Machine Learning Case Study

Name: **Rudy Martinez**

abc123: **Lpe538**

Blank notebook to be used for class exercises.

Real Example using Week 8's Material

In [77]:
```
import csv
import numpy as np
from sklearn.feature_extraction import DictVectorizer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score
from sklearn.model_selection import train_test_split, GridSearchCV
import random
```

Read the data and convert data to matrix

In [78]:
```
X_dicts = [] #Features
Y = [] #Class

with open('churn.csv') as file:
    csv = csv.reader(file, delimiter = ',')

    header = next(csv)

    for row in csv:
        features = {header[3]: float(row[3]),
                    header[4]: row[4],
                    header[5]: row[5],
                    header[6]: float(row[6]),
                    header[7]: float(row[7]),
                    header[8]: float(row[8]),
                    header[9]: float(row[9]),
                    header[10]: float(row[10]),
                    header[11]: float(row[11]),
```

```
                    header[12]: float(row[12])}

        X_dicts.append(features)
        Y.append(int(row[-1]))

vec = DictVectorizer(sparse = False)
X = vec.fit_transform(X_dicts)
y = np.array(Y)
```

Explore Data

In [79]:
```
print(f"Shape: {X.shape}\n")
print(f"Feature Names: {vec.feature_names_}\n")

stats = X.mean(axis = 0)

for f, x in zip(vec.feature_names_, stats):
    print(f, x)

print()
print(y.mean())
```

```
Shape: (10000, 13)

Feature Names: ['Age', 'Balance', 'CreditScore', 'EstimatedSalary', 'Gender=Female', 'Gender=Male', 'Geography=
France', 'Geography=Germany', 'Geography=Spain', 'HasCrCard', 'IsActiveMember', 'NumOfProducts', 'Tenure']

Age 38.9218
Balance 76485.88928799961
CreditScore 650.5288
EstimatedSalary 100090.2398809998
Gender=Female 0.4543
Gender=Male 0.5457
Geography=France 0.5014
Geography=Germany 0.2509
Geography=Spain 0.2477
HasCrCard 0.7055
IsActiveMember 0.5151
NumOfProducts 1.5302
Tenure 5.0128

0.2037
```

Train model on Data (GridSearchCV with LinearSVC)

In [100…
```
np.random.seed(42)
random.seed(42)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

parameters_1 = {'C':[0.001, 0.01, 0.1, 1.]}

svc_1 = LinearSVC()
clf_1 = GridSearchCV(svc_1, parameters_1, scoring = 'f1', cv = 2)

clf_1.fit(X_train, y_train)
```

```
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
/opt/anaconda3/lib/python3.8/site-packages/sklearn/svm/_base.py:976: ConvergenceWarning: Liblinear failed to co
nverge, increase the number of iterations.
  warnings.warn("Liblinear failed to converge, increase "
```

Out[100…

```
GridSearchCV(cv=2, estimator=LinearSVC(),
             param_grid={'C': [0.001, 0.01, 0.1, 1.0]}, scoring='f1')
```

Evaluate GridSearchCV with LinearSVC

In [105…

```
print(f"Best Score: {clf_1.best_score_}\n")
print(f"Best Params: {clf_1.best_params_}\n")

preds_1 = clf_1.predict(X_test)
print(f"Evaluation: {f1_score(y_test, preds_1)}")
```

```
Best Score: 0.35301370643722546

Best Params: {'C': 1.0}

Evaluation: 0.35147928994082844
```

Train model on Data (GridSearchCV with RandomForestClassifier)

In [102…
```
parameters_2 = {'n_estimators': [10, 100, 200, 300, 400]}

svc_2 = RandomForestClassifier()
clf_2 = GridSearchCV(svc_2, parameters_2, scoring = 'f1', cv = 2)

clf_2.fit(X_train, y_train)
```

Out[102…
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'n_estimators': [10, 100, 200, 300, 400]},
             scoring='f1')
```

Evaluate GridSearchCV with RandomForestClassifier model

In [104…
```
print(f"Best Score: {clf_2.best_score_}\n")
print(f"Best Params: {clf_2.best_params_}\n")

preds_2 = clf_2.predict(X_test)
print(f"Evaluation: {f1_score(y_test, preds_2)}")
```

```
Best Score: 0.5701264802131772

Best Params: {'n_estimators': 400}

Evaluation: 0.5852895148669796
```

F1 Score

- The model trained with RandomForestClassifier has an f1 score of **0.5852895148669796**
- The model trained with LinearSVC has an f1 score of **0.35147928994082844**

Conclusions

- The RandomForestClassifier model performed better

In [ ]: