Project Name: Telco Churn Classification Project

Project Description: develop and optimize machine learning model that will predict churn using teclo dataset

Project Objectives:

- **Data Pipeline: Acquire data from SQL database and convert into a panda dataframe, prepare the data for exploration, explore the data and document key takeaways, visualize feature attributes and incorporate into machine learning model.**
- **Document data preparation/exploration progress, document key takeaways and present results in Jupyter Notebook**
- **Create modules (acquire.py, prepare.py) that make process repeatable for 3rd party**
- **Present thought process and modeling results to cohort utilizing Jupyter Notebook as presentation material**
- **Be prepared to handle questions about code, process, findings, key takeaways, and model.**

Project Goals:

- **Find drivers for customer churn in the telco data that can be utilized to construct a predictive model**
- **Construct and optimize a machine learning classification model that accurately predicts churn while maximizing for recall**
- **Document the process so that 3rd party can read like a report and easily follow along/replicate**

Project Goals:

Stage	Tools	Brief Description of Process	Takeaways/Lessons Learned
Acquire	• python • pandas • SQL • user-defined functions	• Defined two functions that established connection, ran query and joined tables in SQL ACE and imported into panda df	• Consider importing multiple queries into separate dfs as opposed to bringing the data in on one joined query

Prepare	● python ● pandas ● user-defined functions	● Cleaned raw data (dealt with na, null, blank, values), dropped redundant columns, imputed and encoded data ● Defined function that automated the process	● Keeping the raw data fully intact and untouched in a separate df is a good practice
Explore	● python ● pandas ● matplotlib ● seaborn ● SciPy ● sklearn	● Plotted and visualized features independently and visualized feature relationships vs churn, tenure & monthly charges ● Tested selected features for statistical relevance vs churn for model selection	● In my opinion, the most important part of the pipeline ● Some of the best insights into the data came from experimentation and unplanned exploration. ● A methodically predetermined start didn't yield results but sparked fruitful thoughts ● The power of the visuals can not be overstated and ultimately framed the statistical and model results
Model	● sklearn	● Defined 3 models and evaluated them against each other ● I used Decision Tree, Random Forest & Logistic Regression machine learning models	● Think of modeling as an inverted pyramid. The Test set is where everything is done. The Validate set is touched with every model and the Test is only touched with gloves
Evaluate	● python ● pandas ● sklearn	● Established baseline ● Fit the models to the train data and then evaluated initially on accuracy ● Used sklearn confusion matrix and classification	● Understanding what common evaluation metrics like (Precision, Recall, Accuracy) measured was key to optimizing

		reports to compare model performance	models ● Understanding where and how the predictive value addressed the root problem was key to selecting which metrics to measure which model outperformed
Model Explanation	**How does your algorithm work?**	All 3 of my models appeared to perform equally well on my accuracy metric. Initially, my Logistic Regression appeared to significantly outperform the Decision Tree and Random Forest models on Recall. However, given that I had set out maximizing Recall as a goal, I continued to tweek hyperparameters and re-evaluated my models. Accuracy metrics remained fairly consistent but by adjusting the max_depth on my Decision Tree I was able to keep my accuracy performance in line with previous test but maximize Recall score. For this reason, I chose my Decision Tree model.	Had I not prioritized my Recall score as a goal beforehand, I probably would have chosen my Logistic Regression after my initial tests. The Recall scores on the other models in that initial round badly underperformed so much I would have felt comfortable choosing my logit model at that point. In fact, I did not expect my next iterations to be fruitful for the DT & RF models and was mainly trying to improve Recall on my Logit model. Recommended Good Practices: ● Test all models at each iteration ● Understand the issues you're trying to address by developing a predictive model and establish evaluation metric goals before evaluating models.