Dataset A + SVM

CISC873 Hannah LeBlanc

Data Cleaning

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
customerID          7043 non-null object
gender              7043 non-null object
SeniorCitizen       7043 non-null int64
Partner             7043 non-null object
Dependents          7043 non-null object
tenure              7043 non-null int64
PhoneService        7043 non-null object
MultipleLines       7043 non-null object
InternetService     7043 non-null object
OnlineSecurity      7043 non-null object
OnlineBackup        7043 non-null object
DeviceProtection    7043 non-null object
TechSupport         7043 non-null object
StreamingTV         7043 non-null object
StreamingMovies     7043 non-null object
Contract            7043 non-null object
PaperlessBilling    7043 non-null object
PaymentMethod       7043 non-null object
MonthlyCharges      7043 non-null float64
TotalCharges        7043 non-null object
Churn               7043 non-null object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
Month-to-month	Yes	Electronic check	29.85	29.85	No
One year	No	Mailed check	56.95	1889.5	No
Month-to-month	Yes	Mailed check	53.85	108.15	Yes
One year	No	Bank transfer (automatic)	42.30	1840.75	No
Month-to-month	Yes	Electronic check	70.70	151.65	Yes

MonthlyCharges	TotalCharges	Churn
52.55	NaN	No
20.25	NaN	No
80.85	NaN	No
25.75	NaN	No
56.05	NaN	No
19.85	NaN	No
25.35	NaN	No
20.00	NaN	No
19.70	NaN	No
73.35	NaN	No
61.90	NaN	No

	tenure	MonthlyCharges	TotalCharges	Contract
0	1	29.85	29.85	Month-to-month
1	34	56.95	1889.50	One year
2	2	53.85	108.15	Month-to-month
3	45	42.30	1840.75	One year
4	2	70.70	151.65	Month-to-month

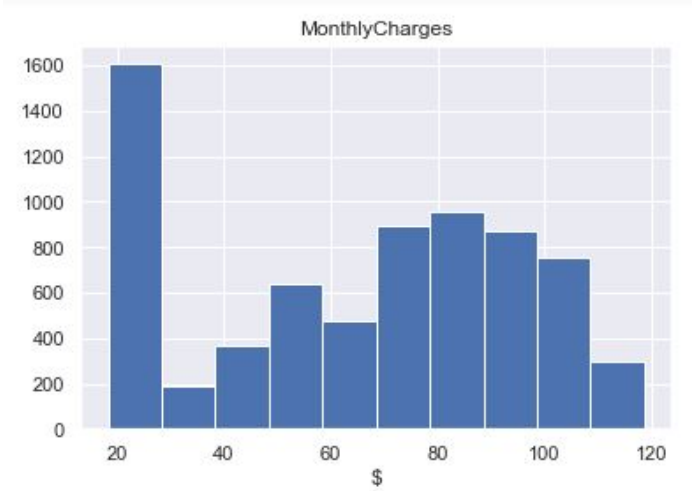
```
df[df.isnull().any(axis=1)]["tenure"]
```

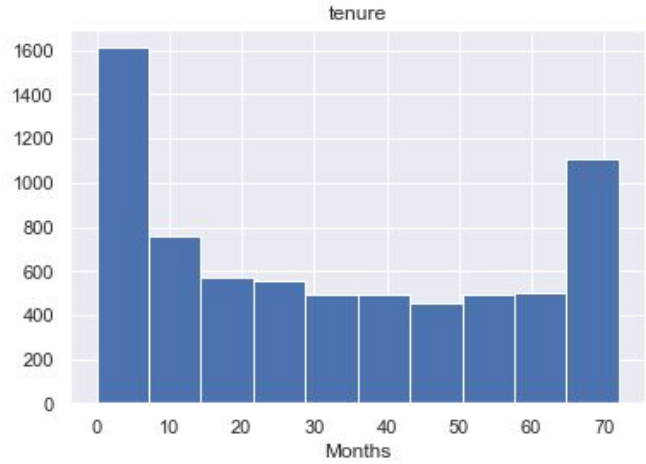
```
488     0
753     0
936     0
1082    0
1340    0
3331    0
3826    0
4380    0
5218    0
6670    0
6754    0
Name: tenure, dtype: int64
```

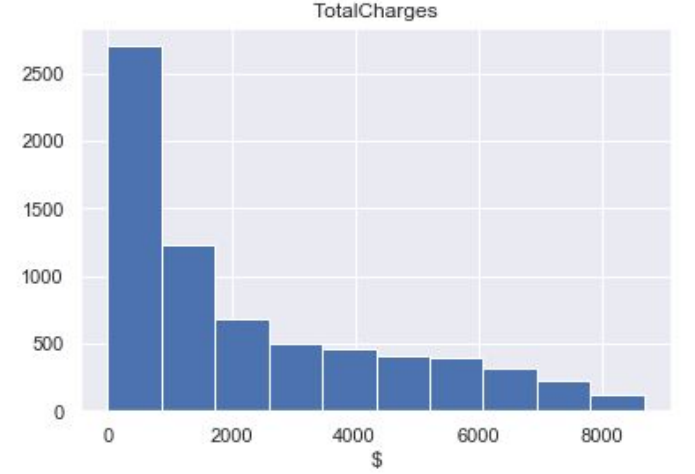
Column Encodings

```
Gender:  ['Female' 'Male']
Partner:  ['Yes' 'No']
Dependents:  ['No' 'Yes']
PhoneService:  ['No' 'Yes']
MultipleLines:  ['No phone service' 'No' 'Yes']
InternetService:  ['DSL' 'Fiber optic' 'No']
OnlineSecurity:  ['No' 'Yes' 'No internet service']
OnlineBackup:  ['Yes' 'No' 'No internet service']
DeviceProtection:  ['No' 'Yes' 'No internet service']
TechSupport:  ['No' 'Yes' 'No internet service']
StreamingTV:  ['No' 'Yes' 'No internet service']
StreamingMovies:  ['No' 'Yes' 'No internet service']
Contract:  ['Month-to-month' 'One year' 'Two year']
PaperlessBilling:  ['Yes' 'No']
PaymentMethod:  ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
 'Credit card (automatic)']
Churn:  ['No' 'Yes']
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 43 columns):
```



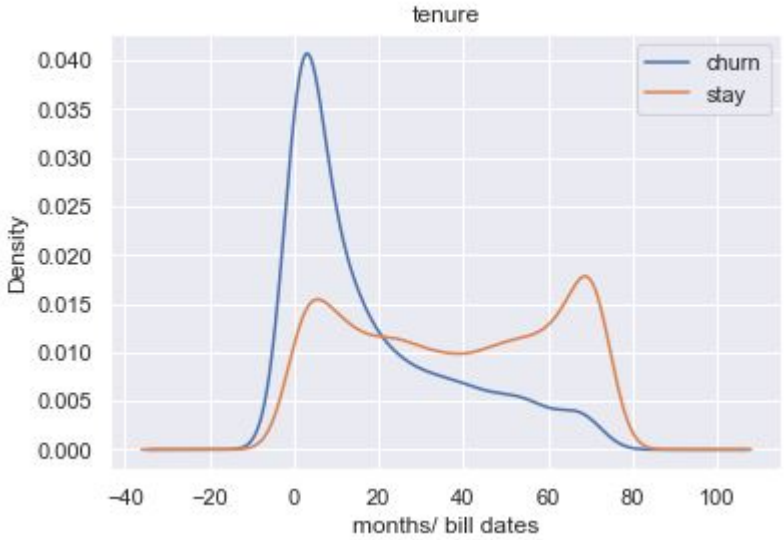


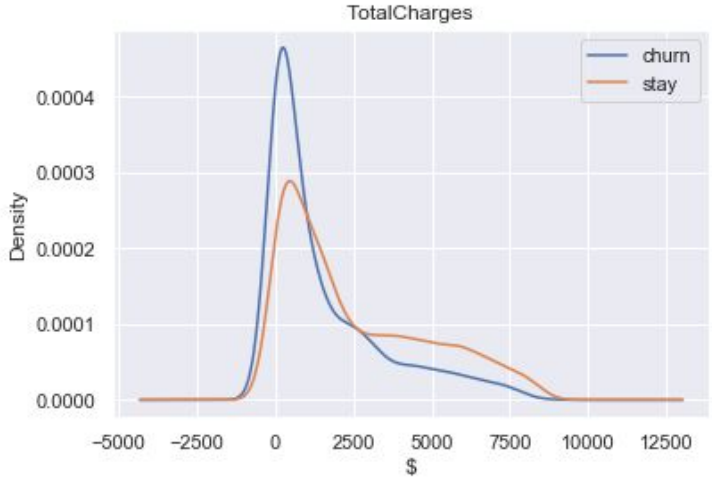


	tenure	MonthlyCharges	TotalCharges
count	7043.000000	7043.000000	7043.000000
mean	32.371149	64.761692	2279.734304
std	24.559481	30.090047	2266.794470
min	0.000000	18.250000	0.000000
25%	9.000000	35.500000	398.550000
50%	29.000000	70.350000	1394.550000
75%	55.000000	89.850000	3786.600000
max	72.000000	118.750000	8684.800000

```
print("Churned: ", len(df[df["Churn"] == 1]))
print("Stayed: ", len(df[df["Churn"] == 0]))
```

```
Churned:  1869
Stayed:  5174
```





Baseline

```
0.7702683515547352

[[4850 1294]
 [ 324  575]]
```

	precision	recall	f1-score	support
0	0.79	0.94	0.86	5174
1	0.64	0.31	0.42	1869
avg / total	0.75	0.77	0.74	7043

Scaling

```
scaler = StandardScaler()
X = scaler.fit_transform(X)
baseline_model_scale = SVC()
y_pred = cross_val_predict(baseline_model_scale, X, y, cv=5)
```

```
0.7986653414738037

[[4699  943]
 [ 475  926]]

             precision    recall  f1-score   support

          0       0.83      0.91      0.87      5174
          1       0.66      0.50      0.57      1869

avg / total       0.79      0.80      0.79      7043
```

Different Kernels & Parameters

```
Cs = [0.001, 0.01, 0.1, 1, 10]
gammas = [0.001, 0.01, 0.1, 1]
degrees = [1,2,3,4]
```

Kernel	Hyper parameters	Accuracy	Confusion Matrix	f1
Linear	{'C': 0.01, 'gamma': 0.001}	0.8003	[[4632 864] [542 1005]]	precision recall f1-score support 0 0.84 0.90 0.87 5174 1 0.65 0.54 0.59 1869 avg / total 0.79 0.80 0.79 7043
RBF (default)	{'C': 10, 'gamma': 0.001}	0.8025	[[4675 892] [499 977]]	precision recall f1-score support 0 0.84 0.90 0.87 5174 1 0.66 0.52 0.58 1869 avg / total 0.79 0.80 0.79 7043
Polynomial	{'degree': 2,'C': 10}	0.7980	[[4669 918] [505 951]]	precision recall f1-score support 0 0.84 0.90 0.87 5174 1 0.65 0.51 0.57 1869 avg / total 0.79 0.80 0.79 7043

Feature Selection

(1) TotalCharges ≠ tenure*MonthlyCharges

(2)

```
# Create correlation matrix
corr_matrix = df.corr().abs()

# Select upper triangle of correlation matrix
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.bool))

# Find index of feature columns with correlation greater than 0.95
to_drop = [column for column in upper.columns if any(upper[column] > 0.95)]
```

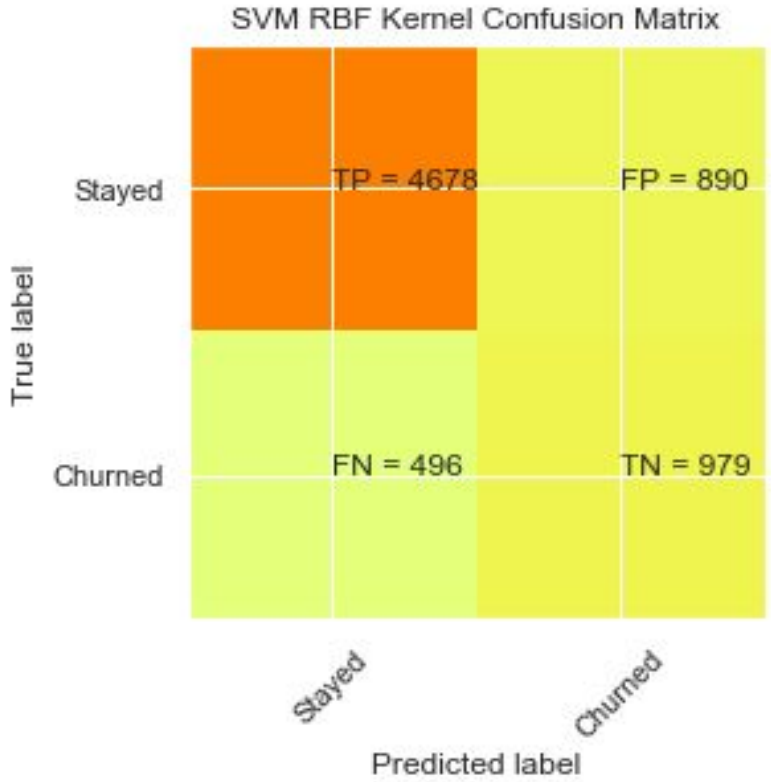
```
to_drop
```

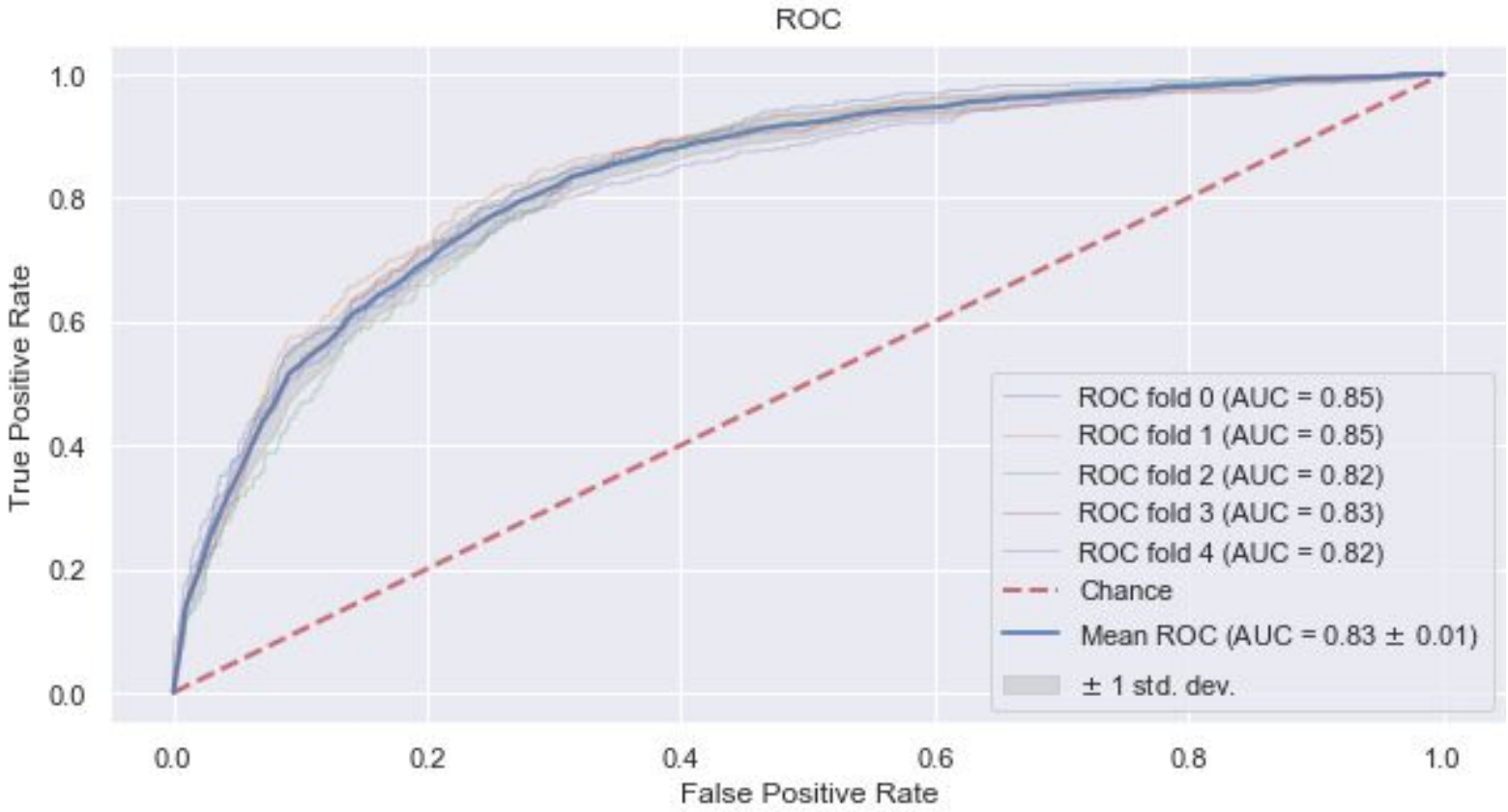
```
['PhoneService_Yes',
 'MultipleLines_No phone service',
 'OnlineSecurity_No internet service',
 'OnlineBackup_No internet service',
 'DeviceProtection_No internet service',
 'TechSupport_No internet service',
 'StreamingTV_No internet service',
 'StreamingMovies_No internet service']
```

Final SVM

Accuracy: 0.8032088598608548

	precision	recall	f1-score	support
0	0.84	0.90	0.87	5174
1	0.66	0.52	0.59	1869
avg / total	0.79	0.80	0.80	7043




ROC
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC fold 0 (AUC = 0.85)
ROC fold 1 (AUC = 0.85)
ROC fold 2 (AUC = 0.82)
ROC fold 3 (AUC = 0.83)
ROC fold 4 (AUC = 0.82)
Chance
Mean ROC (AUC = 0.83 ± 0.01)
± 1 std. dev.