through combining several classification algorithms

- **[Kaggle] Bike Sharing Demand** | Random Forest | Rank: 1,357 / 3,251 (41.7%)
  - Predicted bicycle demand using R language

- **[Kaggle] Digit Recognizer** | Keras Sequential Model | Rank : 1,139 / 2,502 (45.5%)
  - Identified digits from a dataset of tens of thousands of handwritten images using python

Certifications

- **Fast Campus**
  - Machine learning with R
  - Apache Hadoop

- **Coursera (Professor)**
  - Machine learning *(Andrew Ng)*
  - Machine learning foundation *(Carlos, Emily)*
  - Introduction to probability and data *(Mine)*
  - Python programming *(Charles Severance)*

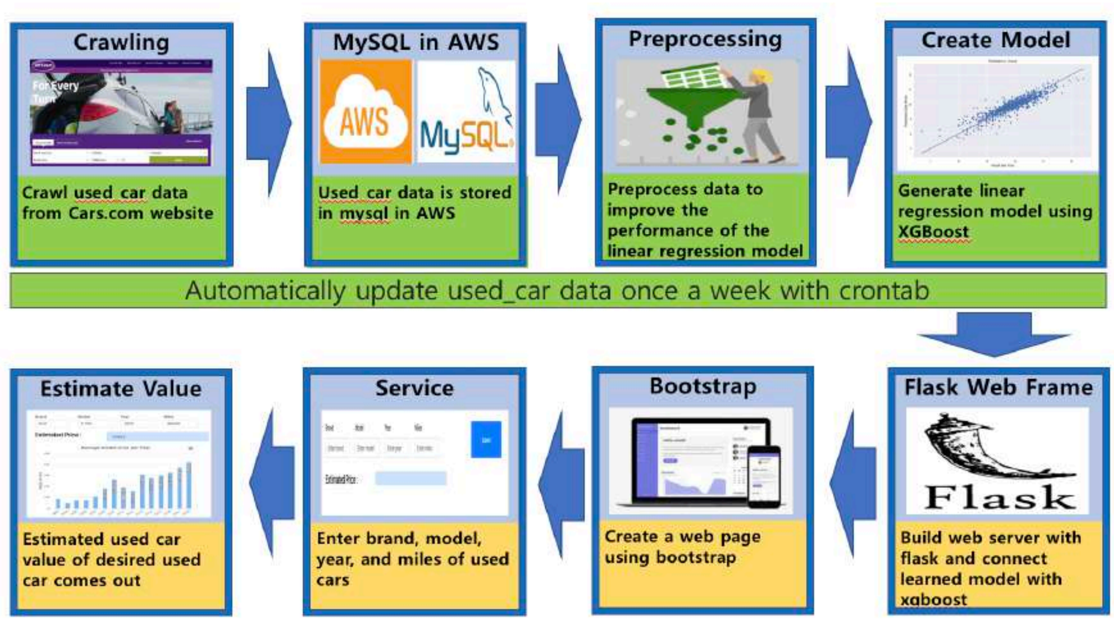
Additional Experience & Education

- **Dongwon Autopart** | Sales and Logistics Assistant Manager — AUG 2015 - DEC 2017
- **Kukdo Chemical Co.,LTD** | Purchasing Assistant Manager — DEC 2013 - JUL 2015
- **Lotte Chemical Alabama Corp** | Purchasing Assistant Manager — OCT 2011 - NOV 2013
- **Myoungji University** — MAR 2000 - FEB 2009
  - Business Administration and International Business [GPA: 3.62 / 4.0]

PROJECT

(1) Predict Used-Car Price in Georgia

Subject : Machine learning based vehicle forecasting program
Period : 2018. 03 - 2018. 04
Tech : Python (Pandas, Scikit-learn), Data Crawl , AWS, Flask, MySQL, Bootstrap
Model : XGBooster (Accuracy : 88%)
Structure :



Service Website : http://dannyki.ga/
How to use the Service Website :

→ Fill in the information and then press the submit button.

→ You can check the price of the used car you want, and you can also check the average price for different years with the same model.

Comment : If I use many variables to increase the accuracy of the expected car value, it is very inconvenient for the user to enter all the variable information into the web service page(http://dannyki.ga/). So, only the four most influential variables (Brand, Model, Year and Miles) are used, and when you enter these variables, you get the expected car value.

PROJECT

(2) [KAGGLE Competition] Predict House Prices

Subject : Predict house prices in Ames, Lowa
Period : 2018. 01 - 2018. 03
Data : Train Data - 81 variables and 1,460 house data
Test Data - 80 variables and 1,459 house data
Python : Preprocessing - Numpy, Pandas
Graph - Matplotlib, Seaborn
Model : **OLS (Ordinary Least Squares) Model**

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.945
Model:                            OLS   Adj. R-squared:                  0.942
Method:                 Least Squares   F-statistic:                     403.6
Date:                Mon, 26 Mar 2018   Prob (F-statistic):               0.00
Time:                        21:13:50   Log-Likelihood:                 1405.0
No. Observations:                1383   AIC:                            -2696.
Df Residuals:                    1326   BIC:                            -2398.
Df Model:                          56
Covariance Type:            nonrobust
===============================================================================================
                                coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
Intercept                    12.1018      0.059    204.753      0.000      11.986      12.218
C(Neighborhood)[T.Blueste]   -0.0265      0.069     -0.381      0.703      -0.163       0.110
C(Neighborhood)[T.BrDale]    -0.0495      0.037     -1.347      0.178      -0.122       0.023
C(Neighborhood)[T.BrkSide]   -0.0058      0.031     -0.184      0.854      -0.067       0.056
C(Neighborhood)[T.ClearCr]   -0.0454      0.033     -1.383      0.167      -0.110       0.019
```

Insight : Among the 79 house value related variables, it is best to predict the house value by using 18 numeric variables (GrLivArea, OverallQual and so on) and 5 category variables (Neighborhood, KitchenQual and so on). R-squared was the highest with 0.942 and the kaggle score was 0.12384.

Kaggle Score : 0.12384 / Kaggle rank : 1,042 / 4,548 (22.9%)
Github : https://github.com/kish191919/House_Price_Project_by_Python

(3) [KAGGLE Competition] Spooky Author Identification

Subject : Identify an author from sentences which they wrote
Period : 2018. 03 - 2018. 04
Data :Train Data - 3 variables and 19,579 text data
Test Data - 2 variables and 8,392 text data
Python : Natural Language Processing - Stopword, Stemming
Vectorization - CountVectorizer
Model - Randomforest, AdaBoost, SVM, Naive Bayes Classification
Model : **Naive Bayes Classification**

```
Confusion Matrix :
 [[7414  110  376]
 [ 631 4764  240]
 [ 588   89 5367]]

10-fold Cross Validation Report:
              precision    recall  f1-score   support

          0       0.86      0.94      0.90      7900
          1       0.96      0.85      0.90      5635
          2       0.90      0.89      0.89      6044

avg / total       0.90      0.90      0.90     19579
```

Insight : The performance of Precision and Recall was different according to the method of text processing and machine learning algorithm. Among them, the Naive Bayes Classification distinguished the author well, and the precision and recall were high.

Kaggle Score : 0.48767 / Kaggle rank : 793 / 1,244 (63.7%)
Github : https://github.com/kish191919/Spooky_Author_Identification_by_Python

PROJECT

(4) [KAGGLE Competition] Titanic Machine Learning from Disaster

Subject : Predict survival on the Titanic
Period : 2018. 03 - 2018. 04
Data : Train Data - 12 variables and 891 data
Test Data - 11 variables and 418 data
Python : Preprocessing - Numpy, Pandas
Graph - Matplotlib, Seaborn
Models - DecisionTree, Randomforest, Adaboost, Support Vector Machine,
Naive Bayes Classfication, VotingClassifier
Model : **VotingClassifier**

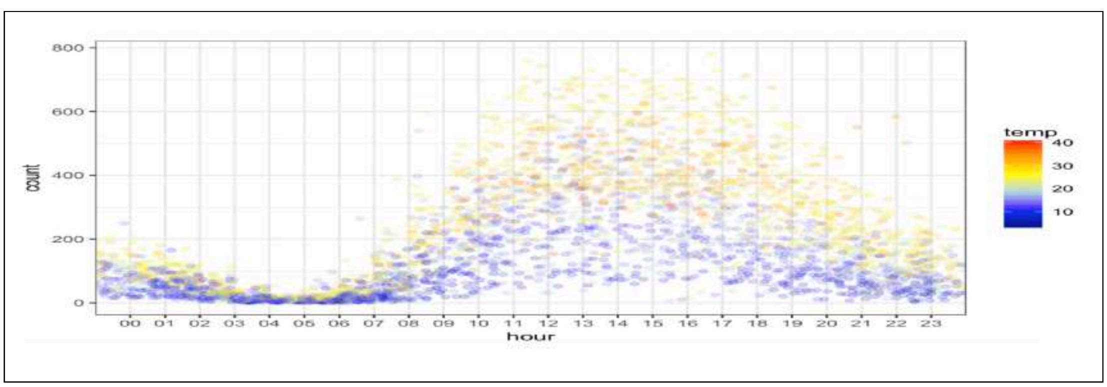
```
Confusion Matrix :
 [[484  57]
 [ 80 260]]

10-fold Cross Validation Report:
             precision    recall  f1-score   support

          0       0.86      0.89      0.88       541
          1       0.82      0.76      0.79       340

avg / total       0.84      0.84      0.84       881
```

Insight : The survival rate was higher when the female was in the 1st class, the ages were in the 20s to 50s, and the family size was 1 or 2.

Kaggle Score : 0.78468 / Kaggle rank : 4,304 / 10,676 (40.3%)
Github : https://github.com/kish191919/Titanic_Machine_Learning_from_Disaster_by_Python

(5) [KAGGLE Competition] Bike Sharing Demand

Subject : Predict demand on bike
Period : 2018. 04
Data :Train Data - 12 variables and 10,886 data
Test Data - 9 variables and 6,493 data
R : Preprocessing - dplyr
Graph - ggplot
Model - Randomforest
Model : **Randomforest**



Insight : The temperature and the demand for bicycles have a correlation with each other. The higher the temperature, the higher the bicycle, especially after lunch and before dinner.

Kaggle Score : 0.48613 / Kaggle rank : 1,357 / 3,251 (41.7%)
Github : https://github.com/kish191919/Bike-Sharing-Demand_by_R

PROJECT

(6) [KAGGLE Competition] Digit Recognizer By Deep Learning

Subject : Identify digits from a dataset of tens of thousands of handwritten images
Period : 2018. 04
Data : Train Data - 42,000 data (Each image is 28 pixels x 28 pixels)
Test Data - 28,000 data (Each image is 28 pixels x 28 pixels)
Python : Preprocessing - Numpy, Pandas
Graph - Matplotlib, Seaborn
Neural Network
Model : **Keras Sequential Model**

Model Layer: 2 layers [Each layer has 4 filter and filter size (5X5) and Relu activation function]

```
model.summary()

_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_7 (Conv2D)            (None, 28, 28, 4)         104
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 4)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 14, 14, 4)         0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 10, 10, 4)         404
_________________________________________________________________
flatten_3 (Flatten)          (None, 400)               0
_________________________________________________________________
dense_3 (Dense)              (None, 10)                4010
=================================================================
Total params: 4,518
Trainable params: 4,518
Non-trainable params: 0
_________________________________________________________________
```

Model Training: Epoch [150], Loss [0.0325], Accuracy [0.9893]

```
Epoch 148/150
 - 8s - loss: 0.0347 - acc: 0.9885
Epoch 149/150
 - 8s - loss: 0.0341 - acc: 0.9888
Epoch 150/150
 - 8s - loss: 0.0325 - acc: 0.9893
CPU times: user 1h 24min 21s, sys: 5min 34s, total: 1h 29min 56s
Wall time: 20min 18s
```

Insight : The model took about 20 mins to run all epoch, and it achieved the accuracy rate of 0.9893.
I compared the performance by changing parameters such as increasing the layer further or controlling the dropout rate.
Setting a deeper layer does not improve performance, but rather makes it worse, and the performance of the model currently configured with two layers has been the best so far.

Kaggle Score : 0.98271 / Kaggle rank : 1,139 / 2,502 (45.5%)
Github : https://github.com/kish191919/Digit_Recognizer_by_Deep_Learning