Predicting Telecom Customer Churn

David Booker-Earley

Presentation Outline

- Intro
  - Problem Statement and Goal
  - Data Overview
- Exploratory Data Analysis
  - Target and Feature Selection
- Modeling
  - Supervised ML Models
  - Classification Reports
  - Model Evaluation
- Next Steps
- Appendix

The Problem

- Business compete!
  - “Get the most customers”
  - “Make more customers happy”
  - “Improve revenue”
- However, things change!
  - Will the customers churn?

Wait,

... quick question!

Hi there!

... I'm the ~~(pseudo)~~ on-screen assistant.

... How can I help?

... what does "churn" mean?

A customer "churns" when they stop their relationship with a business.

Oh, okay! Thanks!

Approaching the Problem

1. *Which customers are likely to continue or discontinue?*
2. *What factors affect these decisions from a customer's perspective?*
3. *How can the company adapt with minimal projected loss?*

Goal

- Predict whether a customer will churn.

How?

- Supervised machine learning solutions!

Data Overview

Telecommunications Churn Data

- 21 columns
- Provides various factors that might explain why telecom customers churn
  - Churn Status, Internet Service, Payment Method, Phone Service, etc.
  - Mostly categorical, text-based data
- 7,043 customer records
- Located on Kaggle

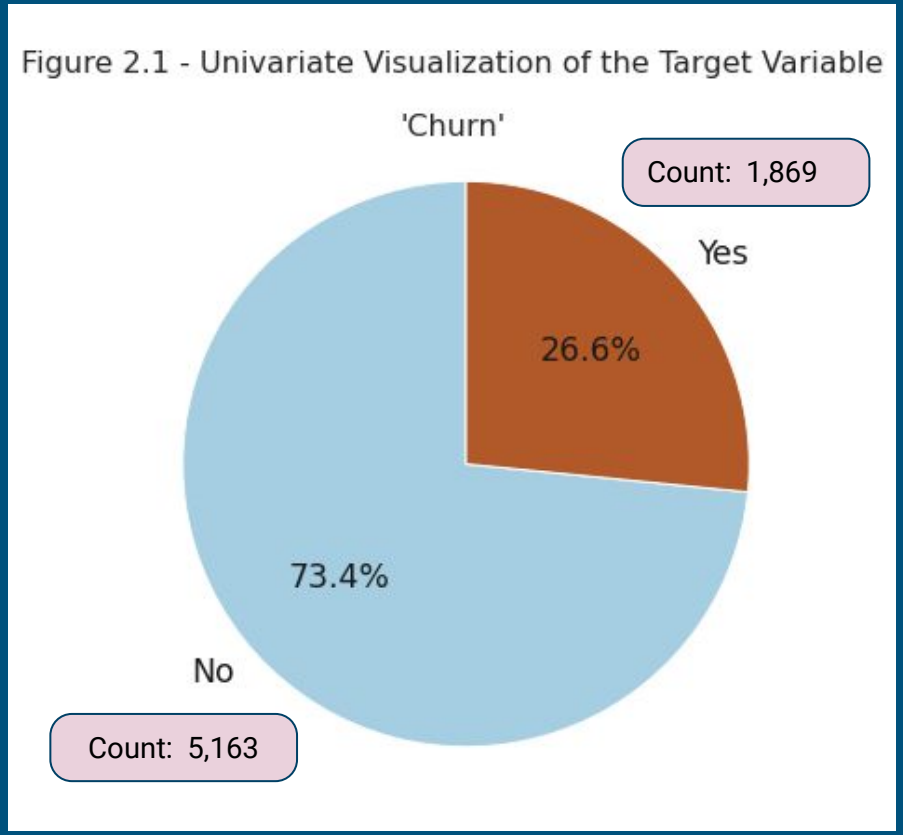
Exploration!

Target?

Churn

Making predictions?

Binary Classification Problem



Classes are Imbalanced!

Identifying the Target Variable (categorical)

How does churn status differ across Internet Service?

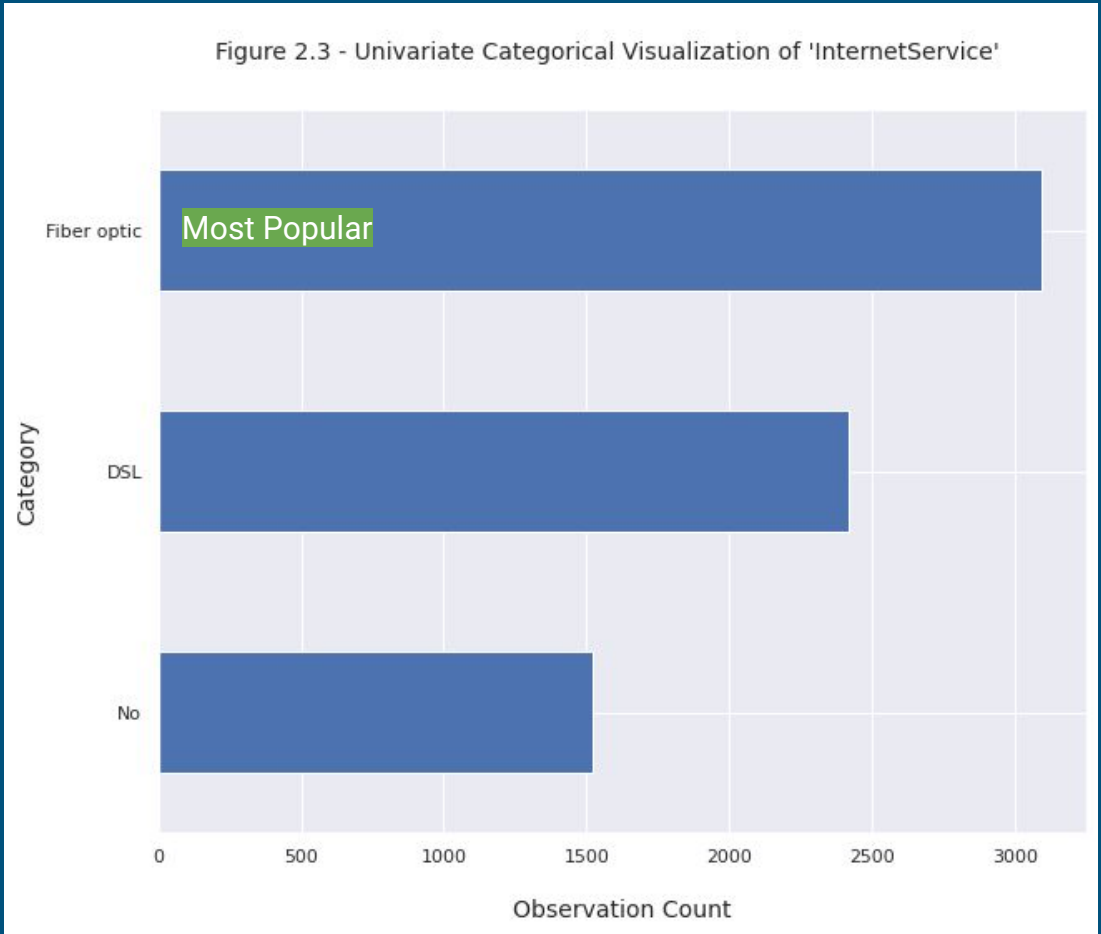


Figure 2.3 - Univariate Categorical Visualization of 'InternetService'
Most Popular
Fiber optic
DSL
No
Category
0
500
1000
1500
2000
2500
3000
Observation Count

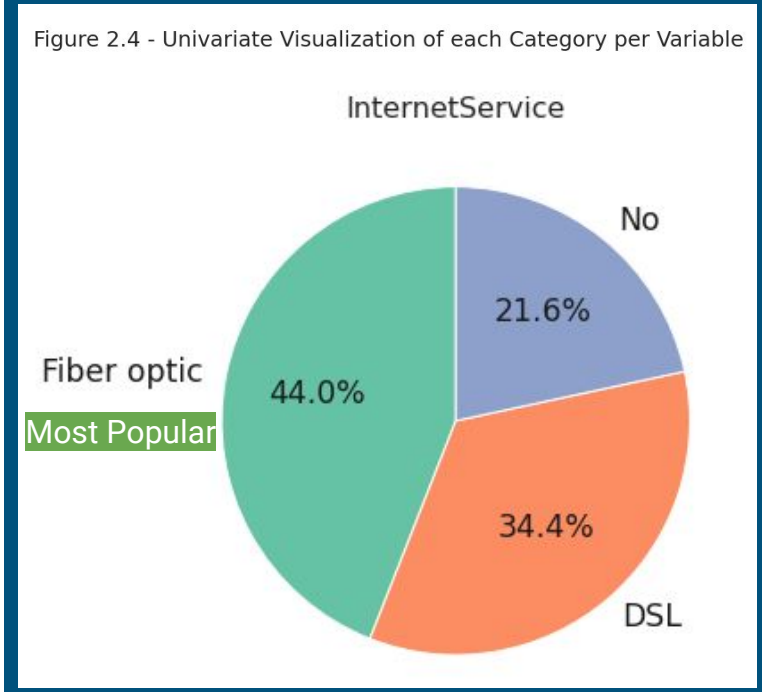


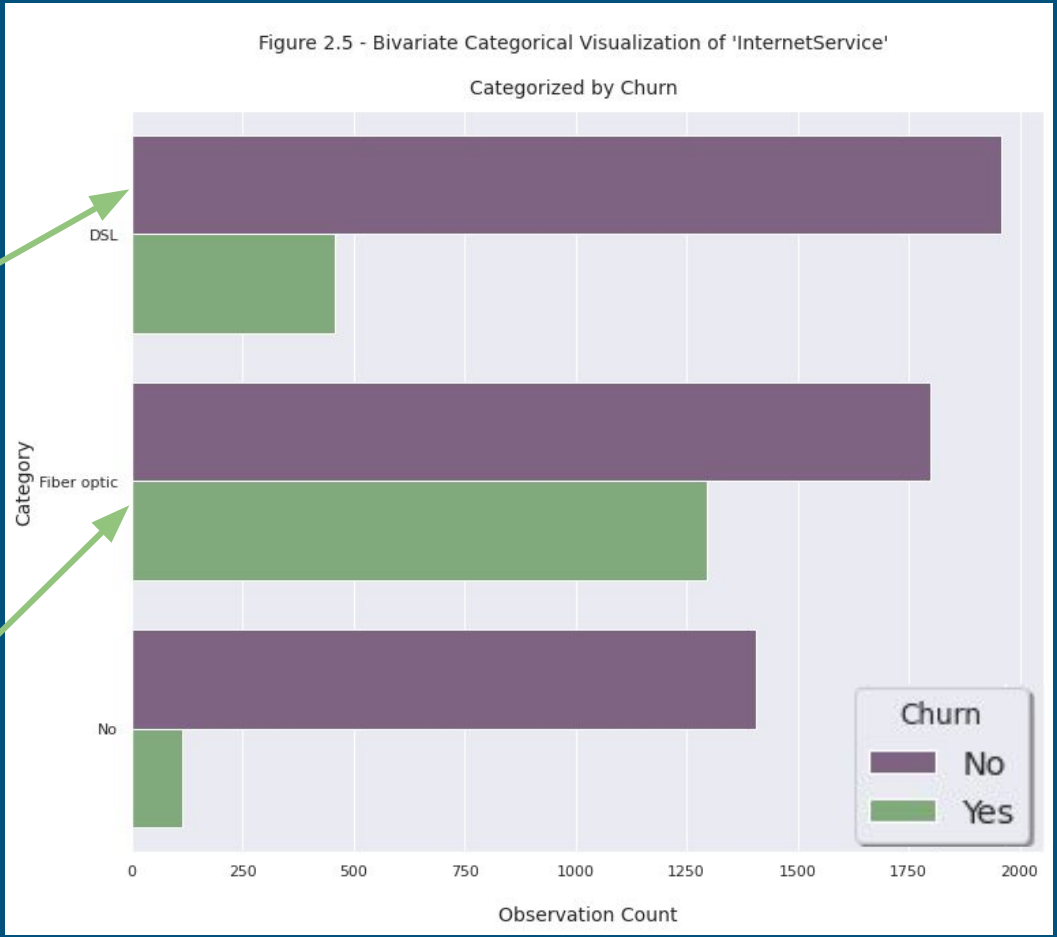
Figure 2.4 - Univariate Visualization of each Category per Variable
InternetService
Fiber optic
Most Popular
No
21.6%
44.0%
34.4%
DSL

DSL → 2nd Most Popular

- Most Non-Churns
- 2nd Lowest Churns

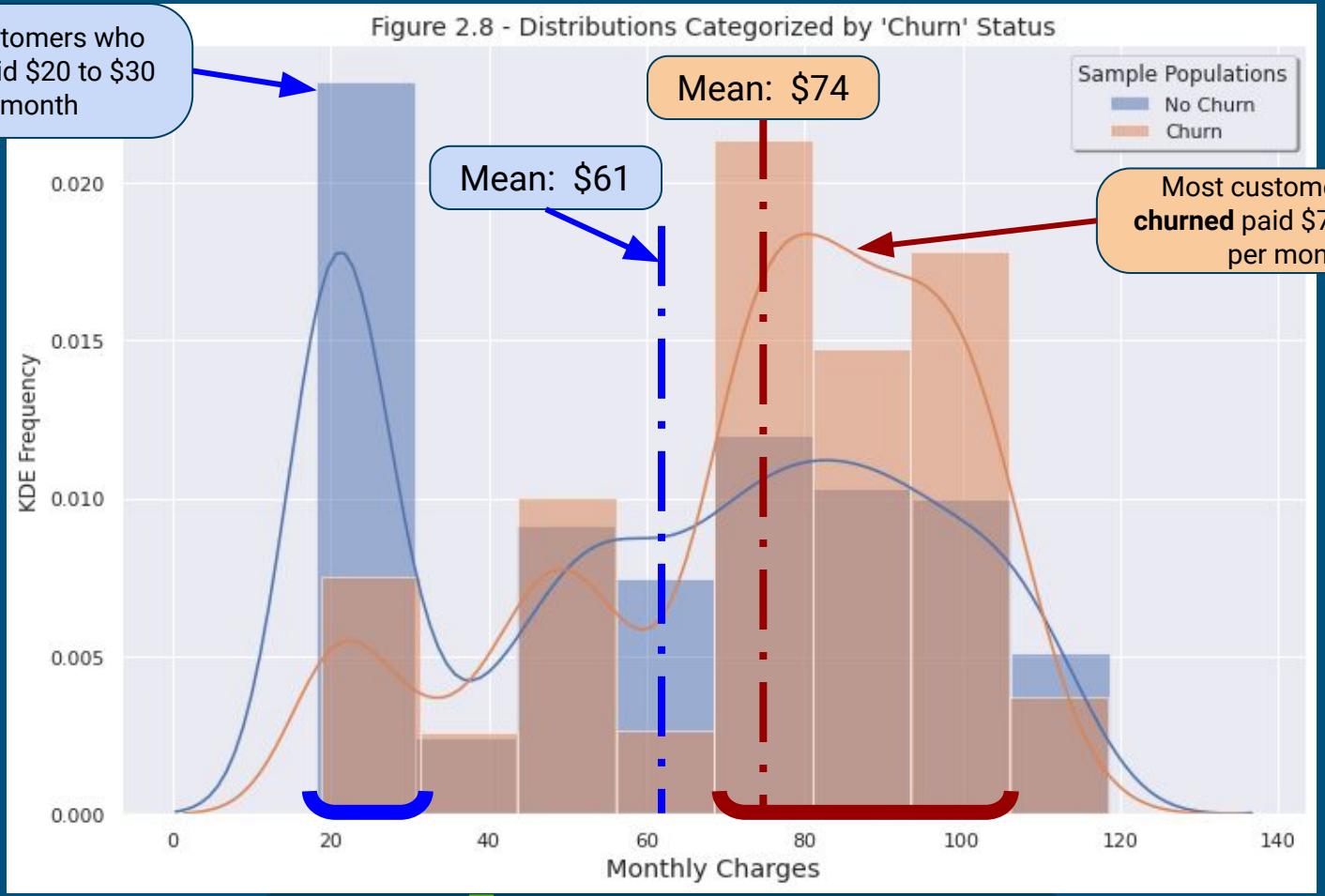
Fiber Optic → Most Popular

- 2nd Most Non-Churns
- Most Churns



How does churn status differ across Monthly Charges?

Most customers who **stayed** paid $20 to $30 per month



Most customers who **churned** paid $70 to $110 per month

Cool,

... so where are the predictions?

Glad you asked!

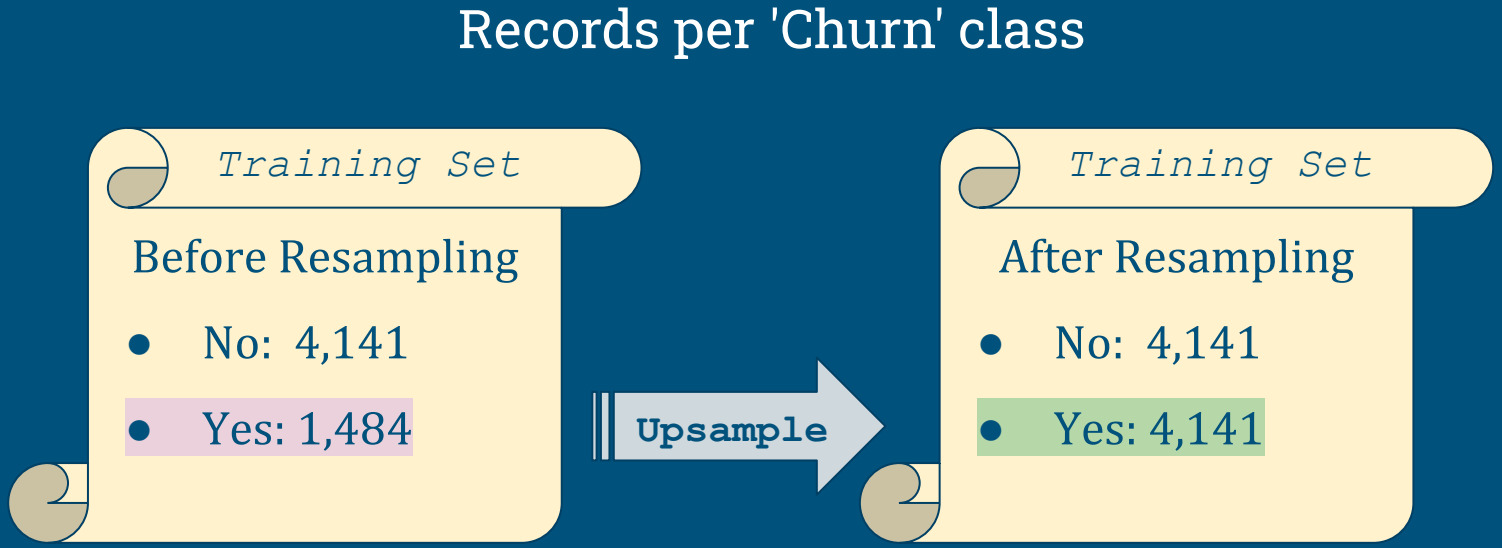
... We'll get to that soon!

... one does not simply "predict"

Model Preparations

- Split Data
  - 80% training, 20% testing
  - Overfitting Reduction
  - Test models on “new” data
- Upsample “Yes-Churn” Class
  - Class Balancing
  - Apply only on training set
- Feature Engineering
  - Dimensionality Reduction
  - Account for Collinearity
  - Reduce overall complexity
- Grid-Search Cross Validation
  - Tune model parameters during learning phase

Model Prep: Fixing Class Imbalance



Cool,

... but why, though??

... Short answer:

If all categories are not equally represented,

... the majority class would "win" most of the predictions.

Oh, I see, Fairness!!

Model Prep: Feature Engineering

Feature Selection: Part 1

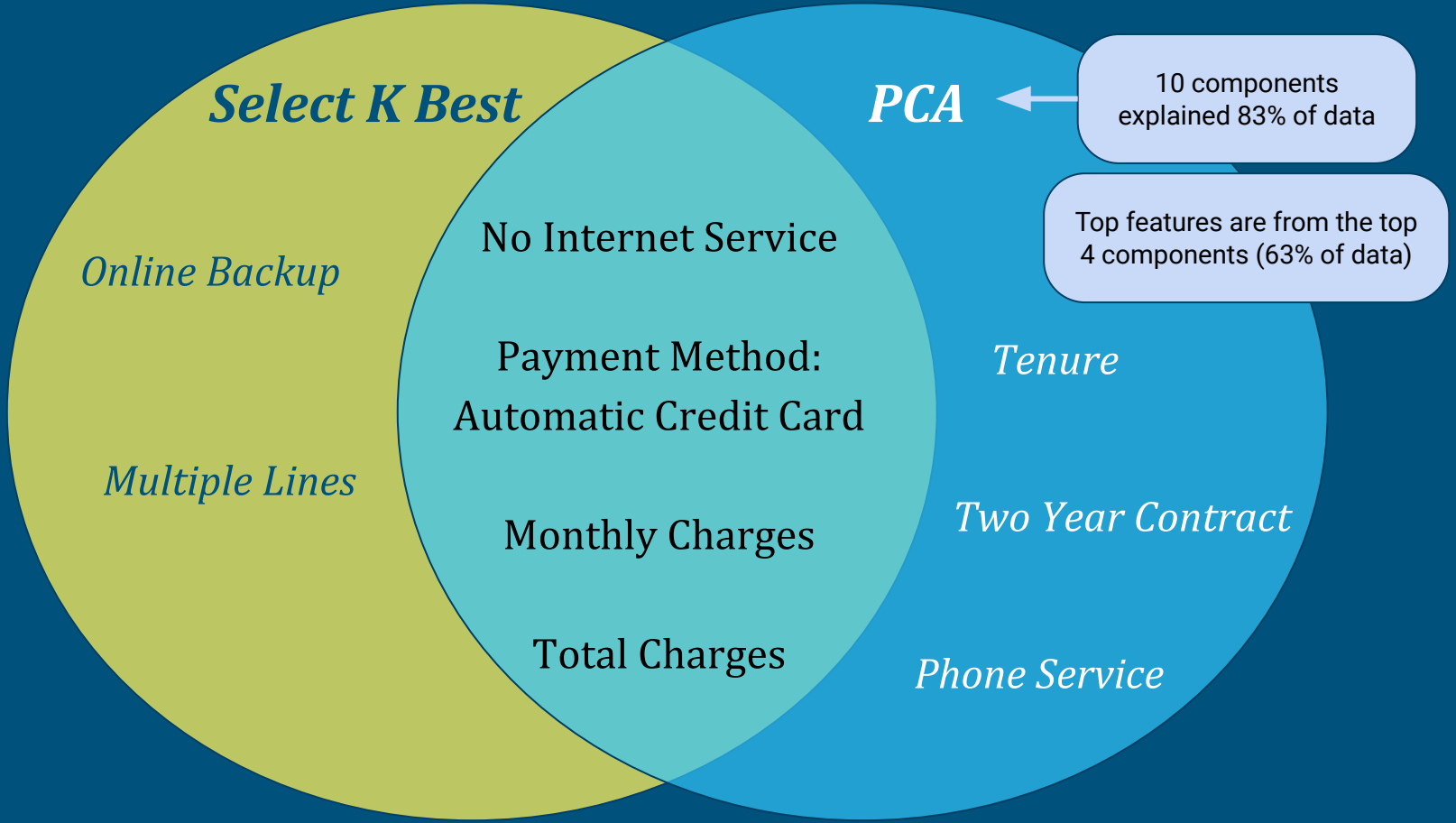
SelectKBest

- Conducts statistical tests
- Selects the *K*-best features

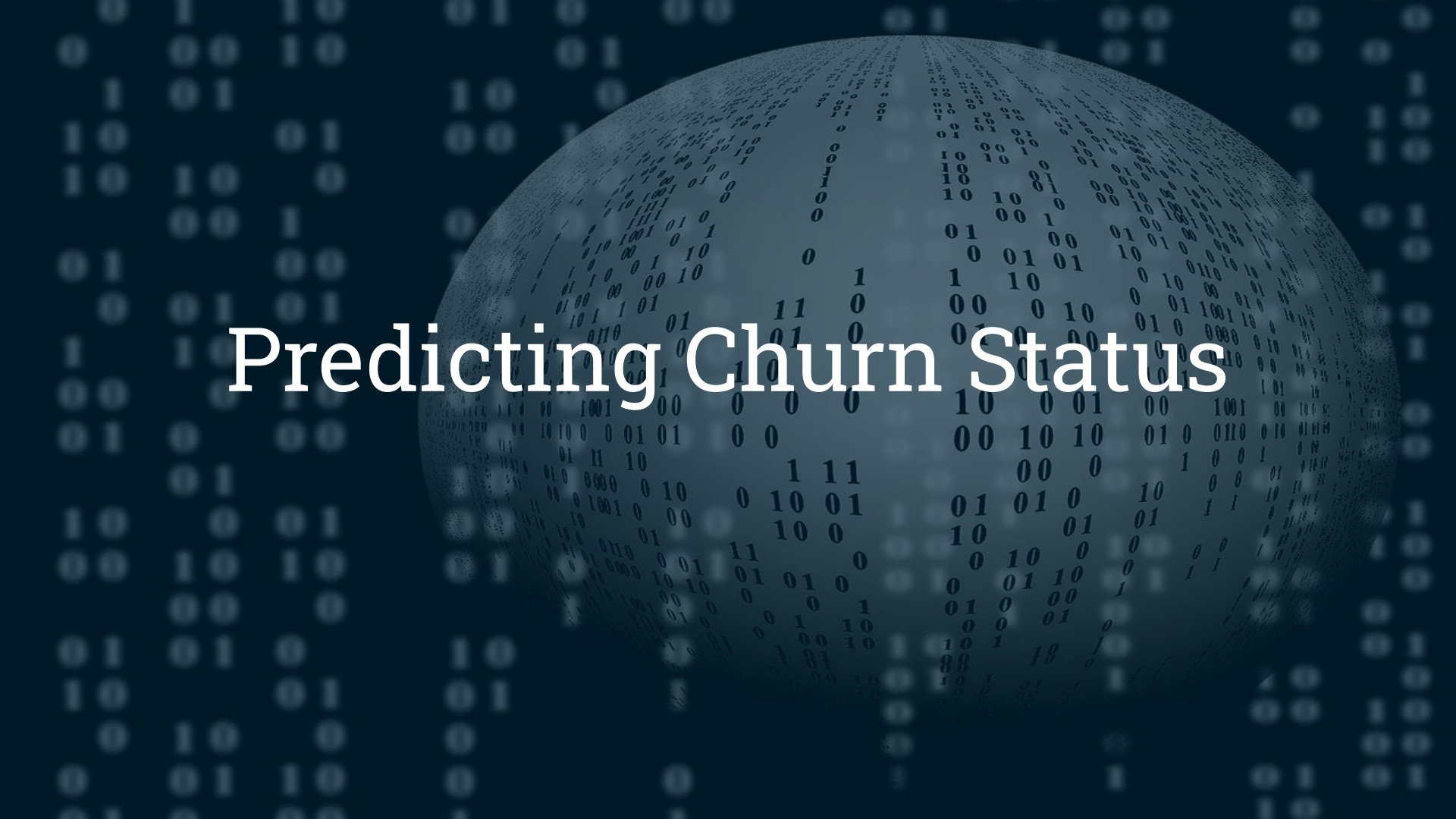
Feature Selection: Part 2

Principal Components Analysis (PCA)

- Captures collinear data
- Projects onto shared axis
- Reduces dimensionality
- Maintains explanatory power



Most important features that explain the most variance


Predicting Churn Status

Applying 6 Supervised ML Algorithms

- Logistic Regression as a classifier (LR)
- K Nearest Neighbors (KNN)
- Decision Tree (DT)
- Random Forest (RF)
- Support Vector Machines (SVM)
- Gradient Boosted Model of Decision Trees (GB)

Each with two feature-set variations

That's 12 models!

... but which is the best at predicting?

Great question!

... Let's jump right in!

We'll cover predictions from the top 4 ML models.

... The others can be found in the appendix :)

Awesome!!

Evaluating Model Performance

Which evaluation metric?

- Accuracy
- Precision
- Recall
- F_1 Score

Evaluating Model Performance

Which evaluation metric?

- ~~Accuracy~~
- Precision
- Recall
- F_1 Score

Trade-off! But both are important!

Models should predict each target class fairly well

K Nearest Neighbors

38% difference

```
---- CONFUSION MATRIX ----
[[660 362]
 [252 133]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.72      0.65      0.68      1022
         Yes       0.27      0.35      0.30       385

    accuracy                           0.56      1407
   macro avg       0.50      0.50      0.49      1407
weighted avg       0.60      0.56      0.58      1407
```

Feature Selection using *SelectKBest*

Random Forest

42% difference

```
---- CONFUSION MATRIX ----
[[691 331]
 [267 118]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.72      0.68      0.70      1022
         Yes       0.26      0.31      0.28       385

    accuracy                           0.57      1407
   macro avg       0.49      0.49      0.49      1407
weighted avg       0.60      0.57      0.58      1407
```

Feature Selection using *PCA*

Logistic Regression

Lowest False Negatives

```
---- CONFUSION MATRIX ----
[[417 605]
 [ 90 295]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.82      0.41      0.55      1022
         Yes       0.33      0.77      0.46       385

    accuracy                           0.51      1407
   macro avg       0.58      0.59      0.50      1407
weighted avg       0.69      0.51      0.52      1407
```

```
---- CONFUSION MATRIX ----
[[408 614]
 [ 87 298]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.82      0.40      0.54      1022
         Yes       0.33      0.77      0.46       385

    accuracy                           0.50      1407
   macro avg       0.58      0.59      0.50      1407
weighted avg       0.69      0.50      0.52      1407
```

Feature Selection using *SelectKBest*

Feature Selection using *PCA*

Logistic Regression

```
---- CONFUSION MATRIX ----
[[417 605]
 [ 90 295]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.82      0.41      0.55      1022
         Yes       0.33      0.77      0.46       385

    accuracy                           0.51      1407
   macro avg       0.58      0.59      0.50      1407
weighted avg       0.69      0.51      0.52      1407
```

9% difference

```
---- CONFUSION MATRIX ----
[[408 614]
 [ 87 298]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.82      0.40      0.54      1022
         Yes       0.33      0.77      0.46       385

    accuracy                           0.50      1407
   macro avg       0.58      0.59      0.50      1407
weighted avg       0.69      0.50      0.52      1407
```

8% difference

Feature Selection using *SelectKBest*

Feature Selection using *PCA*

Best Model

... Although all models performed poorly overall ...

- 1st Place → Logistic Regression
  - PCA Feature-Set Variation
  - Correctly predicted 77% of customers who would churn
  - Missed 87 churned customers
  - Runtime < 7 seconds

That was pretty cool!

... So, who is all of this for?

Chief executive officers, machine learning enthusiasts, anyone who is inherently inquisitive,

... and, of course,
... Data Scientists! :)

Next Steps

Next Steps for Further Research

1. Train and test other models on the given data.
2. Explore the relationship of how customer churn changed over time.
3. Investigate various aspects of churn-rates around the time(s) that other services or deals were offered.
4. Train and test all models with additional data (more customer records, more factors, etc.).
5. Expand on the variations for evaluation metrics and tools.
6. Based on those research results, discuss the newly discovered patterns surrounding customer churn and retention.

Thank you for your time!

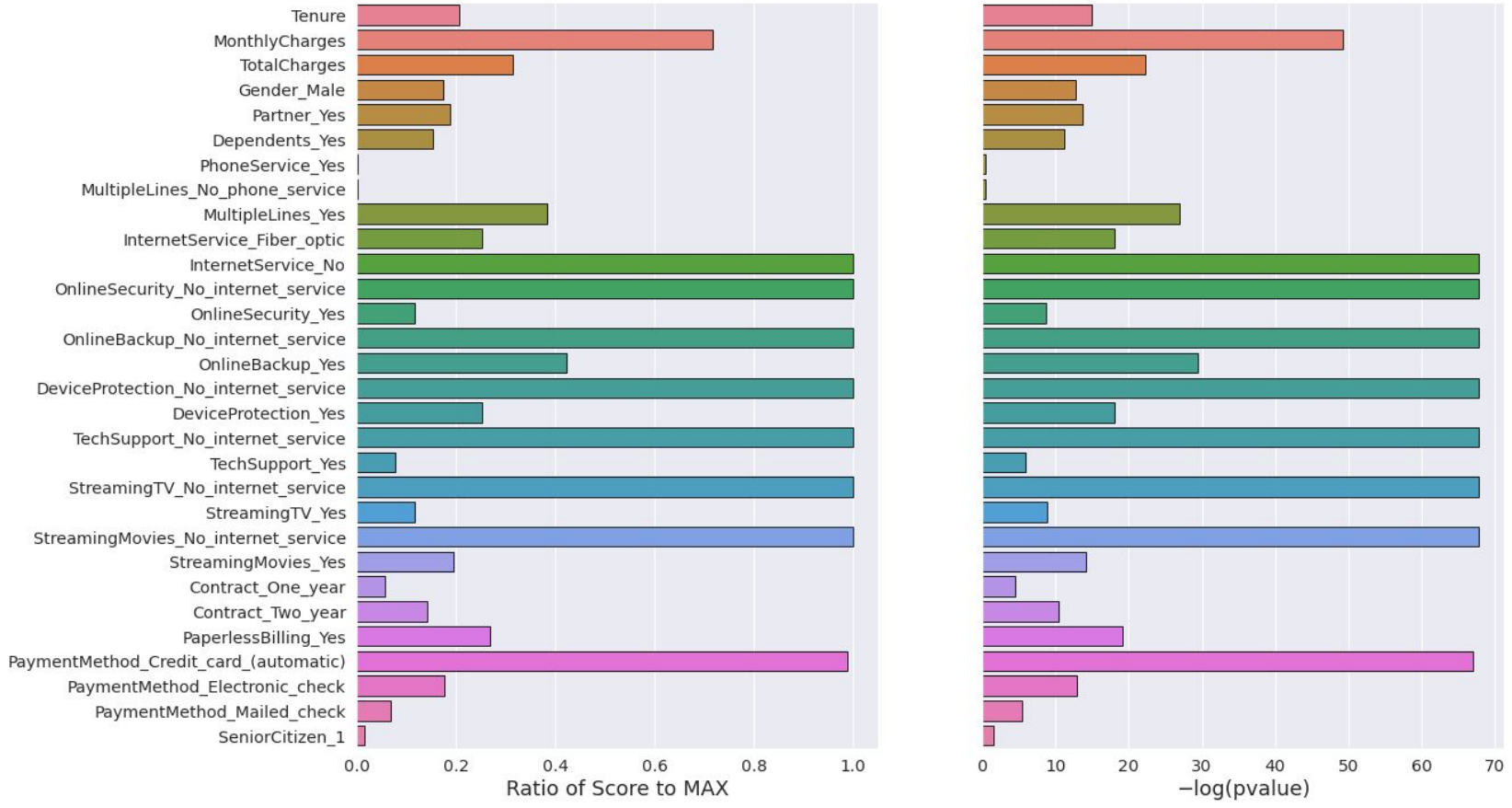
Any questions?

Appendix

Appendix Outline

- Feature Engineering
- Model Evaluation: Definitions and Equations
- Prediction Results
- Other Considerations
- Sources

Figure 3 - Feature Importance from 'SelectKBest'



```
Finding Collinearity among Features

The percentage % of "total variance in the dataset" captured
and explained by each principal component:

[39.36649679 11.13114163  7.72868565  5.03781444  4.08265487  3.69115619
  3.34892104  3.09432284  2.85020032  2.55426608]

Total: 83%
```

Together, these 10 components explain nearly 83% of the data.

- Thus, they can effectively replace the original 30 features.
- While there is a loss of nearly 17% of underlying information, the patterns that have been captured should suffice for each of the models to predict an outcome.

The 10 principal components used could explain nearly 83% of the variance in the data.

Figure 4.1 - Scores sorted by Correlation between Feature & PC:
Most Descriptive Features for 'PC_1'

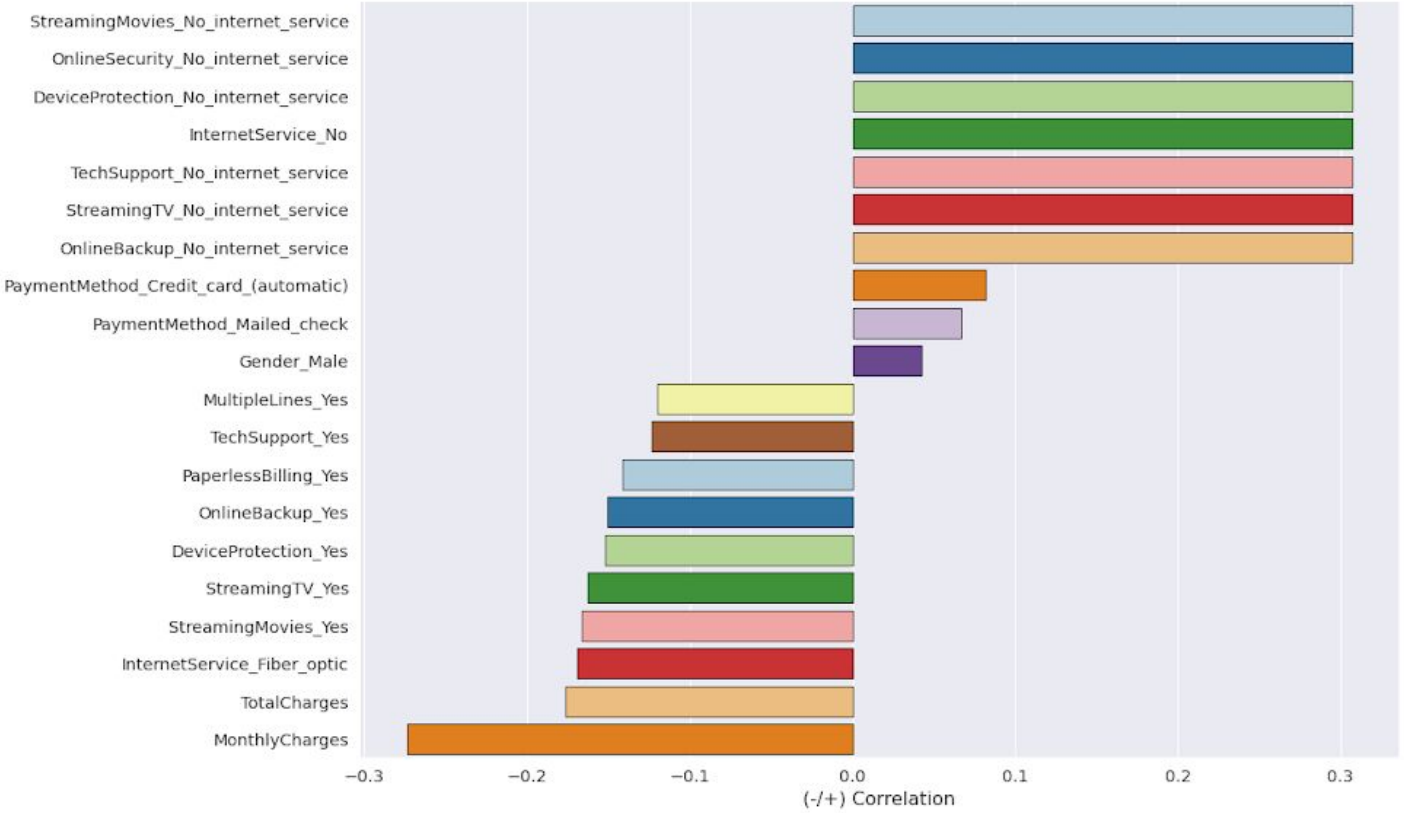
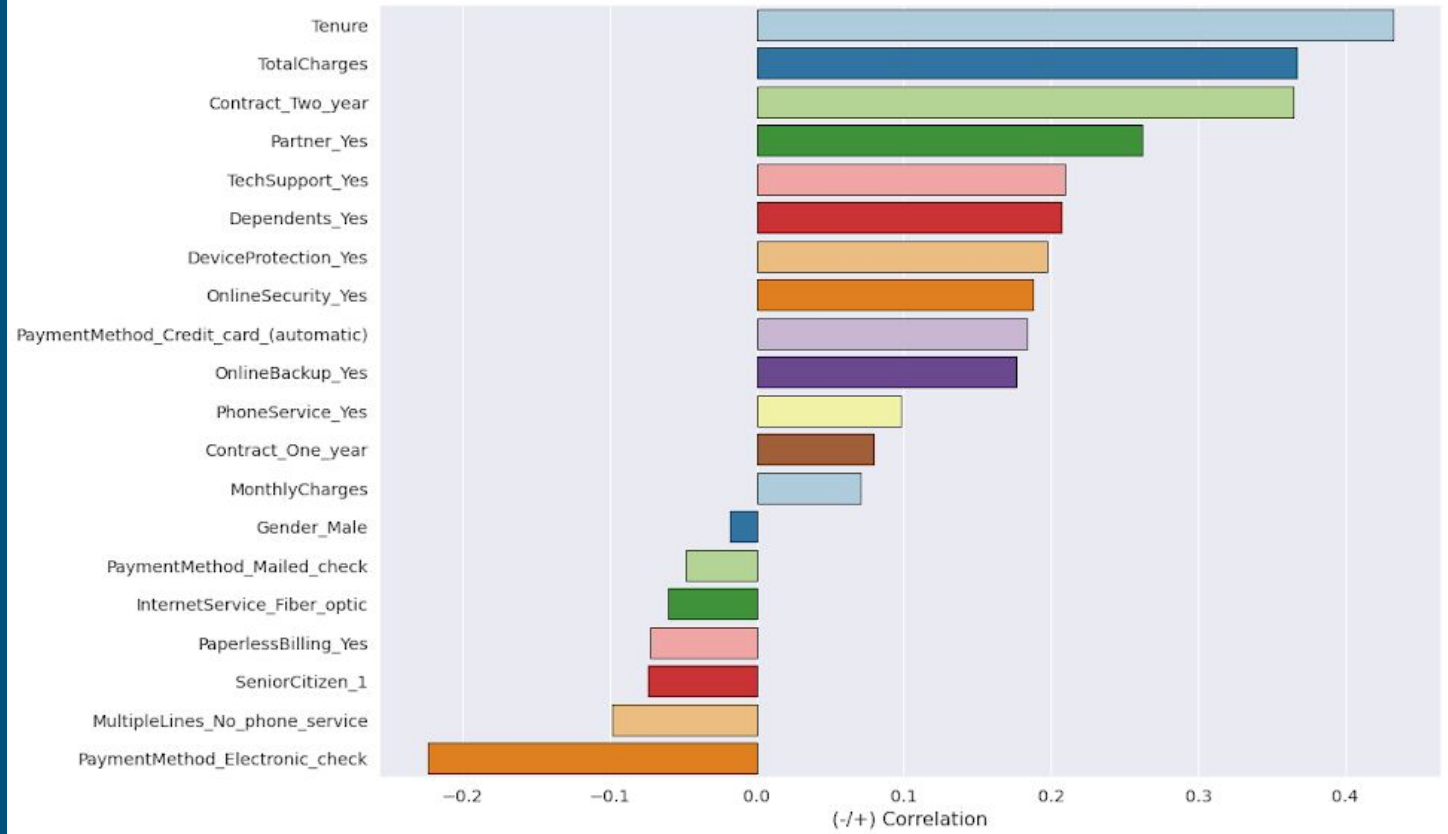


Figure 4.2 - Scores sorted by Correlation between Feature & PC:
Most Descriptive Features for 'PC_2'



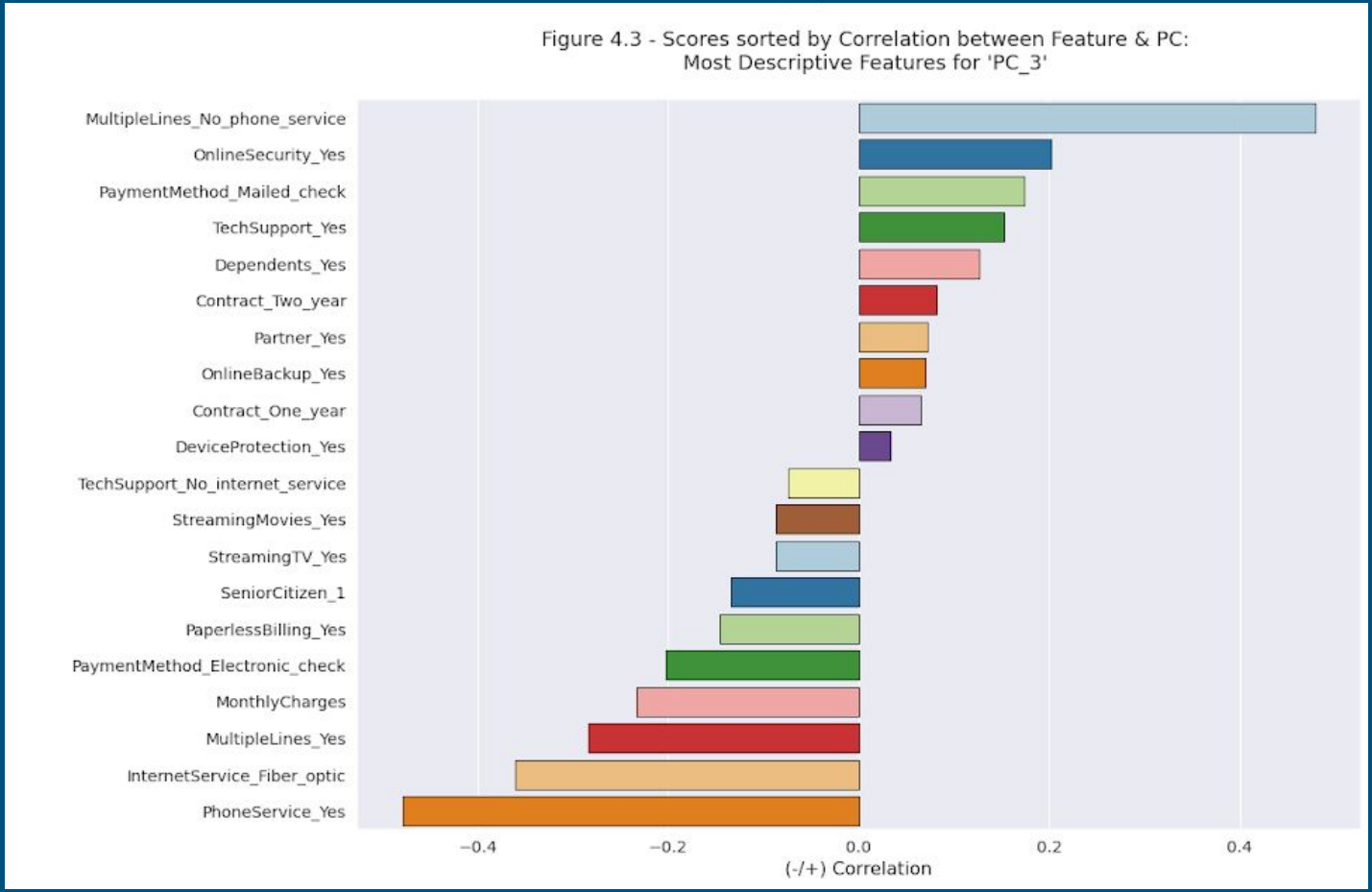


Figure 4.3 - Scores sorted by Correlation between Feature & PC:
Most Descriptive Features for 'PC_3'
MultipleLines_No_phone_service
OnlineSecurity_Yes
PaymentMethod_Mailed_check
TechSupport_Yes
Dependents_Yes
Contract_Two_year
Partner_Yes
OnlineBackup_Yes
Contract_One_year
DeviceProtection_Yes
TechSupport_No_internet_service
StreamingMovies_Yes
StreamingTV_Yes
SeniorCitizen_1
PaperlessBilling_Yes
PaymentMethod_Electronic_check
MonthlyCharges
MultipleLines_Yes
InternetService_Fiber_optic
PhoneService_Yes
−0.4
−0.2
0.0
0.2
0.4
(-/+) Correlation

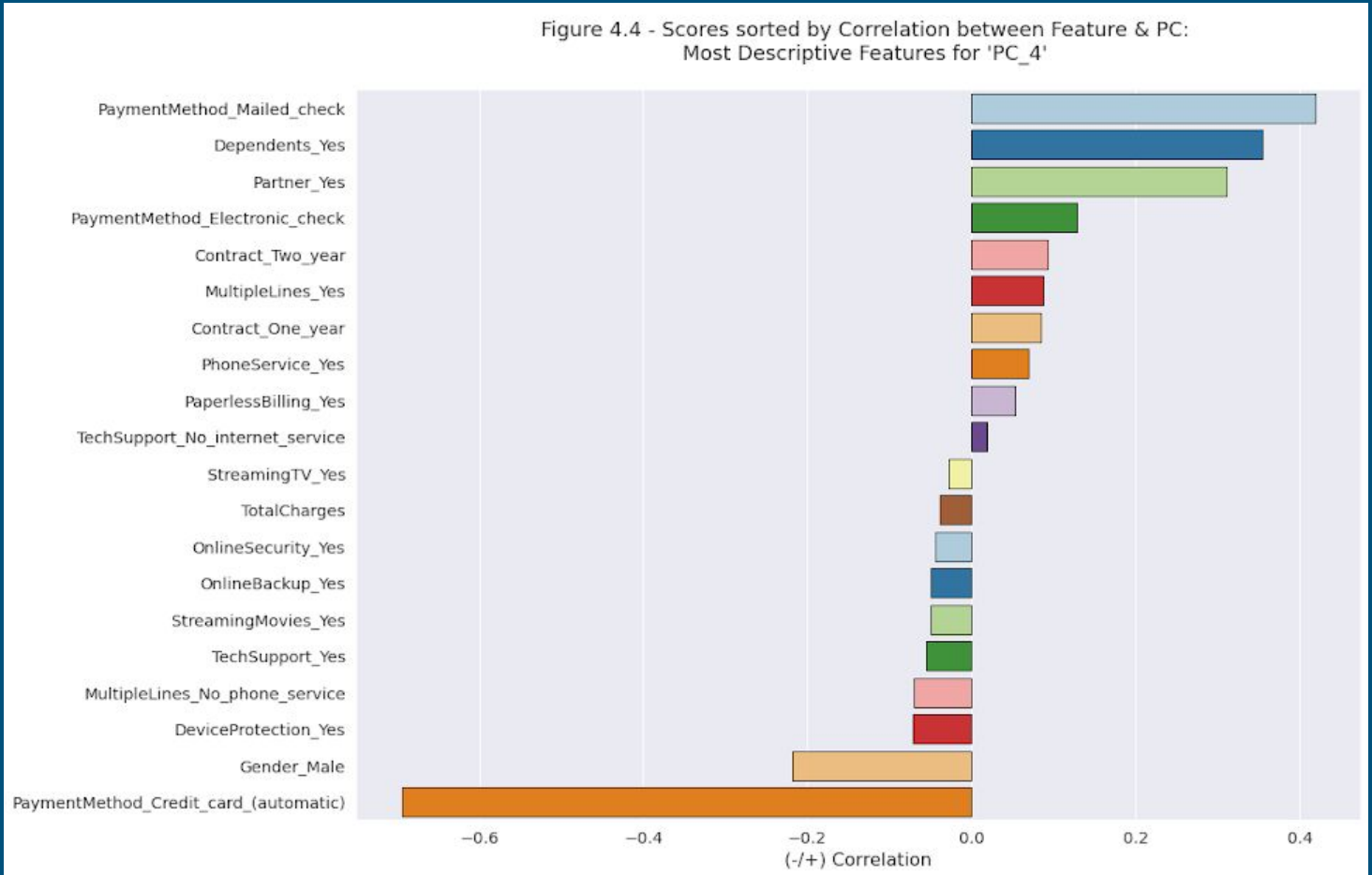


Figure 4.4 - Scores sorted by Correlation between Feature & PC:
Most Descriptive Features for 'PC_4'
PaymentMethod_Mailed_check
Dependents_Yes
Partner_Yes
PaymentMethod_Electronic_check
Contract_Two_year
MultipleLines_Yes
Contract_One_year
PhoneService_Yes
PaperlessBilling_Yes
TechSupport_No_internet_service
StreamingTV_Yes
TotalCharges
OnlineSecurity_Yes
OnlineBackup_Yes
StreamingMovies_Yes
TechSupport_Yes
MultipleLines_No_phone_service
DeviceProtection_Yes
Gender_Male
PaymentMethod_Credit_card_(automatic)
−0.6
−0.4
−0.2
0.0
0.2
0.4
(-/+) Correlation

Each *confusion matrix* has the following format:

$$\begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix}$$

- More specifically:

$$\begin{array}{rc} & \begin{bmatrix} True\ Negatives & False\ Positives \\ False\ Negatives & True\ Positives \end{bmatrix} \\ \text{True No} & \\ \text{True Yes} & \\ \ldots & \text{Predicted No} \quad \text{Predicted Yes} \end{array}$$

Confusion Matrix

On average, how often was each classifier correct?

> The `accuracy` of a given model can answer this question, and can be calculated by the following:

$$Accuracy = \frac{\#\ of\ Correct\ Predictions}{Total\ \#\ of\ Predictions} = \frac{TN + TP}{TN + FN + FP + TP}$$

$Where$:

- TN = True Negatives
- FN = False Negatives
- FP = False Positives
- TP = True Positives

Accuracy

What fraction of all **positive** predictions are actually correct?

`Precision` represents a classifier's ability to not improperly label a sample that is actually negative as positive (and vice versa).

- Here, it shows how many `Yes` outcomes were correctly predicted relative to the total amount predicted as `Yes`.
- Higher values indicate fewer false positives.

$$Precision = \frac{TP}{FP + TP}$$

Precision

What fraction of all **positive** instances does the model correctly predict as **positive**?

`Recall` represents a classifier's ability to identify all positive samples.

- It is a measure of correctness with respect to the total amount of true positives.
- Higher values indicate fewer false negatives.
- As such, it also indicates the amount of missed positive instances.
    - Other names for this evaluation metric: *true positive rate, sensitivity, probability of detection*
    - Note that for binary classification, recall of the positive class is known as *sensitivity*, while recall of the negative class is known as *specificity*.

$$Recall = \frac{TP}{FN + TP}$$

Recall

`F1 Score` combines precision and recall into a single number.

- It is the weighted harmonic mean of precision and recall.
- The best score is a value of 1, and the worst score is 0.
    - This is an implementation of the `F-Beta Score`, which weights recall more than precision by a factor of `beta`.
    - For the purpose of this project, `No` and `Yes` have equal importance when evaluating prediction outcomes.
    - Thus, `beta` is set to `1.0` to ensure that recall and precision are weighted equally.

$$F_{\beta=1} = 2 * \frac{Precision * Recall}{Precision + Recall} = \frac{2 * TP}{FP + FN + (2 * TP)}$$

F1 Score

Example: Perfect Recall

- 100 % of customers who would actually churn would be accounted for
- Low precision
- High number of false positives
- Poor resource allocation

→ Prioritize resource allocation for customers who'd churn!

→ Equal importance between "No" (negatives) and "Yes" (positives)!

```
'Logistic Regression (as Classifier)' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.571 (+/- 0.202)

Best parameter-set found after tuning:
{'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}

Best estimator:
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[417 605]
 [ 90 295]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.82      0.41      0.55      1022
         Yes       0.33      0.77      0.46       385

    accuracy                           0.51      1407
   macro avg       0.58      0.59      0.50      1407
weighted avg       0.69      0.51      0.52      1407
```

LR (SelectKBest)

```
'Logistic Regression (as Classifier)' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.554 (+/- 0.202)

Best parameter-set found after tuning:
{'max_iter': 100, 'penalty': 'l2', 'solver': 'lbfgs'}

Best estimator:
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[408 614]
 [ 87 298]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.82      0.40      0.54      1022
         Yes       0.33      0.77      0.46       385

    accuracy                           0.50      1407
   macro avg       0.58      0.59      0.50      1407
weighted avg       0.69      0.50      0.52      1407
```

LR (PCA)

```
'K-Nearest Neighbors Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.709 (+/- 0.251)

Best parameter-set found after tuning:
{'algorithm': 'auto', 'n_neighbors': 3, 'p': 1, 'weights': 'distance'}

Best estimator:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=1,
                     weights='distance')

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[660 362]
 [252 133]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.72      0.65      0.68      1022
         Yes       0.27      0.35      0.30       385

    accuracy                           0.56      1407
   macro avg       0.50      0.50      0.49      1407
weighted avg       0.60      0.56      0.58      1407
```

KNN (SelectKBest)

```
'K-Nearest Neighbors Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.806 (+/- 0.157)

Best parameter-set found after tuning:
{'algorithm': 'auto', 'n_neighbors': 3, 'p': 2, 'weights': 'distance'}

Best estimator:
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=3, p=2,
                     weights='distance')

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[681 341]
 [273 112]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.71      0.67      0.69      1022
         Yes       0.25      0.29      0.27       385

    accuracy                           0.56      1407
   macro avg       0.48      0.48      0.48      1407
weighted avg       0.59      0.56      0.57      1407
```

KNN (PCA)

```
'Decision Tree Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.840 (+/- 0.117)

Best parameter-set found after tuning:
{'criterion': 'gini', 'max_depth': 30}

Best estimator:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=30, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[751 271]
 [287  98]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.72      0.73      0.73      1022
         Yes       0.27      0.25      0.26       385

    accuracy                           0.60      1407
   macro avg       0.49      0.49      0.49      1407
weighted avg       0.60      0.60      0.60      1407
```

DT (SelectKBest)

```
'Decision Tree Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.739 (+/- 0.283)

Best parameter-set found after tuning:
{'criterion': 'gini', 'max_depth': 30}

Best estimator:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=30, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=None, splitter='best')

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[747 275]
 [273 112]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.73      0.73      0.73      1022
         Yes       0.29      0.29      0.29       385

    accuracy                           0.61      1407
   macro avg       0.51      0.51      0.51      1407
weighted avg       0.61      0.61      0.61      1407
```

DT (PCA)

```
'Random Forest Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.847 (+/- 0.095)

Best parameter-set found after tuning:
{'criterion': 'gini', 'max_depth': 14, 'n_estimators': 1000}

Best estimator:
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=14, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=1000,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[853 169]
 [285 100]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.75      0.83      0.79      1022
         Yes       0.37      0.26      0.31       385

    accuracy                           0.68      1407
   macro avg       0.56      0.55      0.55      1407
weighted avg       0.65      0.68      0.66      1407
```

RF (SelectKBest)

```
'Random Forest Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.800 (+/- 0.119)

Best parameter-set found after tuning:
{'criterion': 'gini', 'max_depth': 14, 'n_estimators': 50}

Best estimator:
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=14, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=50,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[691 331]
 [267 118]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.72      0.68      0.70      1022
         Yes       0.26      0.31      0.28       385

    accuracy                           0.57      1407
   macro avg       0.49      0.49      0.49      1407
weighted avg       0.60      0.57      0.58      1407
```

RF (PCA)

```
'Support Vector Machines Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.747 (+/- 0.332)

Best parameter-set found after tuning:
{'C': 100.0, 'gamma': 0.9, 'kernel': 'rbf'}

Best estimator:
SVC(C=100.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.9, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[853 169]
 [311  74]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.73      0.83      0.78      1022
         Yes       0.30      0.19      0.24       385

    accuracy                           0.66      1407
   macro avg       0.52      0.51      0.51      1407
weighted avg       0.62      0.66      0.63      1407
```

SVM (SelectKBest)

```
'Support Vector Machines Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.733 (+/- 0.308)

Best parameter-set found after tuning:
{'C': 100.0, 'gamma': 0.9, 'kernel': 'rbf'}

Best estimator:
SVC(C=100.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma=0.9, kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[806 216]
 [300  85]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.73      0.79      0.76      1022
         Yes       0.28      0.22      0.25       385

    accuracy                           0.63      1407
   macro avg       0.51      0.50      0.50      1407
weighted avg       0.61      0.63      0.62      1407
```

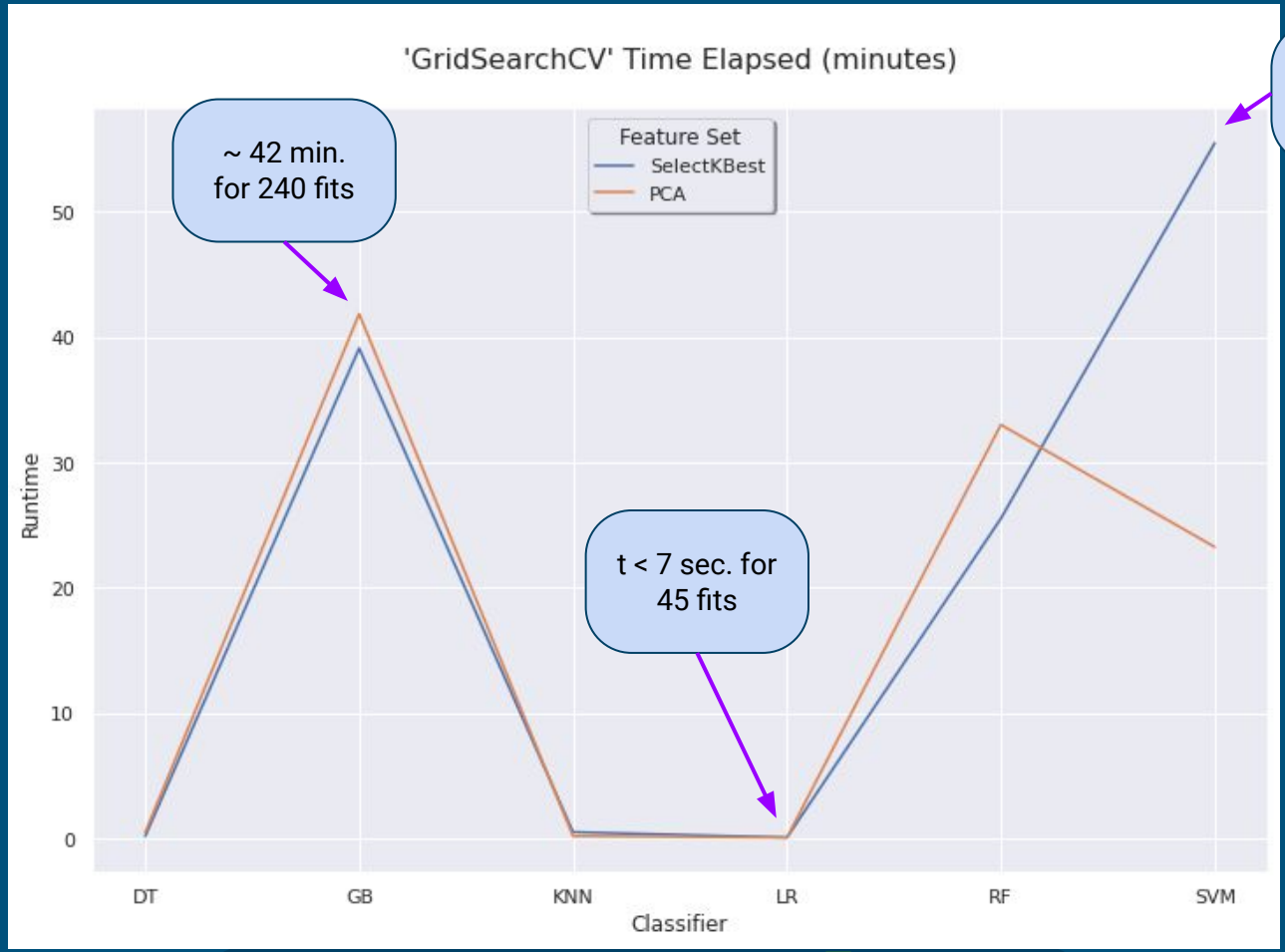
SVM (PCA)

```
'Gradient Boosting Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.852 (+/- 0.099)

Best parameter-set found after tuning:
{'criterion': 'friedman_mse', 'learning_rate': 0.1, 'loss': 'deviance', 'max_depth': 5,
 'min_samples_split': 2, 'n_estimators': 1000, 'subsample': 1.0}

Best estimator:
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.1, loss='deviance', max_depth=5,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=1000,
                           n_iter_no_change=None, presort='deprecated',
                           random_state=None, subsample=1.0, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[830 192]
 [304  81]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.73      0.81      0.77      1022
         Yes       0.30      0.21      0.25       385

    accuracy                           0.65      1407
   macro avg       0.51      0.51      0.51      1407
weighted avg       0.61      0.65      0.63      1407
```

GB (SelectKBest)

```
'Gradient Boosting Classifier' Complete
Predictions and run-time have been saved for model evaluation!

**********************************************************************

Mean Cross Validation Score (95% confidence interval)
0.762 (+/- 0.307)

Best parameter-set found after tuning:
{'criterion': 'friedman_mse', 'learning_rate': 0.1, 'loss': 'deviance', 'max_depth': 5,
 'min_samples_split': 2, 'n_estimators': 750, 'subsample': 1.0}

Best estimator:
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.1, loss='deviance', max_depth=5,
                           max_features=None, max_leaf_nodes=None,
                           min_impurity_decrease=0.0, min_impurity_split=None,
                           min_samples_leaf=1, min_samples_split=2,
                           min_weight_fraction_leaf=0.0, n_estimators=750,
                           n_iter_no_change=None, presort='deprecated',
                           random_state=None, subsample=1.0, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)

**********************************************************************

PREDICTION RESULTS

---- CONFUSION MATRIX ----
[[819 203]
 [313  72]]

---- DETAILED CLASSIFICATION REPORT ----
              precision    recall  f1-score   support

          No       0.72      0.80      0.76      1022
         Yes       0.26      0.19      0.22       385

    accuracy                           0.63      1407
   macro avg       0.49      0.49      0.49      1407
weighted avg       0.60      0.63      0.61      1407
```

GB (PCA)


'GridSearchCV' Time Elapsed (minutes)
~ 42 min.
for 240 fits
~ 1 hour for
150 fits
t < 7 sec. for
45 fits
Feature Set
SelectKBest
PCA
Runtime
0
10
20
30
40
50
DT
GB
KNN
LR
RF
SVM
Classifier

Optimizing the Balance between Precision and Recall

> For the purpose of this project, "No Churn" (negatives) and "Yes Churn" (positives) had equal importance during the evaluation phase. Thus, precision and recall were considered to be equally important. Further research could include variations of this; as an example, recall could be weighted more than precision by a factor of beta in the calculation of the F-Beta Score.

> To navigate trade-offs between these two metrics, further evaluation can also occur through a precision-recall curve, which outlines the relationship between precision and recall as the desired threshold is varied from 0 to 1 (the ideal value).

Optimizing Computational Power

> A Cross Validation Grid Search is a powerful approach for iteratively testing multitudes of parameter candidates. Further training and testing could be implemented by executing a memory-efficient program on a technologically well-crafted computer or cloud based platform.

Other Considerations

Recommended Updates for Data Collection

> The `Tenure` data provided the *duration* of maintained customer status. However, the respective *start* and *end* dates are not provided. Further research should be conducted to uncover additional patterns regarding dated churn status and dated prices (or payments). As an example, several customers might discontinue their contracts due to relatively high-priced services; after which, a company might offer a significant discount to get new customers to sign-up quickly. Underlying value might also exist in tracked payments; for example, some customers who are frequently late on payments might be more likely to churn than others.

. . . . . .

> For future data collection processes, tracking some or all of the following items could help avoid adverse effects in data analysis and interpretation:
>
> - Timing and Quality of Services
>   - *At which time(s) of the year were customers churning or joining, respectively?*
>   - *At which time(s) of the year were discounts offered by a given company?*
>   - *How were the services rated by the customers?*
>   - *How did these ratings change over time?*

> These questions outline how the sample population of customers may differ from the desired (actual) population, along with secondary metrics that could provide insight into the observed outcomes. Without knowing the answers to these questions, it's unclear how one could test for *sampling / selection bias*.

Sources

1. Kaggle Datasets
    a. https://www.kaggle.com/zagarsuren/telecom-churn-dataset-ibm-watson-analytics
2. Decorative Pictures (binary code, computers, cape)
    a. https://pixabay.com/