Predicting Telecom Churn

Rock Shi

Problem

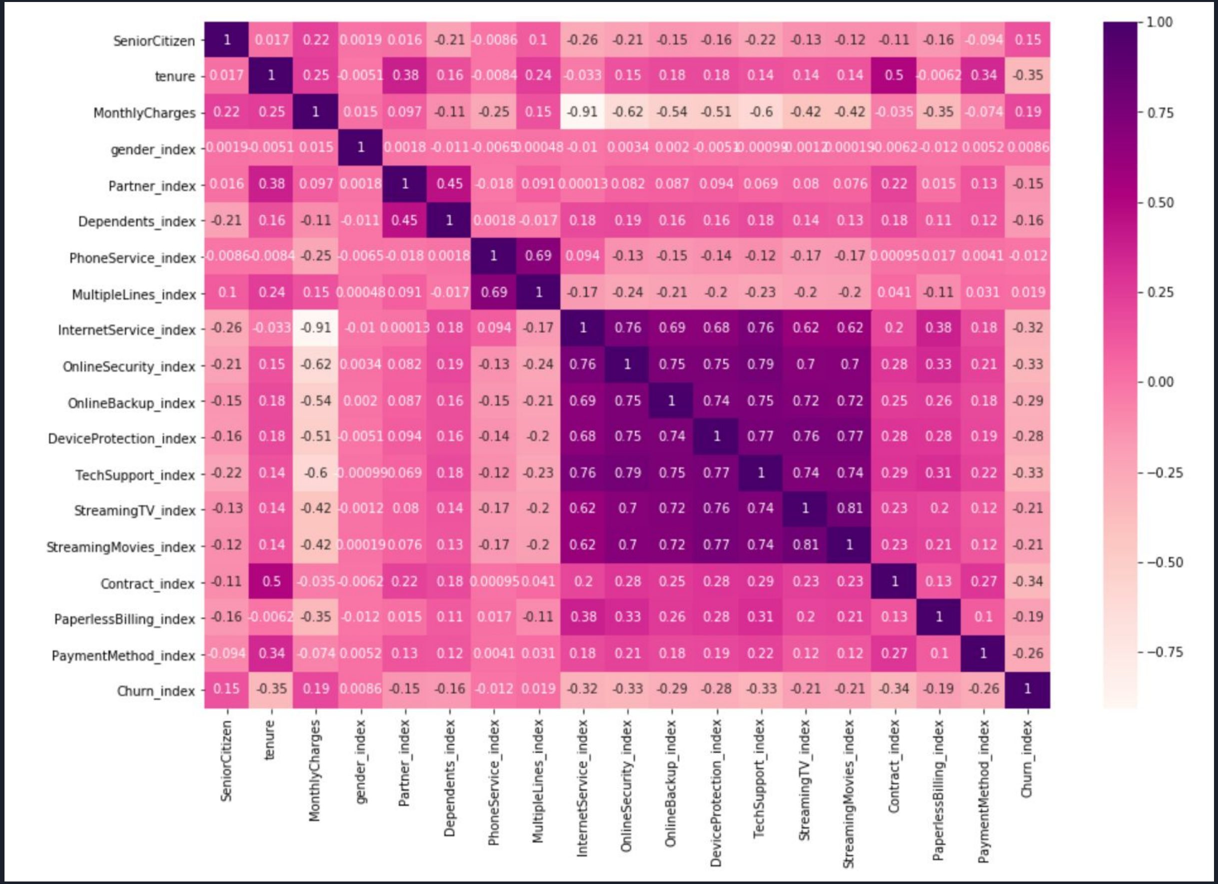
Dailytalk Telecom needs analysis of their data to predict churn of their customers so that they can effectively target those customers with customized offers

Why is customer retention important?
(aka Why is this project important?)

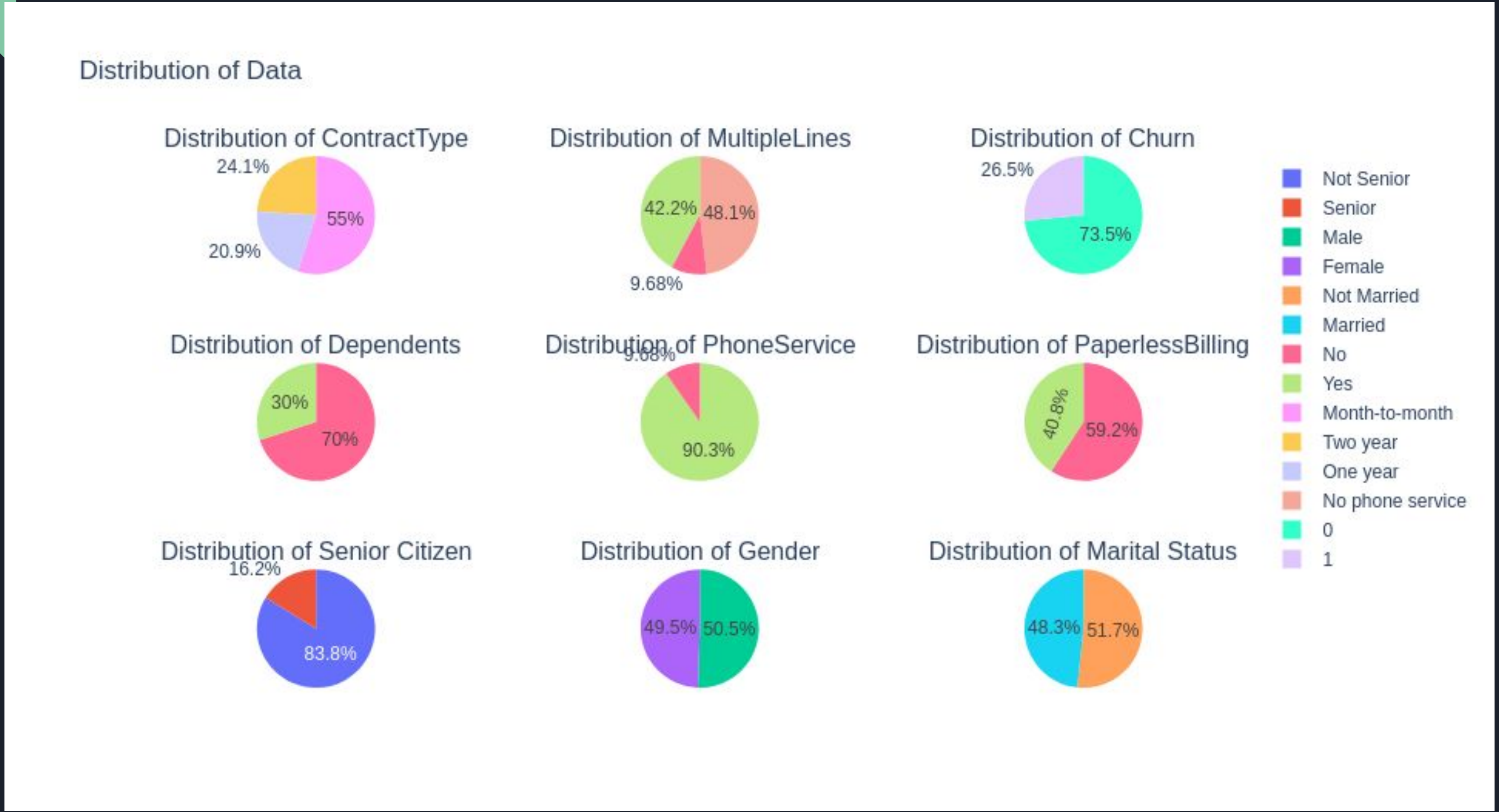
There are many reasons but here are a few…

- ➔ Loyal customers are more profitable
- ➔ Loyal customers are more forgiving
- ➔ Loyal customers will welcome your marketing
- ➔ Loyal customers will provide more helpful feedback
- ➔ The company will earn more customer to customer referrals
- ➔ You become a place where customers trust you with money because you give them good value in exchange

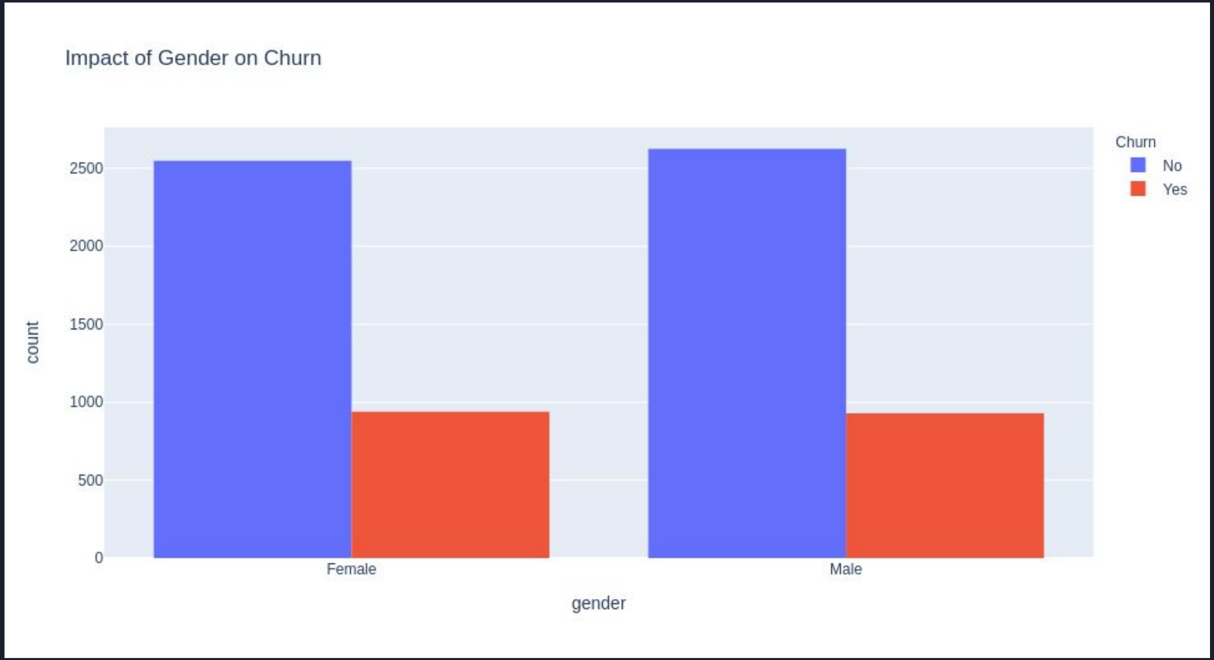
Feature Correlation Heatmap



Distribution of Important Features



Gender Vs Churn



Relatively even distribution of both genders as well as churn.

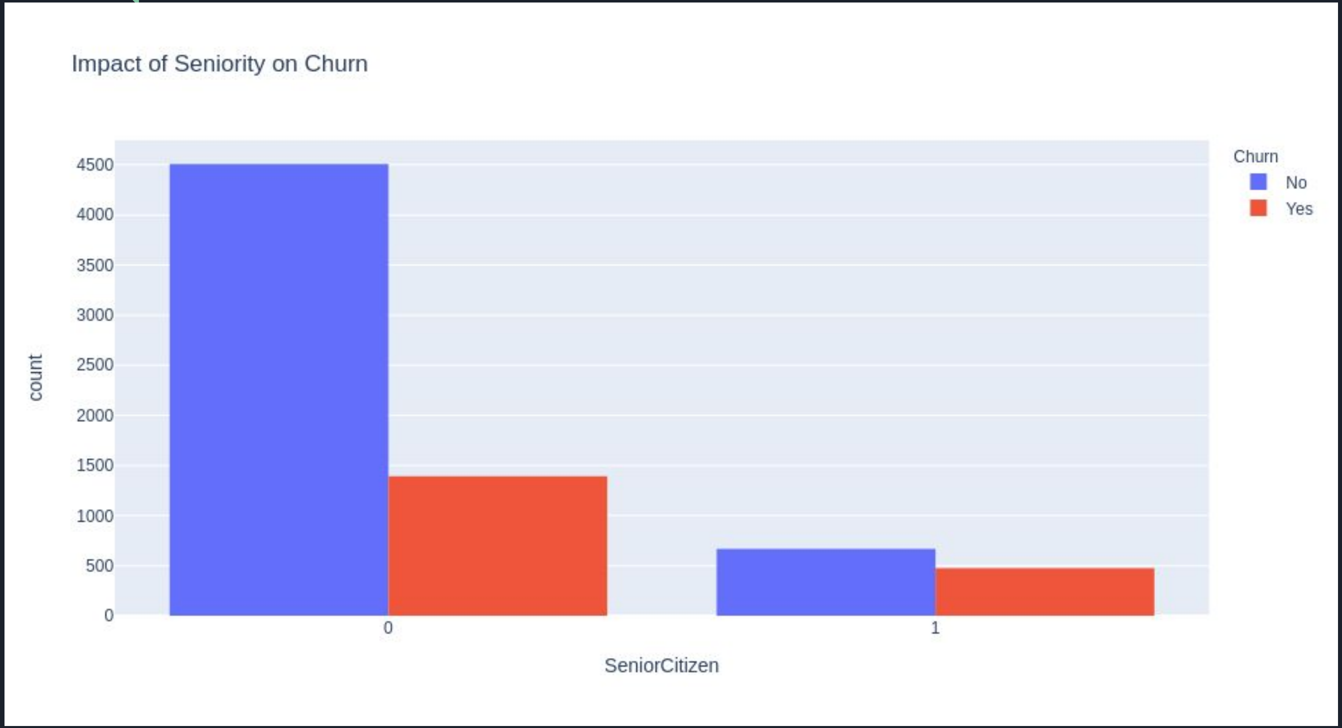
➔ Gender has no effect on Churn

Gender vs Churn cont.

```
# testing gender impact on churn
male_df = churn_pd_df[churn_pd_df.gender == 'Male']
female_df = churn_pd_df[churn_pd_df.gender == 'Female']

statistics, p_value = stats.ttest_ind(male_df['Churn'], female_df['Churn'])

print('Statistics:', statistics)
print('p_value:', p_value)


if p_value <= 0.05:
    print("Gender has an impact on Churn.")
else:
    print("Gender has no impact on Churn.")
```

```
Statistics: -0.722673440663404
p_value: 0.46990453909804797
Gender has no impact on Churn.
```

Seniority vs Churn



Significantly less seniors in the data but the proportion of people leaving is higher in seniors

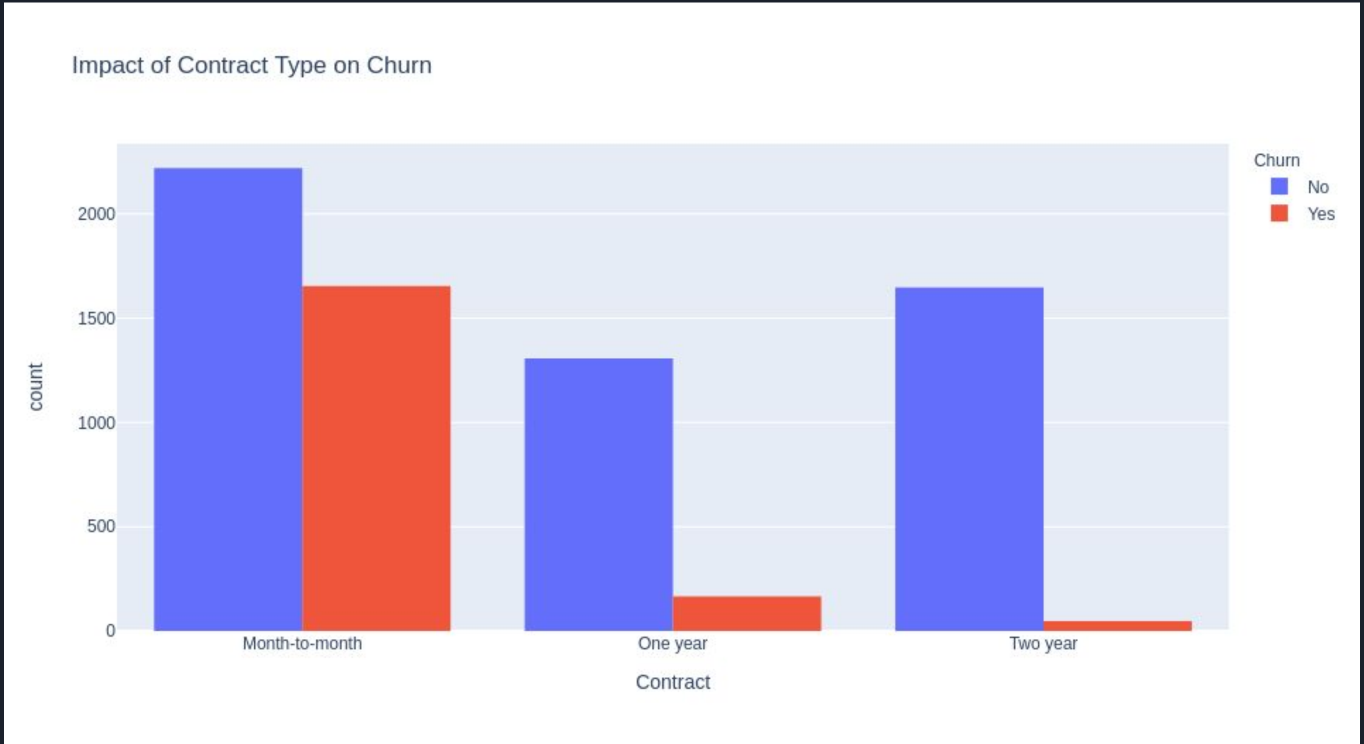
➔ Seniority has an effect on Churn

Seniority vs Churn cont

```
# testing if being a senior citizen impacts churn
senior_df = churn_pd_df[churn_pd_df.SeniorCitizen == 'Senior']
not_senior_df = churn_pd_df[churn_pd_df.SeniorCitizen == 'Not Senior']

statistics, p_value = stats.ttest_ind(senior_df['Churn'], not_senior_df['Churn'])

print('Statistics:', statistics)
print('p_value:', p_value)


if p_value <= 0.05:
    print("Seniority has an impact on Churn.")
else:
    print("Seniority has no impact on Churn.")
```

```
Statistics: 12.807865726034748
p_value: 3.839860055784895e-37
Seniority has an impact on Churn.
```

Contract Type vs Churn



Most people are in month to month contracts and have highest chance of leaving.

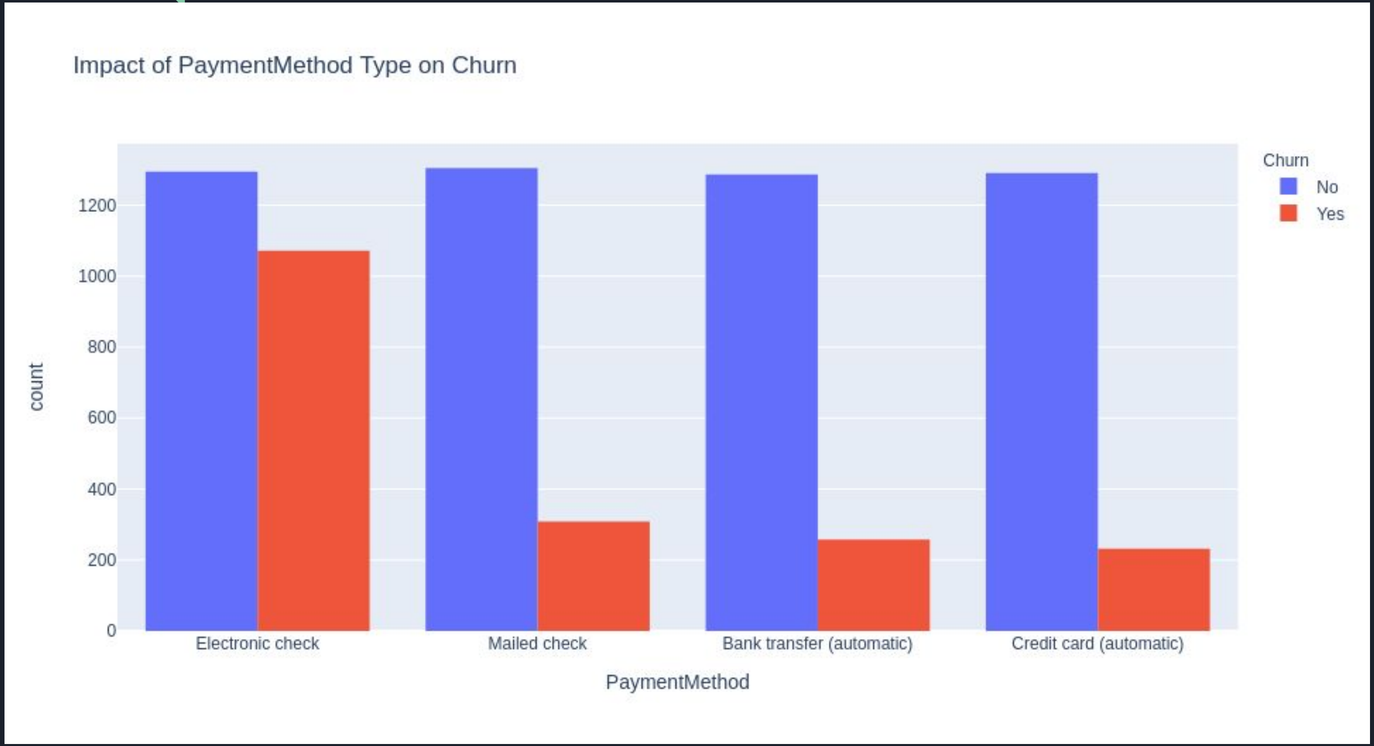
➔ Contract Type has effect on Churn

Contract Type vs Churn cont

```
# Testing if contract type will have impact on churn (1-way Anova)
mtm_df = churn_pd_df[churn_pd_df.Contract == 'Month-to-month']
oy_df = churn_pd_df[churn_pd_df.Contract == 'One year']
ty_df = churn_pd_df[churn_pd_df.Contract == 'Two year']

f_stat, p_value = stats.f_oneway(mtm_df['Churn'],oy_df['Churn'],ty_df['Churn'])

print('f_stat:', f_stat)
print('p_value:', p_value)

if p_value <= 0.05:
    print("Contract types have an effect on Churn.")
else:
    print("Contract types do not have an effect on Churn.")
```

```
f_stat: 711.7604625631104
p_value: 2.9921663019905003e-282
Contract types have an effect on Churn.
```

Payment Method vs Churn



Relatively equal distribution of people using each payment type with e-check payers leaving noticeably more.

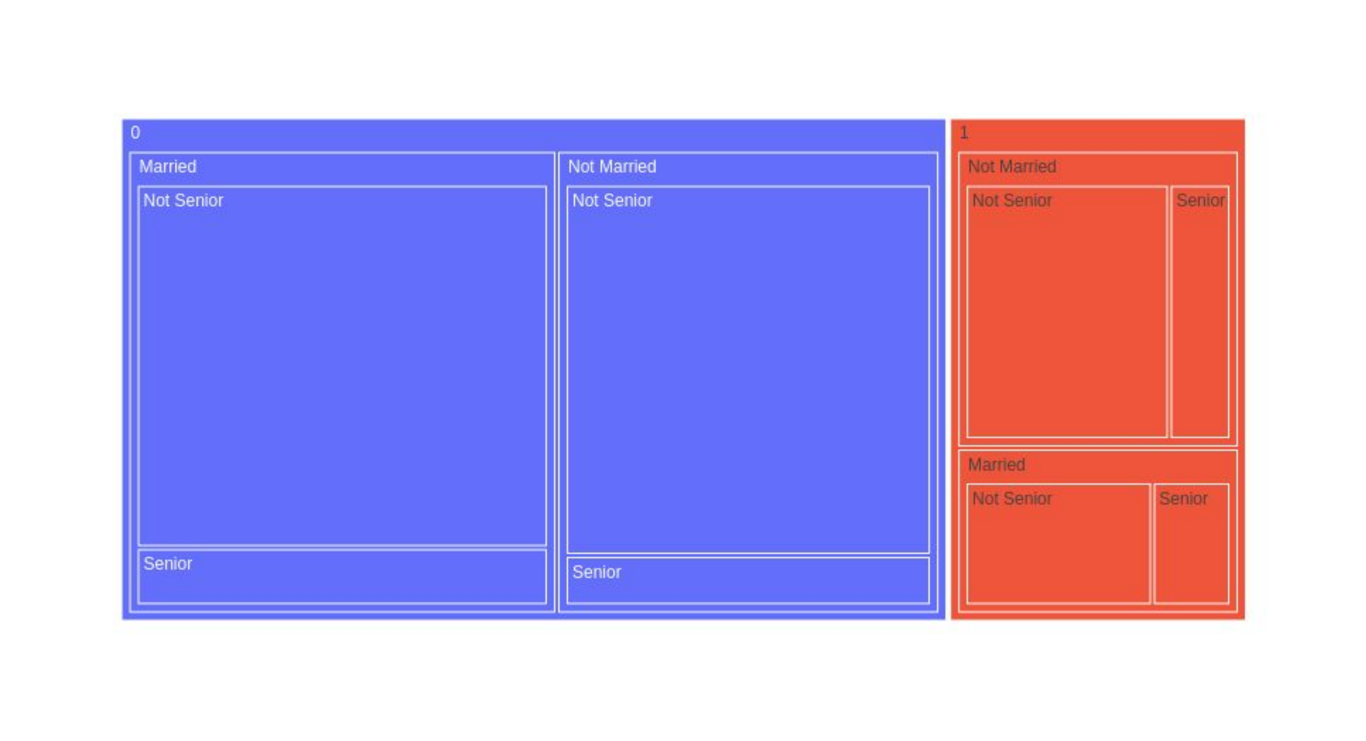
➔ Payment method has an effect on Churn

Payment Method vs Churn cont

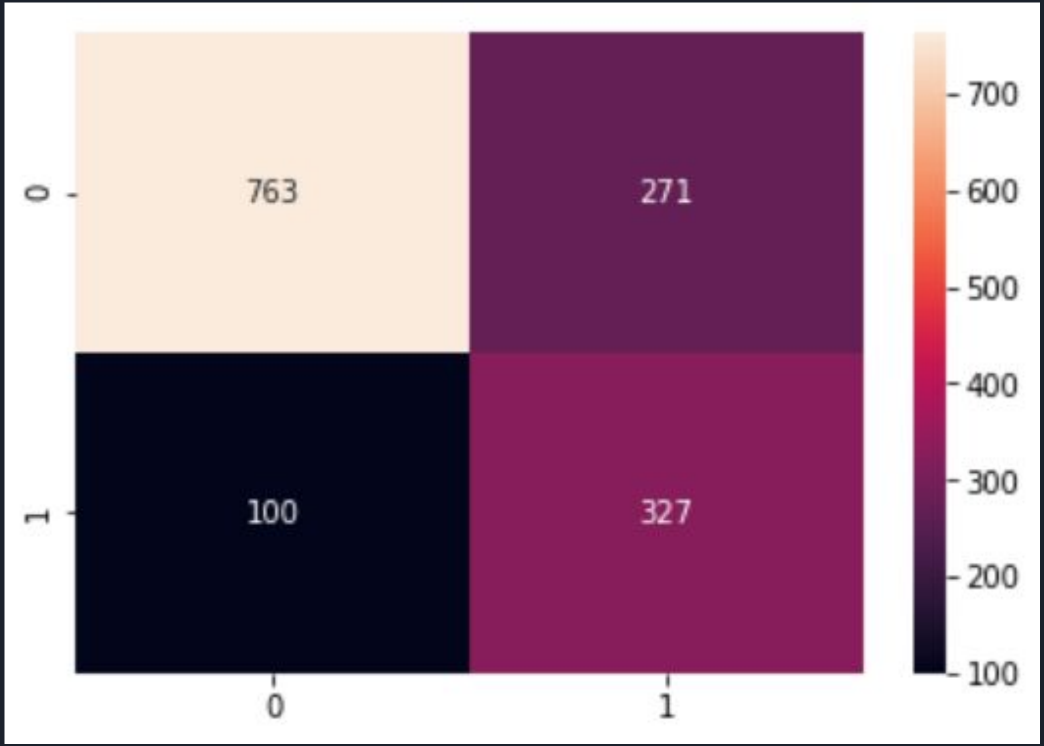
```
# Testing if Payment Method type will have impact on churn
ec_df = churn_pd_df[churn_pd_df.PaymentMethod == 'Electronic check']
mc_df = churn_pd_df[churn_pd_df.PaymentMethod == 'Mailed check']
bt_df = churn_pd_df[churn_pd_df.PaymentMethod == 'Bank transfer (automatic)']
cc_df = churn_pd_df[churn_pd_df.PaymentMethod == 'Credit card (automatic)']

f_stat, p_value = stats.f_oneway(ec_df['Churn'],mc_df['Churn'],bt_df['Churn'],cc_df['Churn'])

print('f_stat:', f_stat)
print('p_value:', p_value)

if p_value <= 0.05:
    print("Payment Method types have an effect on Churn.")
else:
    print("Payment Method types do not have an effect on Churn.")
```

```
f_stat: 237.80950657419586
p_value: 5.583595720582596e-147
Payment Method types have an effect on Churn.
```

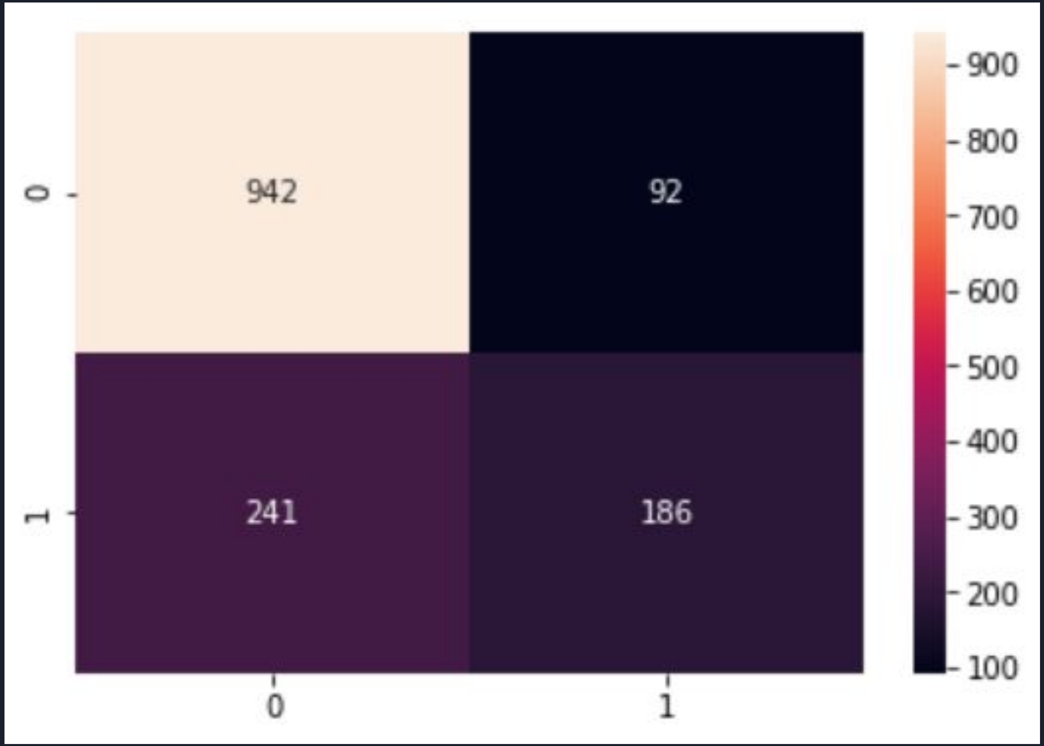
Treemap Analysis



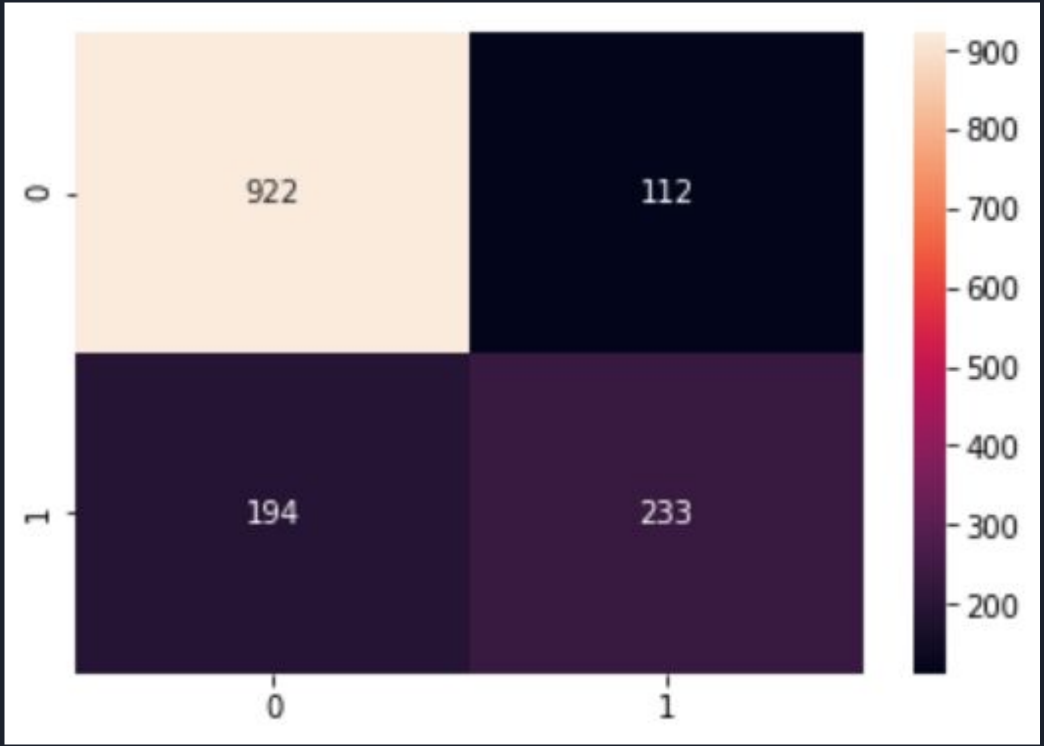
Logistic Regression



Decision Tree Confusion Matrix



Random Forest Confusion Matrix



Final Results

	Classifier Type	Accuracy %	Sensitivity %	Specificity %	Precision (1.0)	Recall (1.0)	f1-score
0	Logistic Regression	74.606434	76.580796	73.791103	0.55	0.77	0.64
1	Decision Tree	77.207392	43.559719	91.102515	0.67	0.44	0.53
2	Random Forest	79.055441	54.566745	89.168279	0.68	0.55	0.60

Logistic Regression - Highest Sensitivity
Decision Tree - Highest Specificity
Random Forest - Best Overall with Highest Accuracy

We choose Logistic Regression!

Questions?