Stroke Prediction

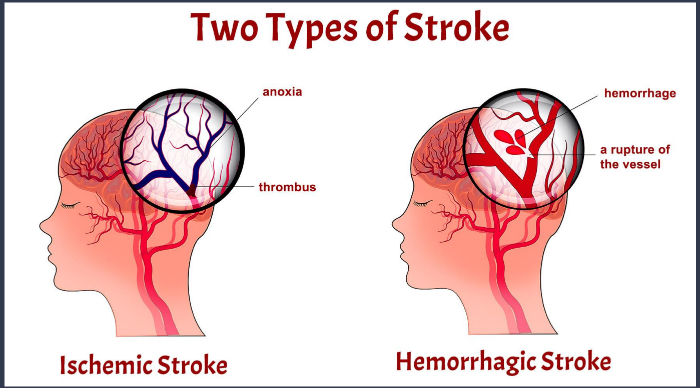
Group 10
Jared Clarke & Amr Salem

Why strokes?

- Strokes are the 2nd leading cause of death in developed countries after heart disease (WHO)
- The #5 cause of death in the US
- One of the leading causes of disability
- Sometimes can be prevented
- Could be of use medically

American Stroke Association.
A division of the
American Heart Association.

What is a stroke?



- Blood vessel that supplies nutrients to the brain is hindered by a clot (Ischemic -- 87 % of strokes)
- Blood vessel bursts (Hemorrhagic)
- Brain tissues can't receive necessary nutrients
- Leads to brain cells dying in only a few minutes
- Difficulty walking, speaking and understanding, paralysis of the face and or extremities
- Deadly if not treated immediately.

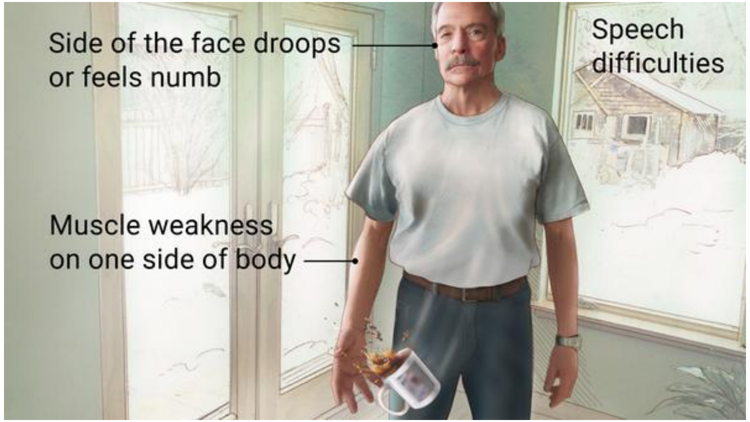
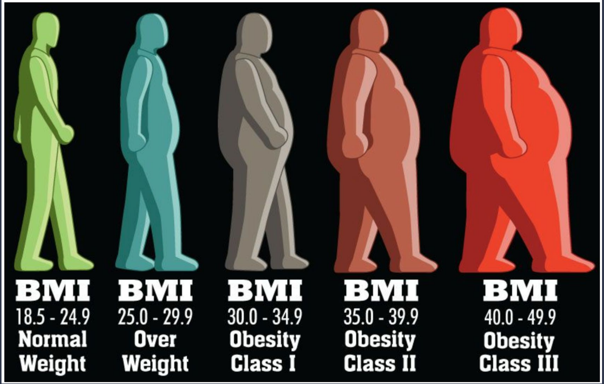


Image source

Image source

Stroke Risk Factors



- Age -- Especially for people > 65
- Gender -- women > men
- Hypertension
- High Cholesterol
- Smoking
- Diabetes
- Obesity - high BMI (over 30)
- Stress (work, marriage, location of residence)

Data Sources:

Kaggle

Analytics Vidhya

McKinsey Analytics Online Hackathon - Healthcare Analytics

Question we hope to answer:

Can we reliably predict a stroke based on certain features of a person’s medical history?

Features

- Gender (Male, Female, Unknown)
- Age
- Hypertension (high blood pressure)
- Heart Disease (yes or no)
- Has ever been married (yes or no)
- Work type (Private, self-employed, government job, etc.)
- Residence type (Urban or Rural)
- Avg glucose level
- BMI (>30 considered high risk)
- Smoking status
- Has patient ever experienced a stroke

Processing the data

```python
# Import data from PostgreSQL database
db_string = f"postgres://postgres:{db_password}@127.0.0.1:5432/strokes_db"
engine = create_engine(db_string)
stroke_df = pd.read_sql_table('total_stroke_data', con=engine)

stroke_df.head()
```

	id	gender	age	hypertension	heart_disease	ever_married	work_type	residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	18069	Male	70.0	1	0	Yes	Self-employed	Urban	104.24	34.7	formerly smoked	0
1	49086	Female	23.0	0	0	No	Private	Urban	60.50	27.1	formerly smoked	0
2	19671	Female	58.0	0	0	Yes	Govt_job	Urban	93.15	34.7	never smoked	0
3	59225	Male	48.0	1	0	Yes	Govt_job	Urban	55.25	49.7	never smoked	0
4	25175	Female	56.0	0	0	No	Private	Rural	108.50	28.0	never smoked	0

```python
stroke_df.shape
```

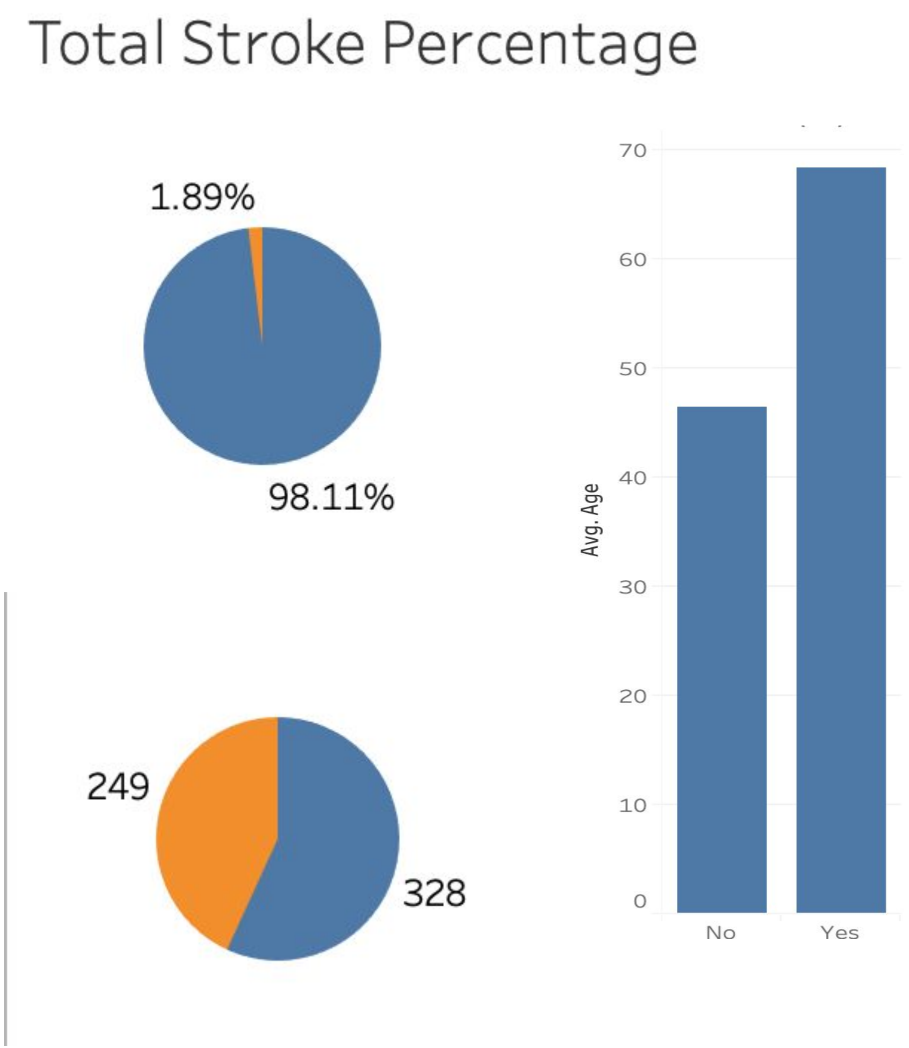
```
(30555, 12)
```

```python
stroke_df['ever_married'] = stroke_df['ever_married'].apply(lambda x: 1 if x == 'Yes' else 0)
stroke_df['residence_type'] = stroke_df['residence_type'].apply(lambda x: 1 if x == 'Urban' else 0)

# Encoding the gender column
gender_num = []
for i in stroke_df['gender']:
    if i == 'Male':
        gender_num.append(0)
    if i == 'Female':
        gender_num.append(1)

stroke_df['gender'] = gender_num
```

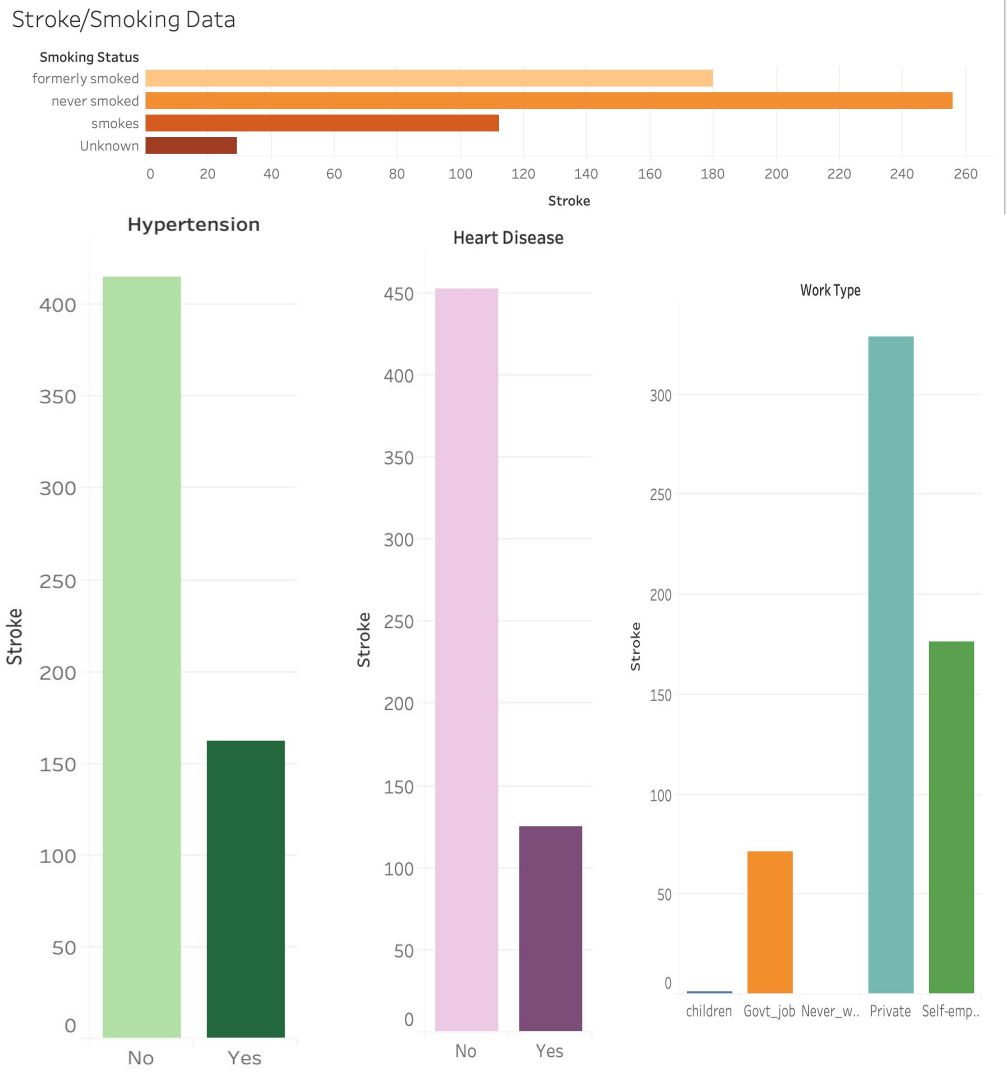
```python
# Encoding the 'work_type' column
label_encoder = LabelEncoder()
label_encoder.fit(stroke_df['work_type'])
stroke_df['work_type_le'] = label_encoder.transform(stroke_df['work_type'])

work_type_num = {'Private': 0,
                 'Self-employed': 1,
                 'Govt_job': 2,
                 'children': 3,
                 'Never_worked': 4}
stroke_df['work_type_num'] = stroke_df['work_type'].apply(lambda x: work_type_num[x])
stroke_df.drop(columns=['work_type', 'work_type_le'], inplace=True)
```

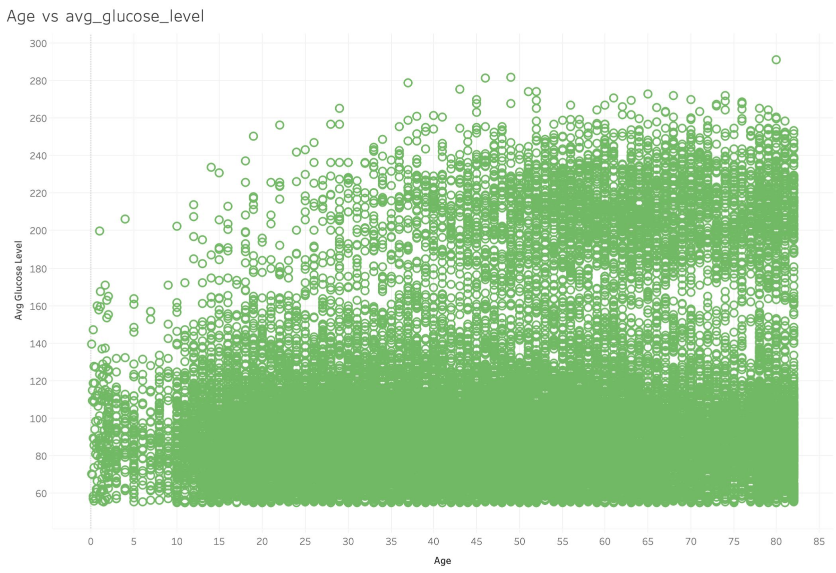
Exploring the data:

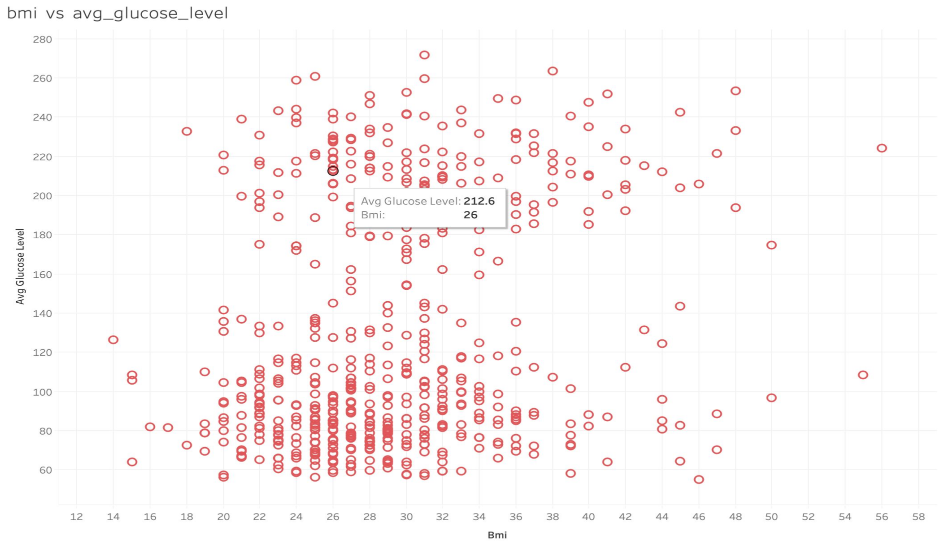
- Nearly 2% of people in the dataset had a stroke
- Slight gender imbalance toward women
- Average age is 68.35 years

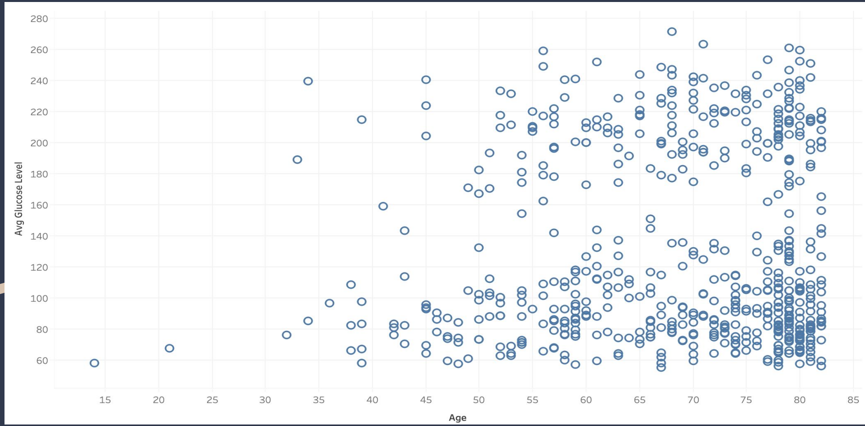


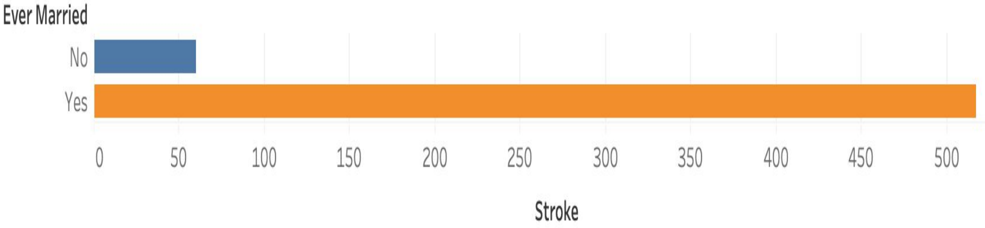
- Higher number of strokes for former smokers compared to current smokers (180 : 112)
- 28% or 162 people had hypertension
- 22% or 125 people had heart disease
- 329 (57%) Private company employers and 176 self-employed (31%) had strokes



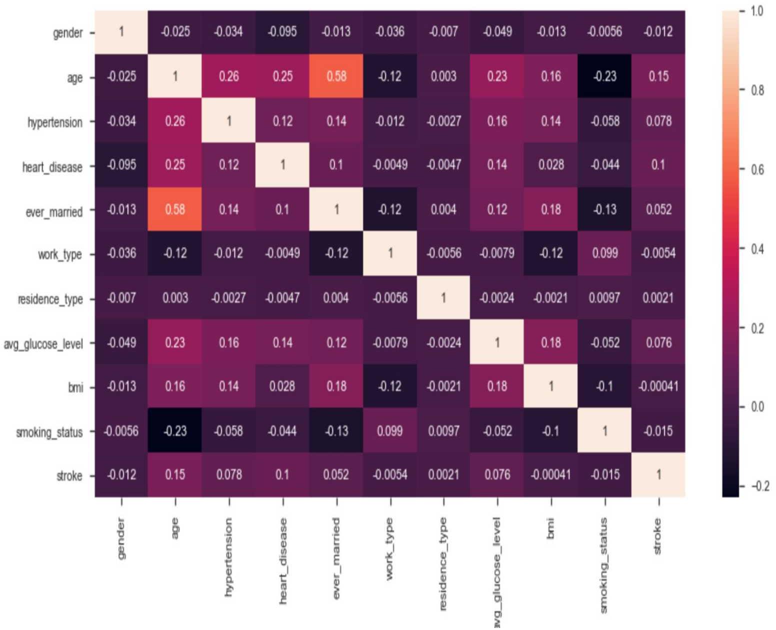

Age vs avg_glucose_level
Avg Glucose Level
300 280 260 240 220 200 180 160 140 120 100 80 60
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85
Age


bmi vs avg_glucose_level
Avg Glucose Level
280 260 240 220 200 180 160 140 120 100 80 60
Avg Glucose Level: 212.6
Bmi: 26
12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58
Bmi


Avg Glucose Level
280 260 240 220 200 180 160 140 120 100 80 60
15 20 25 30 35 40 45 50 55 60 65 70 75 80 85
Age


Ever Married
No
Yes
0 50 100 150 200 250 300 350 400 450 500
Stroke

Correlation




```
Residuals:
     Min       1Q   Median       3Q      Max
-0.13081 -0.03178 -0.01290  0.00119  1.01200

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -2.610e-02  4.376e-03  -5.964 2.48e-09 ***
gender             9.339e-06  1.580e-03   0.006   0.9953
age                1.094e-03  5.217e-05  20.973  < 2e-16 ***
hypertension       1.543e-02  2.588e-03   5.961 2.53e-09 ***
heart_disease      3.911e-02  3.636e-03  10.756  < 2e-16 ***
ever_married      -1.424e-02  2.136e-03  -6.669 2.61e-11 ***
work_type          4.570e-04  8.680e-04   0.526   0.5986
residence_type     5.570e-04  1.532e-03   0.364   0.7161
avg_glucose_level  1.142e-04  1.789e-05   6.383 1.76e-10 ***
bmi               -5.194e-04  1.093e-04  -4.752 2.02e-06 ***
smoking_status     3.156e-03  1.000e-03   3.156   0.0016 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1339 on 30537 degrees of freedom
Multiple R-squared:  0.03353,	Adjusted R-squared:  0.03322
F-statistic:   106 on 10 and 30537 DF,  p-value: < 2.2e-16
```

Analysis

- Random Forest Classifier
- Logistic Regression

Random Forest Classifier

```
forest = RandomForestClassifier(n_estimators = 100)
forest.fit(X_train, y_train)

forest_score = forest.score(X_train, y_train)
forest_test = forest.score(X_test, y_test)

y_pred = forest.predict(X_test)

print('Training Score',forest_score)
print('Testing Score \n',forest_test)
print(cm)
print(classification_report(y_test, y_pred))
```

```
Training Score 0.9993777777777778
Testing Score
 0.8173333333333334
[[5848    0]
 [1652    0]]
              precision    recall  f1-score   support

           0       0.84      0.94      0.89      5848
           1       0.65      0.37      0.47      1652

    accuracy                           0.82      7500
   macro avg       0.75      0.66      0.68      7500
weighted avg       0.80      0.82      0.80      7500
```

Logistic Regression

```
model = LogisticRegression(solver="lbfgs",max_iter=200)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print('Testing Score \n',score)

print(classification_report(y_test, y_pred))

cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
Testing Score
 0.7797333333333333
              precision    recall  f1-score   support

           0       0.78      1.00      0.88      5848
           1       0.00      0.00      0.00      1652

    accuracy                           0.78      7500
   macro avg       0.39      0.50      0.44      7500
weighted avg       0.61      0.78      0.68      7500

[[5848    0]
 [1652    0]]
```

Neural Network

```
239/239 - 1s - loss: 0.0930 - accuracy: 0.9811
Loss: 0.09303482621908188, Accuracy: 0.9811444282531738
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 80)                880
_________________________________________________________________
dense_1 (Dense)              (None, 30)                2430
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 31
=================================================================
Total params: 3,341
Trainable params: 3,341
Non-trainable params: 0
_________________________________________________________________
```

Stroke Risks You Can Control:

- High Blood Pressure
- Smoking
- Diabetes
- Diet
- Physical Inactivity
- Cholesterol
- Obesity

Unique Symptoms in Women:

- Loss of consciousness or fainting
- General weakness
- Difficulty or shortness of breath
- Confusion, unresponsiveness or disorientation
- Sudden behavioral change
- Agitation
- Hallucination
- Nausea or vomiting
- Pain
- Seizures
- Hiccups

Takeaways

- Too much data for Logistic Regression
- Unbalance data set -- 1.9% had strokes
- Leave out features that are weak according to the heatmap and test
- Leave out features that are weak according to multiple linear regression and test
- Want precision more than accuracy

Analysis Tools Used

- GitHub
- Python
- Jupyter Notebook
- PostgreSQL
- Scikit learn library
- Keras library
- Tensorflow
- Tableau
- SqlAlchemy
- Visual Studio Code