Machine Learning

In order to run different machine learning models, I had to convert the income bracket from ' <=50K' and ' >50K' to '0' and '1'. This had to be done in order to use binary classification. I used standard scaler to scale the data. For the categorical variables I turned them into dummy variables. When splitting the train and test subsets I used a test size of 20%. For the classification models I used decision tree model, logistic regression model, random forest classifier model and SVC model. From the following models SVC had the best accuracy. I used GridSearchCV to tune the SVC Model.

Decision Tree Classifier

```
# Decision Tree Model
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import classification_report

clf_dtc = DecisionTreeClassifier()
clf_dtc = clf_dtc.fit(X_train,   y_train)

y_pred = clf_dtc.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8085529587270015
              precision    recall  f1-score   support

           0       0.84      0.92      0.88      4503
           1       0.67      0.48      0.56      1530

    accuracy                           0.81      6033
   macro avg       0.75      0.70      0.72      6033
weighted avg       0.80      0.81      0.80      6033
```

Logistic Regression

```
# Logistic Regression Model
from sklearn.linear_model import LogisticRegression
clf_lrm = LogisticRegression()
clf_lrm = clf_lrm.fit(X_train, y_train)

y_pred = clf_lrm.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8102105088678933
              precision    recall  f1-score   support

           0       0.82      0.95      0.88      4503
           1       0.73      0.40      0.52      1530

    accuracy                           0.81      6033
   macro avg       0.78      0.68      0.70      6033
weighted avg       0.80      0.81      0.79      6033
```

Random Forest Classifier

```
# Random Forest Classifier Model
from sklearn.ensemble import RandomForestClassifier
clf_rfc = RandomForestClassifier()
clf_rfc = clf_rfc.fit(X_train, y_train)

y_pred = clf_rfc.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8143543842201226
              precision    recall  f1-score   support

           0       0.84      0.92      0.88      4503
           1       0.68      0.50      0.58      1530

    accuracy                           0.81      6033
   macro avg       0.76      0.71      0.73      6033
weighted avg       0.80      0.81      0.80      6033
```

SVC

```
# SVC Model
from sklearn.svm import SVC
clf_svc = SVC()
clf_svc = clf_svc.fit(X_train, y_train)

y_pred = clf_svc.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
/Users/grannelpinto/anaconda3/lib/python3.7/site-packages/sklearn/svm/base.py:193: FutureWarning: The default value o
f gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explici
tly to 'auto' or 'scale' to avoid this warning.
  "avoid this warning.", FutureWarning)
```

```
Accuracy: 0.8219791148682247
              precision    recall  f1-score   support

           0       0.83      0.97      0.89      4503
           1       0.80      0.40      0.53      1530

    accuracy                           0.82      6033
   macro avg       0.81      0.68      0.71      6033
weighted avg       0.82      0.82      0.80      6033
```

Model Tuning

```
# Tune the SVC Model using GridSearchCV
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [0.1, 10, 100],
              'gamma': [1, 0.01, 0.001],
              }

CV_svc = GridSearchCV(estimator=clf_svc, param_grid=param_grid, cv= 5)
CV_svc.fit(X_train, y_train)
```

```
GridSearchCV(cv=5, error_score='raise-deprecating',
             estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
                           decision_function_shape='ovr', degree=3,
                           gamma='auto_deprecated', kernel='rbf', max_iter=-1,
                           probability=False, random_state=None, shrinking=True,
                           tol=0.001, verbose=False),
             iid='warn', n_jobs=None,
             param_grid={'C': [0.1, 10, 100], 'gamma': [1, 0.01, 0.001]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=0)
```

```
y_pred = CV_svc.predict(X_test)
print("Accuracy:", metrics.accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.8252942151500083
              precision    recall  f1-score   support

           0       0.84      0.95      0.89      4503
           1       0.76      0.45      0.57      1530

    accuracy                           0.83      6033
   macro avg       0.80      0.70      0.73      6033
weighted avg       0.82      0.83      0.81      6033
```