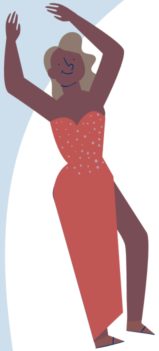
Consumer Churn Prediction w/ ML using Pyspark

Professor: Ming-Hwa Wang
Course: DATA228 Big Data
Group 1: Johnny Qiu, Manyu Zhang, Yuan Pan, Yue Zhang

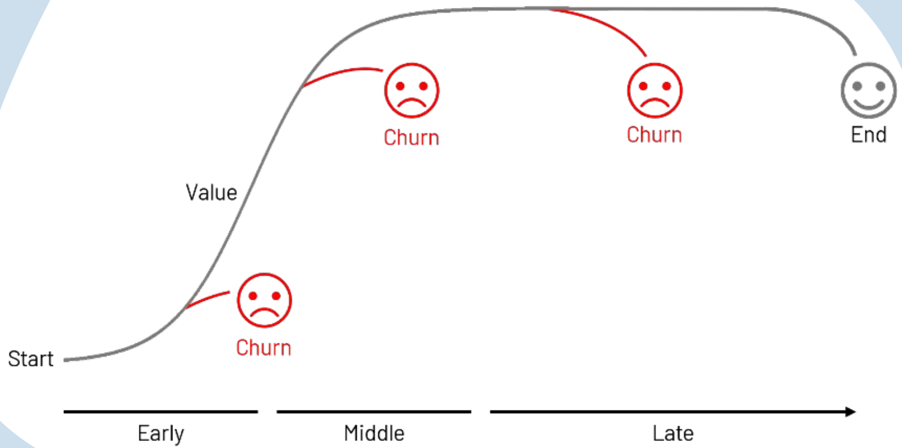
Table Of Contents



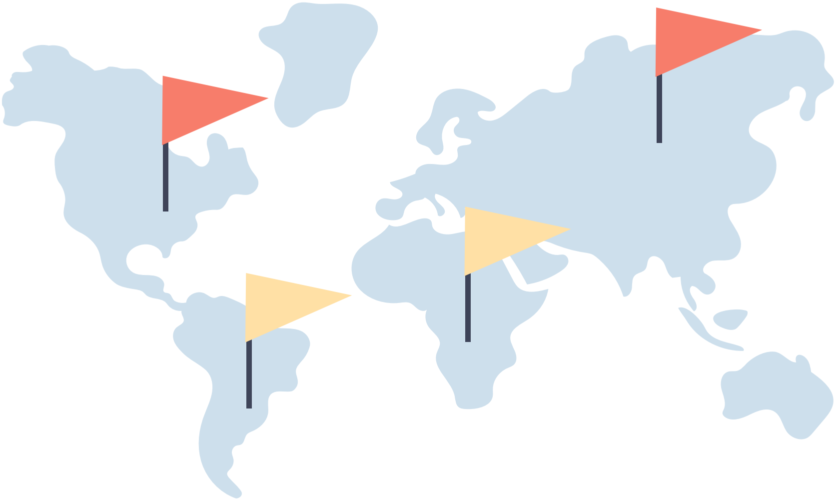
01

Background

What is Consumer Churn?



Forecast User in Telecom Sector Worldwide (2025)

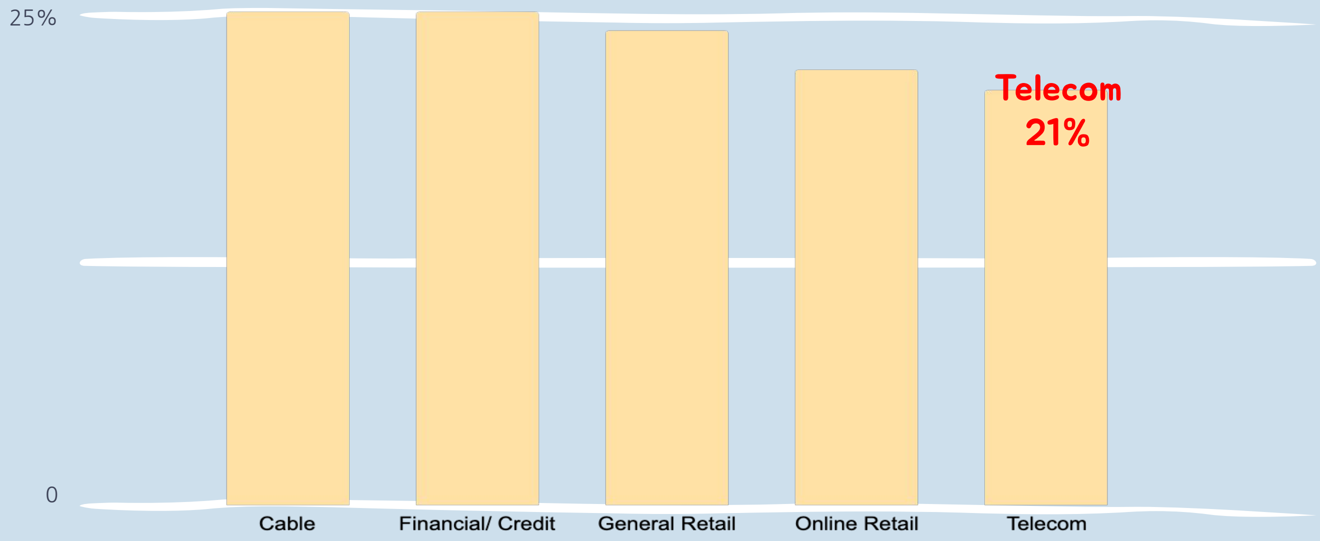


7.49

billion

*Source: https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/

Top 5 Consumer Churn Rate in the US by industry (2020)



*Source: https://www.statista.com/statistics/816735/customer-churn-rate-by-industry-us/

02

What Others Have Done

Literature Review by Steps

Evaluation

Algorithm Selection

Address Data Imbalance

Metrics

Accuracy
Precision
Recall
F1 Score
Area under ROC curve
...

Algorithms

K-means Clustering
Decision Trees
Logistic Regression
Artificial Neural Networks
...

Data Resampling

Random Over Sampling
Random Under Sampling
SMOTE
ADASYN
...

Pros And Cons

Pros

1. Valuable insights in model selection
   (KNN, SVM, LR, XGBoost, etc.)
2. Use ensemble methods to increase accuracy
   (Soft Set Ensemble Selection and Combination, etc.)
3. Create new measurement to improve model performance
   (cost function, etc.)

Cons

1. Imbalanced dataset and no techniques to address it.
2. Consideration of only one performance metric.
3. Overlook of interpretability, chose too complex algorithm

Project Goals

Determine the optimal combination of

techniques to address data imbalance issues and

algorithms for predicting customer churn

in the telecommunications industry.

	Unsupervised Algorithm	Supervised Algorithm	Ensemble Method
Technique 1			
Technique 2			
Technique 3			
...			

03

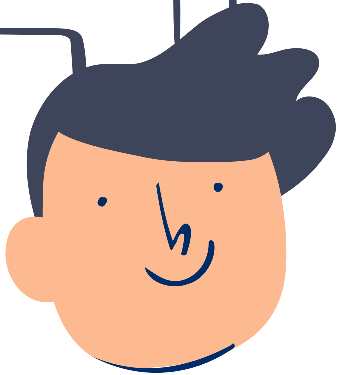
Hypotheses

Study Questions and Hypotheses

1. Necessity of data resampling?
- Algorithm performance improved after data resampling.
- SMOTE will be the most effective resampling method.

2. Which algorithm perform the best?
- DT and RF would deliver the best results with their robustness towards imbalance dataset.

3. Necessity of model tuning?
- Hyperparameter tuning would improve model performance.



04

Methodology & Implementation

Data Selection

IBM Data and AI Community:

5 data modules

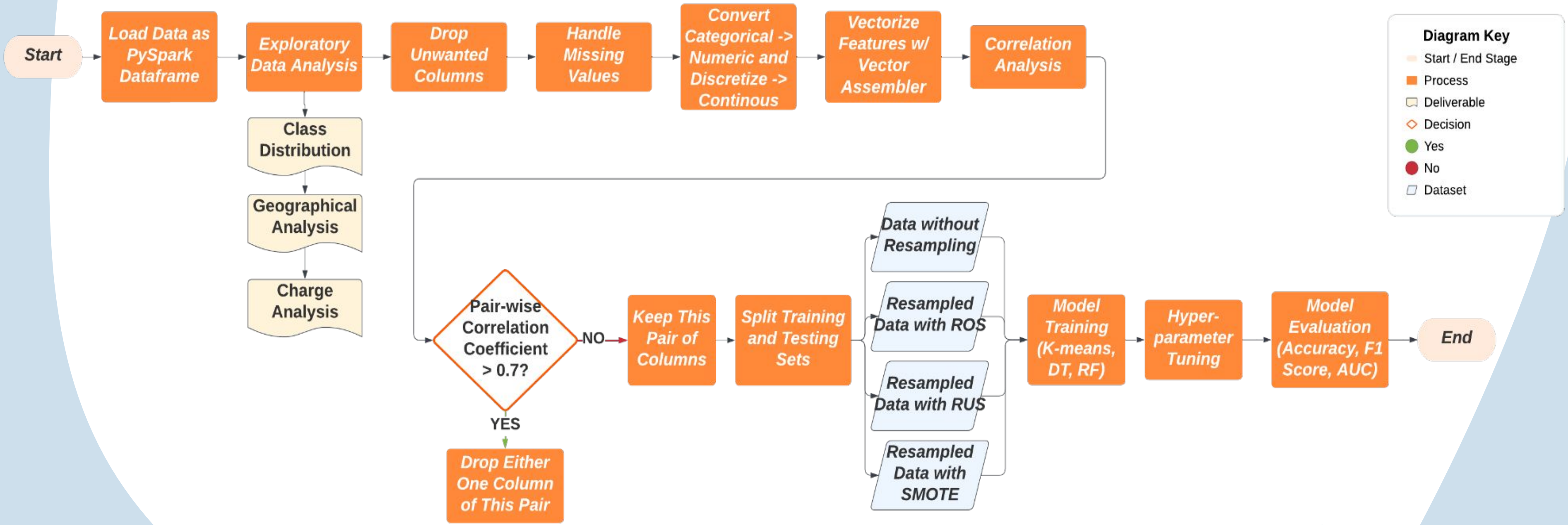
- Demographics
- Location
- Population
- Services
- Status

7K+ data instances

32 features

1 label

Methodology Flowchart



Data Preprocessing

- Drop unwanted columns
- Handle missing values
- Cast columns to correct types

1.Check for missing values

```
df.select([fn.count(fn.when((fn.col(c) == ' ') | (fn.col(c).isNull()), c)).alias(c) for c in df.columns]).show()
```

2.Drop unwanted columns

```
df = df.drop("CustomerID","Count", "Churn Reason","Churn Label", "Country", "Lat Long", "State")
```

3.Handle missing values

```
df = df.withColumn('Total Charges', fn.when(df['Total Charges'] == ' ', None).otherwise(df['Total Charges']))
df = df.withColumn('Total Charges', df['Total Charges'].cast('double'))
print('Number of customers before dropping: {0}'.format(df.count()))
df = df.dropDuplicates()
df = df.na.drop()
print('Number of customers after dropping: {0}'.format(df.count()))
```

4. Output results

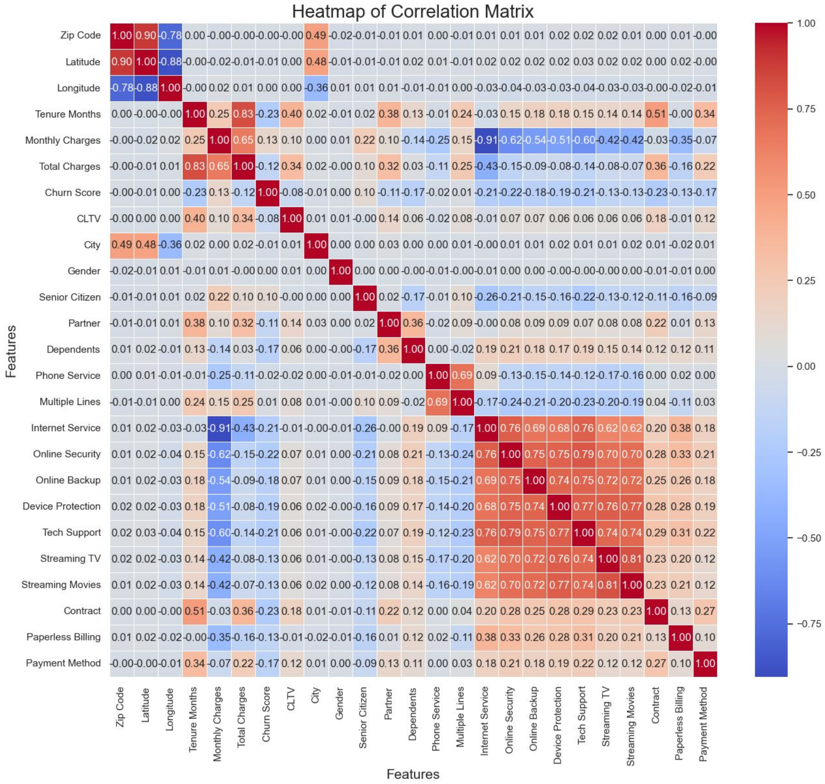
```
Number of customers before dropping: 7043
Number of customers after dropping: 7032
```

Data Transformation

- Convert category to numeric
- Discretize continuous data
- Assemble features into a single vector

```
+-------------------+-----------+
|           features|Churn Value|
+-------------------+-----------+
|(25,[0,1,2,3,4,5,...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|[90810.0,33.81981...|          1|
|[92126.0,32.88692...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|[95412.0,38.73105...|          1|
|(25,[0,1,2,3,4,5,...|          1|
|[90802.0,33.75252...|          1|
|[90046.0,34.10845...|          0|
|(25,[0,1,2,3,4,5,...|          0|
|[92692.0,33.60693...|          0|
|[93402.0,35.27998...|          0|
|[93905.0,36.66779...|          0|
|[94111.0,37.80177...|          0|
|[94569.0,38.03570...|          0|
+-------------------+-----------+
only showing top 20 rows
```

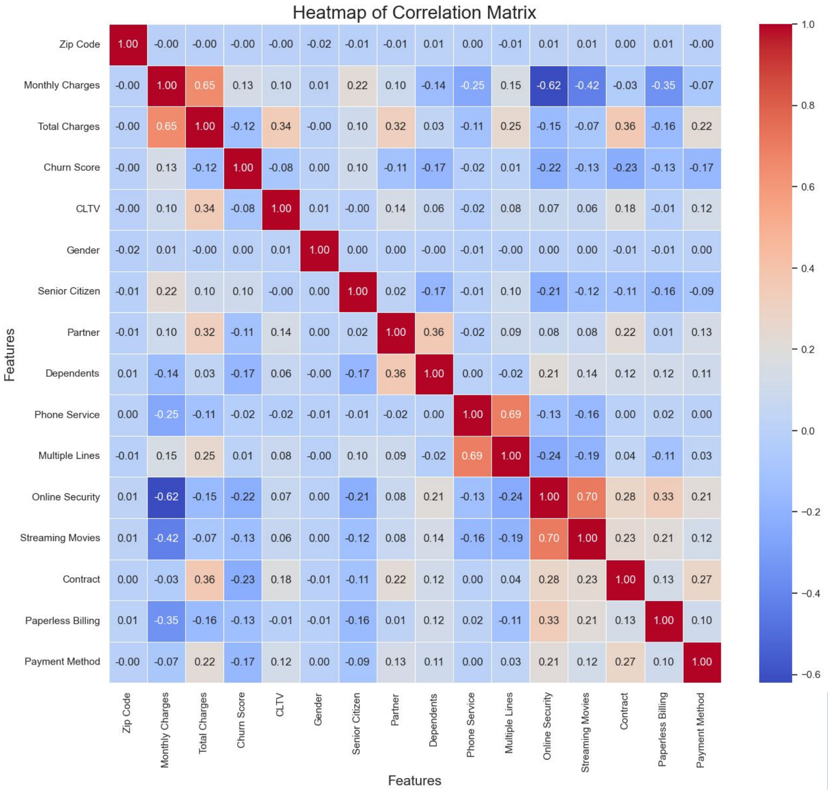
Data Preparation



- Generated correlation matrix
- If pairwise corr <= **0.7**, ignore
- If pairwise corr > **0.7**, drop one and keep the other

Data Preparation

- After dropping highly-correlated columns



Data Resampling

- First split into **70%** training data and **30%** test data
- **ROS increases** the number of the minority class
- **RUS reduces** the number of majority class instances
- **SMOTE creates synthetic samples** by interpolating between existing minority class instances

- **Training data**

Churn Value	count
1	1300
0	3582

- **ROS**

Churn Value	count
0	$\boxed{3635}$
1	3963

- **RUS**

Churn Value	count
0	1208
1	$\boxed{1321}$

- **SMOTE**

Churn Value	count
1	$\boxed{3877}$
0	3635

Model Training

3 Algorithms
Represent 3 Levels of Complexity

- K-means Clustering
- Decision Tree
- Random Forest

ie. Random Forest + ROS

1.Create a Random Forest Classifier

```
# Create an instance of the RandomForestClassifier
rf = RandomForestClassifier(labelCol='Churn Value', featuresCol="features", seed=42)
```

2.Train the model with ros_transformed training set

```
# Train the model
model = rf.fit(df_ros)
```

3.Predict the testing set

```
# Make predictions on the test data
rf_ros_predictions = model.transform(test_df)
rf_ros_predictions.show()
```

Hyperparameter Tuning

Tools Used: `ParamGridBuilder, CrossValidator` **(pyspark.ml.tuning)**

Steps:

1. Build a parameter grid

```
paramGrid = ParamGridBuilder() \
    .addGrid(dt.maxDepth, [2, 5, 10]) \
    .addGrid(dt.maxBins, [10, 20, 30]) \
    .build()
```

2. Create the Cross Validator and fit it with training set

```
# Define the cross-validation method
cv = CrossValidator(estimator=dt, estimatorParamMaps=paramGrid, evaluator=F1_evaluator, numFolds=3)

# Fit the cross-validation model to the training data
cvModel = cv.fit(train_df)
```

3. Output the best combination of parameters

```
# Get the best model
bestModel = cvModel.bestModel

bestMaxDepth = bestModel.getMaxDepth()
bestMaxBins = bestModel.getMaxBins()
print(bestMaxDepth)
print(bestMaxDepth)
```

Model Evaluation

Accuracy

A general metric. It measures the proportion of correctly classified instances out of the total number of instances in a dataset.

Accuracy = (Number of correctly classified instances) / (Total number of instances)

F1 Score

A metric that combines precision and recall into a single value. It is commonly used in binary classification tasks where there is an imbalance between the classes.

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Area Under ROC

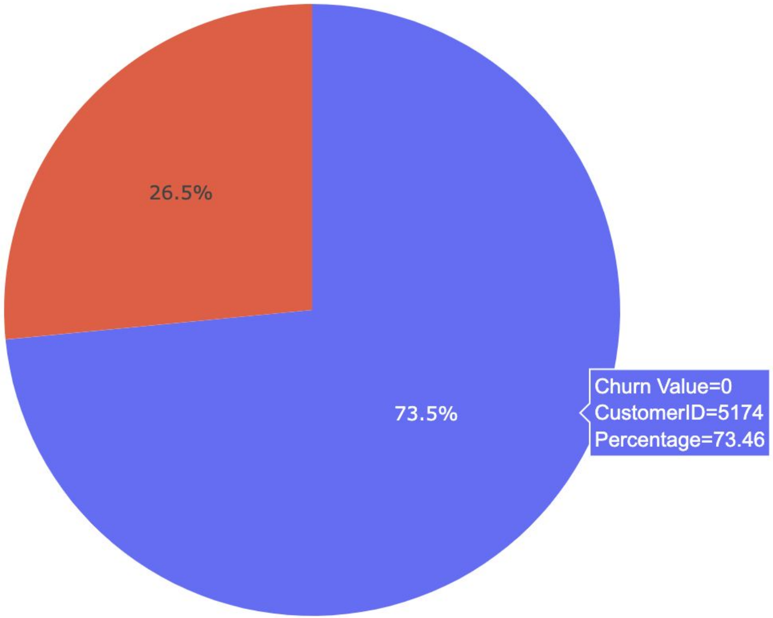
A metric used to evaluate the performance of binary classification models based on their predicted probabilities.

AUC of 0.5 indicates a random classifier, AUC of 1 represents a perfect classifier

05

Data Analysis & Discussion

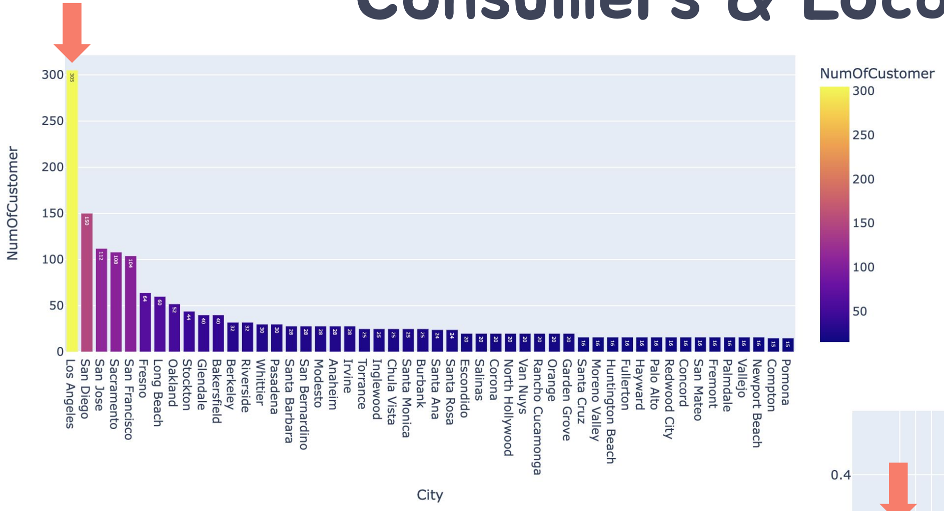
Class Distribution



0: not churn customers
1: churn customers

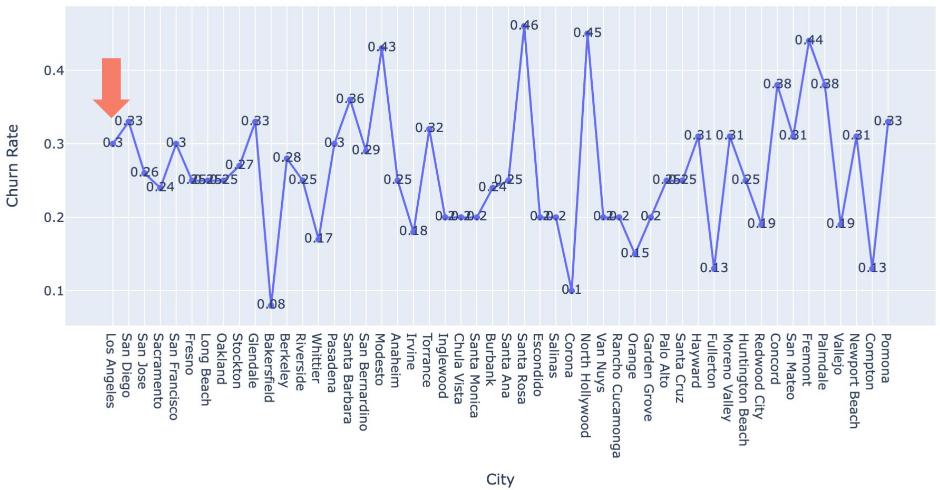
- imbalance data
- most customers not churn

Consumers & Location Analysis

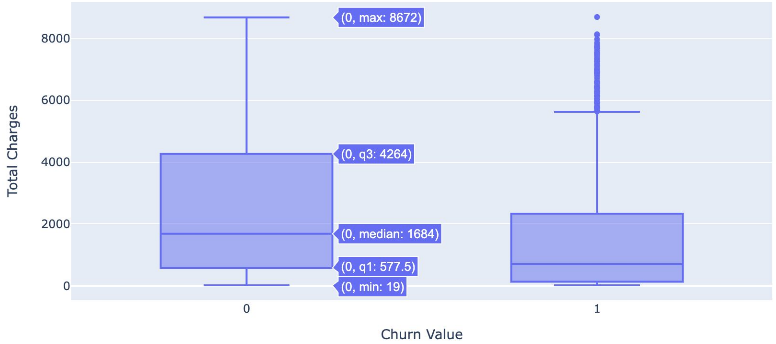


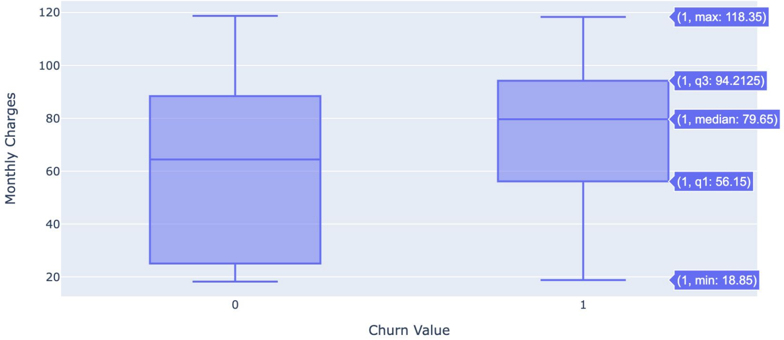
Cities with high consumer numbers do not mean high churn rates

- **Los Angeles**, 305 consumers, 30% churn rate
- **Santa Rosa**, 24 consumers, 46% churn rate



Charges Analysis





churned customers tend to have lower total charges but higher monthly charges

high monthly spend tends to churn users
&
customers who spend more in total are more loyal

Result Discussion

- Random Forest with the Original Dataset performs the best

  Accuracy and AUC achieved 0.892, F1 Score achieved 0.884

	K-means Clustering			Decision Tree			Random Forest		
	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score	Accuracy	AUC	F1 Score
Original	0.750	0.750	0.765	0.882	0.882	0.883	0.892	0.892	0.884
ROS	0.777	0.777	0.789	0.873	0.873	0.879	0.876	0.876	0.882
RUS	0.408	0.408	0.444	0.867	0.867	0.873	0.864	0.864	0.871
SMOTE	0.569	0.569	0.595	0.879	0.879	0.884	0.873	0.873	0.879

Result Discussion

- After Hyperparameter Tuning, performance improved
- Random Forest has the highest accuracy which achieved 0.898
- Accuracy of Decision Tree increased significantly from 0.882 to 0.892

		Not Tune	Tuned
Decision Tree	Accuracy	0.882	0.892
	AUC	0.882	0.851
	F1 score	0.883	0.891
Random Forest	Accuracy	0.892	0.898
	AUC	0.852	0.875
	F1 score	0.891	0.899

06

Conclusion

Project Conclusion

	K-Means Clustering	Decision Trees	Random Forest
Performance Considerations	Average performance; Significant improvement with SMOTE	Excellent performance; Slight decline with ROS and SMOTE	Outstanding performance; Best AUC with ROS
Advantages	Handle large datasets	Easily interpretable; Handles categorical data well	Robust to overfitting; Handles large datasets
Limitations	Struggle with imbalance data	Overfit tendency; Affected by small changes in data	Complicated

Project Conclusion

- These three models have different performance
  - **Decision Tree** and **Random Forest**, renowned for their robustness and versatility, demonstrated **exceptional overall performance**.
  - **SMOTE** exhibited a marked improvement in the AreaUnderROC, accuracy score, and F1-score for **K-Means Clustering**.
- Dealing with **imbalanced data** is **not necessary** under this scenario because the dataset is not extremely imbalanced.
- **Hyperparameters tuning** does **not** have a significant impact.

Future Study Directions

- Telecommunication companies must address the **data imbalance issue** when predicting customer churn because real datasets might be more imbalanced.
- Companies should consider using **different metrics** for model evaluation depending on the business context.

07

Demo

Bibliography

- Geiler, L., Affeldt, S. & Nadif, M. A survey on machine learning methods for churn prediction. Int J Data Sci Anal 14, 217–242 (2022). https://doi.org/10.1007/s41060-022-00312-5
- I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam and S. W. Kim (2019). A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector. In IEEE Access, vol. 7, pp. 60134-60149, 2019, https://doi.org/10.1109/ACCESS.2019.2914999.
- Kihoon Jang, Junwhan Kim, and Byunggu Yu. (2021). On Analyzing Churn Prediction in Mobile Games. I*n 2021 6th International Conference on Machine Learning Technologies (ICMLT 2021)*. https://doi.org/10.1145/3468891.3468895
- Lalwani, P., Mishra, M.K., Chadha, J.S. et al. Customer churn prediction system: a machine learning approach. Computing 104, 271–294 (2022). https://doi.org/10.1007/s00607-021-00908-y
- Li, S., Xia, G., & Zhang, X. (2022, December). Customer Churn Combination Prediction Model Based on Convolutional Neural Network and Gradient Boosting Decision Tree. In Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence (pp. 1-6). https://doi.org/10.1145/3579654.3579666
- Man Zhu and Jieping Liu. (2022). Telecom Customer Churn Prediction Based on Classification Algorithm. *In 2021 International Conference on Aviation Safety and Information Technology (ICASIT 2021)*. https://doi.org/10.1145/3510858.3510945
- Mohd Khalid Awang, Mokhairi Makhtar, and Mohamad Afendee Mohamed. 2020. An Ensemble Method with Cost Function on Churn Prediction. In Proceedings of the 3rd International Conference on Advances in Artificial Intelligence (ICAAI '19). Association for Computing Machinery, New York, NY, USA, 117–121. https://doi.org/10.1145/3369114.3369135
- Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Int. Res. 16, 1 (January 2002), 321–357.
- Nurul Izzati Mohammad, Saiful Adli Ismail, Mohd Nazri Kama, Othman Mohd Yusop, and Azri Azmi. (2020). Customer Churn Prediction In Telecommunication Industry Using Machine Learning Classifiers. *In Proceedings of the 3rd International Conference on Vision, Image and Signal Processing (ICVISP 2019)*. https://doi.org/10.1145/3387168.3387219
- V. S. Spelmen and R. Porkodi, "A Review on Handling Imbalanced Data," 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, 2018, pp. 1-11, doi: 10.1109/ICCTCT.2018.8551020.
- Wang X, Nguyen K, Nguyen BP (2020) Churn prediction using ensemble learning. In: Proceedings of the 4th international conference on machine learning and s computing, (pp. 56–60) https://doi.org/10.1145/3380688.3380710
- Xu, T., Ma, Y., & Kim, K. (2021). Telecom churn prediction system based on ensemble learning using feature grouping. Applied Sciences, 11(11), 4742. https://doi.org/10.3390/app11114742
- Zhu, B., Pan, Y., & Gao, Z. (2018, May). Application of Active Learning for Churn Prediction with Class Imbalance. In Proceedings of the 2018 International Conference on Machine Learning Technologies (pp. 89-93). https://doi.org/10.1145/3231884.3231900

Thanks!
Q & A

Group 1: Johnny Qiu, Manyu Zhang, Yuan Pan, Yue Zhang